

**ESS-2003**  
**SCIENTIFIC PROGRAMME**



**SIMULATION  
METHODOLOGIES,  
METHODS AND  
TECHNIQUES**



# IMPLEMENTATION ISSUES FOR SHARED STATE IN HLA-BASED DISTRIBUTED SIMULATION

Malcolm Yoke Hean Low  
Boon Ping Gan

Singapore Institute of  
Manufacturing Technology  
71 Nanyang Drive, Singapore 638075

Junhu Wei, Xiaoguang Wang,  
Stephen John Turner, Wentong Cai

School of Computer Engineering  
Nanyang Technological University  
Nanyang Avenue, Singapore 639798

## KEYWORDS

Shared State, Distributed Simulation, High Level Architecture, Zero Lookahead.

## ABSTRACT

The problem of shared state is well known to the parallel and distributed simulation research community. In this paper, we revisit the problem of shared state in the context of a High Level Architecture based distributed simulation. A middleware approach is proposed to solve this problem within the framework of the High Level Architecture Runtime Infrastructure. Four solutions to this problem are implemented in the middleware using receive-order messages. We will discuss the implementation issues of these four solutions in the middleware. Experimental results comparing the performance of these four solutions against a simple request-reply approach using time-stamp-order messages are also presented.

## 1 INTRODUCTION

Simulation has traditionally been used as a tool to perform “what if” analysis in complex systems such as the operations of a manufacturing production facility. In a supply chain scenario with multiple business partners each with its own manufacturing production facility, it is not sufficient for each business partner in the supply chain to optimize its operation using simulation. In order to optimize the performance of the whole supply chain, these simulation models need to be integrated before any meaningful analysis can be made. Physically integrating these simulation models into a single simulation environment is often not possible due to 1) the complexity of individual models; 2) business partners being geographically dispersed; and 3) confidentiality in sharing certain parts of the simulation model. Distributed simulation offers a solution to this problem by allowing existing simulation models to be reused and integrated with other simulation models in the supply chain through well-defined interfaces. Each supply chain simulation can also selectively expose only the necessary data to other partners in the supply chain simulation.

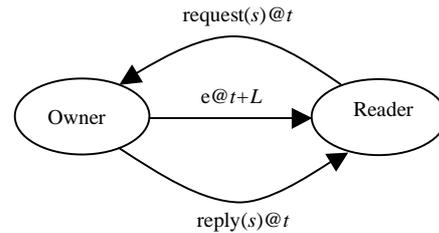


Figure 1: Request-Reply using TSO Messages

An emerging standard for distributed simulation, namely the High Level Architecture (HLA) standard has been proposed by the U.S. Department of Defense (DoD) (Kuhl et al. 1999). The HLA defines the rules and specifications to support reusability and interoperability of different simulators. In HLA terminology, a single simulator is referred to as a federate. A federation is then a set of federates working together to achieve a given goal. Each federate defines the objects and interactions that are shared in its simulation object model (SOM) and interacts with one another over the Runtime Infrastructure (RTI) (DMSO 2002).

The problem of shared state is a well-known problem in both the parallel, as well as the distributed simulation research community. In the context of an HLA-based distributed simulation, the issue involves the following: an “owner” federate updates a local shared variable periodically and multiple remote “reader” federates can access the value of the shared variable instantaneously (at the same simulation time as the read). For the rest of this paper, we assume that the HLA-based distributed simulation runs on an RTI that uses a conservative synchronization protocol for its time management service.

Figure 1 shows a straightforward implementation to support shared state in an HLA-based distributed simulation using a set of time-stamp-order (TSO) request-reply interactions both time-stamped at the simulation time  $t$ . In the example, the owner federate updates a shared variable  $s$  periodically, and also sends a TSO event  $e$  to the reader federate periodically with a lookahead of  $L$ . At simulation time  $t$ , the reader federate accesses the value of the shared variable  $s$  by

sending a TSO request message time-stamped at  $t$ . The owner federate, on receiving the request message, will issue a TSO object update with simulation time  $t$ . We refer to this approach to support shared state on an HLA-based distributed simulation as the *PullTSO* approach.

In order for the *PullTSO* scheme to work, both the owner and the reader federates must be time-constrained and must regulate the federation with a lookahead of zero. However, having zero-lookahead in a federation is often detrimental to the performance of the simulation system whether the underlying RTI is using conservative or optimistic time synchronization. Suppose the request-reply TSO messages in Figure 1 are replaced by receive-order (RO) messages. Since RO messages are not used by the RTI to check lookahead and the time advancement constraint on each federate, the owner federate can regulate the federation with a lookahead of  $L$  instead.

In this paper, we address the issue of shared state in an HLA-based distributed simulation by replacing the TSO request-reply messages in the *PullTSO* approach with RO messages. We propose a middleware approach to support shared state. We will outline four solutions to support shared state within the middleware. While the use of the middleware approach is to hide the implementation of the support for shared state from the users, it must also preserve the semantics of the different RTI APIs that are used by the users. We will explain how this can be achieved in the middleware using an example. The performance of the solutions proposed will also be compared against that of the *PullTSO* approach.

The rest of the paper is organized as follows. Section 2 describes some related work in solving the shared state problem. Our solutions to shared state using the middleware approach will be described in Section 3. In Section 4, we discuss some issues in implementing the solutions in the middleware. Experimental results comparing our solutions to the *PullTSO* approach will be presented in Section 5. We conclude this paper in Section 6 and outline further work in this area.

## 2 RELATED WORK

The issue of shared state has been explored in the context of several parallel simulation research projects. For example, in the work by Mehl and Hammes (Mehl and Hammes 1993), they proposed two general approaches to implement shared variables using a conservative synchronization algorithm, namely 1) request-reply and 2) cached-copy. In the request-reply approach, the owner keeps a history list of the shared variable. When a reader requests the shared variable at simulation time  $t$ , the owner will wait until it is certain that no other write messages will be received with simulation time smaller than  $t$  before it retrieves the

value of the shared variable at time  $t$  from its history list and forwards the reply to the reader.

The request-reply approach frees the owner from the time constraints from its readers and allows the owner to proceed ahead of its readers whenever possible. However, a reader has to always suspend itself whenever it needs to access the value of the shared variable from the owner. The cached-copy approach proposed by Mehl and Hammes solves this problem by having each reader keep a cached-copy of the shared variable. The cached-copy of the shared variable has a time-guarantee associated with it. Whenever a reader needs to access the shared variable, it first checks the validity of its cached-copy of the shared variable. It will send a request to the owner if such copy is not found or the copy is invalid.

Lim et al. (Lim et al. 1998) showed that if the request/write/reply messages are sent using time-stamp order, then the request-reply approach proposed by Mehl and Hammes could function without a history list. While Mehl and Hammes use a cache-on-demand policy to update the reader's local cache copy of the shared variable, Lim et al. explored an always-update-by-writer update policy for the cached-copy approach whereby the owner will forward each update to the shared variable to all its readers.

The issue of shared variables has also been raised during the third meeting of the HLA-CSPIF forum (HLA-CSPIF 2002). The aim of this forum is to create a standardized approach to distributed simulation using HLA to support interoperation of discrete event models created in commercial-off-the-shelf (COTS) simulation packages. The group is currently looking at specifying reference models for testing the interoperability of different simulators. In particular, one of the reference models under discussion requires the implementation of shared variables across two or more different COTS simulation packages.

A preliminary report on the work described in this paper can be found in Gan et al. (Gan et al. 2003) in which two of the solutions presented in this paper are first proposed. In this paper, we extend on our previous work and propose two more solutions to solving the shared state problem. We will also discuss implementation issues to enable support for shared state in a middleware for RTI.

## 3 THE SOLUTIONS

Both the request-reply and the cached-copy solutions proposed by Mehl and Hammes can be implemented easily under HLA. However, both the request and reply messages, as well as the update message for the cached-copy, will have to be sent using TSO interactions and object updates at the current simulation time. This implies that both the owner and

reader federates must regulate the federation with zero lookahead.

In this section, we describe four solutions to solve the problem of shared state. We will refer to these four solutions as: 1) *PullRO*, 2) *PushRO*, 3) *PullROTG* and 4) *PushROTG*. The *PullRO* and *PushRO* solutions will be described briefly as they have previously been described in detail in Gan et al. (Gan et al. 2003). All four solutions involve replacing the TSO interactions and object updates used in the standard request-reply and cached-copy solutions with RO messages. This eliminates one source of zero lookaheads in the federation and allows both the owner federate as well as the reader federates to regulate the federation with the next smallest (possibly non-zero) lookahead. Similar to the work carried out by Lim et al. (Lim et al 1998), our solutions currently assume that the owner is the only writer for the shared variable. We plan to eliminate this assumption in the next stage of our work.

### 3.1 Solution 1: *PullRO*

In the *PullRO* approach, whenever a reader needs the latest value of a shared variable, it sends an RO interaction to the owner to request for the value at a specific request time. The owner, on the other hand, maintains a history list of all the updated values of the shared variable with their associated update times. Suppose the owner federate is at simulation time  $t_1$  and it receives a request from a reader with request time  $t_2$ . If  $t_1 \geq t_2$ , the owner federate searches its history list for two consecutive entries with update times  $t_j$  and  $t_k$  such that  $t_j \leq t_2 < t_k$ . An RO object update with the value of the shared variable at simulation time  $t_j$  will be sent back to the reader. If  $t_1 < t_2$ , the request will be buffered and serviced when the owner's simulation time reaches  $t_2$ .

### 3.2 Solution 2: *PushRO*

In the *PushRO* solution, we use a cached-copy approach with an always-update-by-writer update policy similar to that used by Lim et al. (Lim et al. 1998). Whenever the owner updates the shared variable, it also sends the update to each of its readers. This update is associated with the timestamp of the update and is sent using RO interactions or object updates. As the owner may be sending an update with timestamp greater than the simulation time of a reader, each reader must also keep a future list to store the updates received from the owner.

In situations in which one of the readers runs ahead of the owner, the reader will not find any valid entry in its future list. In this case, the reader adopts a *PullRO* approach and sends the owner an RO interaction to request for the value of the shared variable. This request will be serviced as in the case of *PullRO* when the owner's simulation time reaches the request time.

### 3.3 Solution 3: *PullROTG*

In the *PullROTG* approach, we augment the *PullRO* solution with a time-guarantee feature. Each update entry in the owner's history list is now associated with a time-guarantee for the valid duration of the entry. This time-guarantee is automatically determined by our middleware and requires no input from the users. For example, if the owner updates a shared variable  $A$  at time  $t_1$ , a time-guarantee is also associated to the length of validity for the value of  $A$ . Initially, without any additional information from the user, the middleware cannot assign any effective time-guarantee to the value of  $A$  at the point of the update. Hence, the update entry for  $A$  at time  $t_1$  is associated with a time-guarantee of  $t_1$ , effectively representing a time-guarantee of zero.

However, if the owner subsequently updates the shared variable  $A$  at time  $t_2$ , the entry of  $A$  at time  $t_1$  in the history list can then be modified to be associated with a time-guarantee  $t_2$ . This means that the update at time  $t_1$  is valid up to time  $t_2$ . Each reader, on the other hand, will also keep a copy of the shared variable and its associated update time and time-guarantee. Suppose a reader requests the shared variable at time  $t_3$  ( $t_1 \leq t_3 < t_2$ ), it will receive an update from the owner with the value of the shared variable at time  $t_1$  and an associated time-guarantee  $t_2$ . The reader can use this local copy of the update for subsequent requests with time smaller than  $t_2$ .

If the owner receives a request with time  $t > t_{\text{curr}}$ , where  $t_{\text{curr}}$  is the current simulation time of the owner, the request will first be buffered. The request will be serviced when the owner's simulation time reaches  $t$ . The value of the shared variable at time  $t$ , together with the associated time-guarantee,  $t_g$ , will be forwarded to the reader. Note that in this case, the time-guarantee,  $t_g$ , for the update message is computed as follows:

$$t_g = \min(t_{\text{NER}}, t_{\text{MNE}}) \quad (1)$$

where  $t_{\text{NER}}$  is the cutoff time specified by the owner federate when invoking any of the RTI time advancement APIs; and  $t_{\text{MNE}}$  is the time obtained by invoking the `RTI:queryMinNextEventTime()`. One assumption made here is that if the owner federate is at simulation time  $t$  and asks the RTI for time advance to time  $t_{\text{NER}}$ , it will not update its shared variables between  $t$  and  $t_{\text{NER}}$ .

### 3.4 Solution 4: *PushROTG*

Similarly, we also augment the *PushRO* solution with the time-guarantee feature. In the *PushROTG* approach, each update entry in the future list maintained by a reader is also associated with a time-guarantee. Note that each of these updates received from the owner initially carries no effective time-guarantee, i.e. the time-guarantee is the same as the

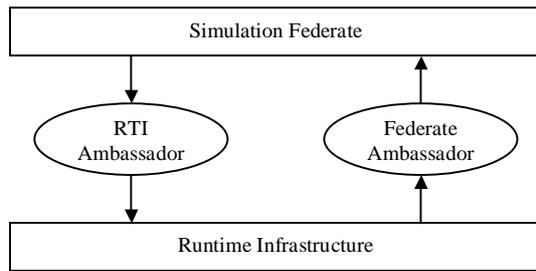


Figure 2: Architecture of RTI

update time. However, whenever a new update is received from the owner, the time-guarantee for the previous entry in the future list is associated with the update time of the newly received update. We note that this form of time-guarantee offers no distinct advantage over the *PushRO* approach whenever the reader runs behind the owner in simulation time since the search for a match between two consecutive entries in the future list already uses the concept of time-guarantee implicitly.

However, whenever a reader runs ahead of the owner, it adopts the *PullRO* approach and sends a request to the owner. The update message it receives from the owner has a time-guarantee computed using equation 1. This update message will be placed at the top of the future list. The time-guarantee provided by this update can further reduce the number of request messages being forwarded to the owner.

## 4 IMPLEMENTATION ISSUES

In this section, we discuss the implementation issues of providing shared state support using the four solutions in a middleware for HLA-based distributed simulation. We will first describe the RTI+ middleware that supports shared state in Section 4.1. The implementation of the four solutions in RTI+ will be described in Section 4.2. Section 4.3 describes how some of the methods in RTI+ should be implemented in order to preserve the semantics of the HLA-RTI APIs. Section 4.4 discusses the implementation issues for fossil collection and late arriving federates.

### 4.1 Middleware to Support Shared State

Figure 2 shows how a simulation federate is typically integrated with the HLA-RTI. The simulation federate can send interactions or object updates to other federates through the RTI using the RTI ambassador. Conversely, the RTI delivers interactions or object updates to the simulation federate via the federate ambassador (implemented as callback functions by the simulation federate).

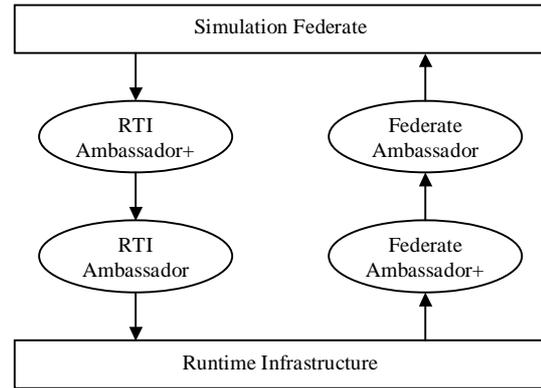


Figure 3: RTI+ Middleware

We extended the HLA-RTI architecture with a middleware layer, which we refer to as the RTI+. Figure 3 shows the extended HLA-RTI architecture. In the extended architecture, all outgoing interactions from the simulation federate to the RTI are routed through the RTI ambassador+. All incoming interactions from the RTI to the simulation federate are routed through the federate ambassador+. Both the RTI ambassador+ and the federate ambassador+ support the full set of interfaces in RTI ambassador and federate ambassador respectively. All four solutions described in the previous section, together with the *PullTSO* solution, are implemented in the RTI+.

For the rest of this paper, we will use the format `RTI:xxx` and `FedAmb:xxx` to denote methods in the original RTI library; and `RTI+:xxx` and `FedAmb+:xxx` to denote methods in the RTI+ middleware. Some of the interfaces are extended in the RTI+ library in order to support shared state. For example, the `RTI+:registerObjectInstance` method is extended to create a history list for objects at an owner federate, and a future list for objects at a reader federate.

Two additional methods are also provided in the RTI ambassador+ to allow a simulation federate to request for an object or class update at a specific simulation time. Figure 4 shows the APIs of these two methods.

```
void requestObjectAttributeValueUpdate(ObjectHandle
                                     theObject, RTI::FedTime theTime)

void requestClassAttributeValueUpdate(ClassHandle
                                     theClass, RTI::FedTime theTime)
```

Figure 4: Shared State APIs for RTI+

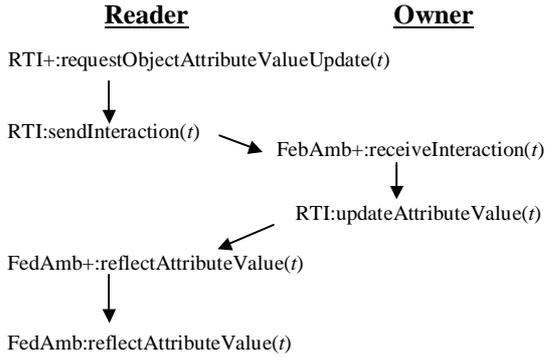


Figure 5: The *PullTSO* Approach

Using the *PullTSO* approach, both methods will be translated to a TSO interaction that is sent from the reader to the owner. This interaction contains the object/class handle that the reader is requesting, and the simulation time at which the value is needed. Figure 5 illustrates this mechanism. A call to the `RTI+:requestObjectAttributeValueUpdate` method in the middleware is translated to a TSO `RTI:sendInteraction` call in the original RTI ambassador at the reader federate. A `FebAmb+:receiveInteraction` callback is triggered in the middleware’s federate ambassador at the owner side. The middleware at the owner federate processes the request by replying through a call to `RTI:updateAttributeValue`. Note that the `FebAmb+:receiveInteraction` is not allowed to call the `RTI:updateAttributeValue` within the federate ambassador+. Hence, the middleware needs to first record the request, and processes it once control is returned to the RTI ambassador+. Control is transferred to the middleware when the user invokes the `RTI+:tick` method. The `RTI+:tick` method needs to perform two tasks: 1) process all pending requests from external readers in the method `RTI+:processSysInteraction`; and 2) yield control to the RTI by calling the original `RTI:tick` method.

#### 4.2 Implementation using RO Messages

In this section, we discuss the implementation using RO messages to deliver the request-reply messages between owners and readers of shared variables. While the details of the implementation are illustrated using the *PullRO* solution, the same principle applies to the other three solutions.

Figure 6 illustrates the sequence of method calls to implement the *PullRO* approach. Note that the entry and exit points to the sequence of method calls are the same as those in Figure 5. This means that the underlying solutions used to support shared state in the middleware is transparent to the user. The middleware at the reader federate translates the TSO request call to an RO `RTI:sendInteraction`. The owner federate in turn replies to the reader using an RO `RTI:updateAttributeValue`.

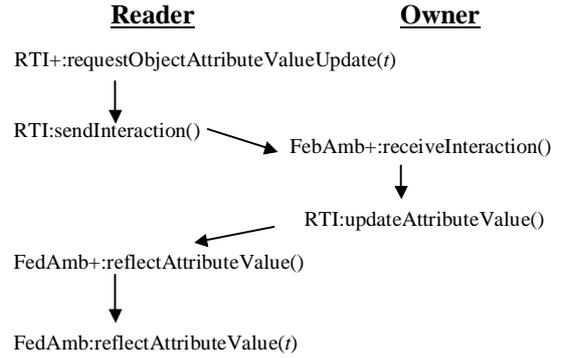


Figure 6: The *PullRO* Approach

At the reader end, once a request is issued, the reader is not allowed to progress in time, until it receives the update that corresponds to its request. This is realized by keeping track of pending requests, and withholding the time advance request (a primitive in RTI that federates use to try to advance their time) until the request is received.

#### 4.3 Semantics of RTI APIs

The semantics of RTI APIs has to be preserved when implementing support for shared state in the RTI+ middleware no matter which solutions the user chooses to use. A naive implementation may alter the semantics of the original RTI APIs. We illustrate this issue using the implementation of `RTI+:tick`. As mentioned in the previous section, the `RTI+:tick` method needs to perform two specific tasks, `RTI+:processSysInteraction` and call `RTI:tick`. However, the order in which these two tasks are carried out will potentially violate the specification of the RTI APIs.

Suppose the tick method is implemented such that the `RTI+:processSysInteraction` task is executed first, followed by the call to the original `RTI:tick` method. Consider the example in Figure 7 showing a sequence of method calls from an owner federate using the *PullTSO* approach. The owner federate first requests for time advance to simulation time 4. The RTI+ middleware will forward the request to the RTI. The owner federate subsequently yields control to the RTI+ middleware by calling the `RTI+:tick` method. The `RTI+:processSysInteraction` subtask in the `RTI+:tick` method does nothing since there is no pending request. However, when the original `RTI:tick` method is called, a TSO interaction carrying a request from a reader to access a shared variable at simulation time 3 is delivered. Note that this request will not be processed until the next call to `RTI+:processSysInteraction`. The owner federate is subsequently granted a time advance to simulation time 4.

Suppose the owner next issues a time advance request to advance to simulation time 8, and proceeds to yield control to the RTI+ middleware by calling the `RTI+:tick`

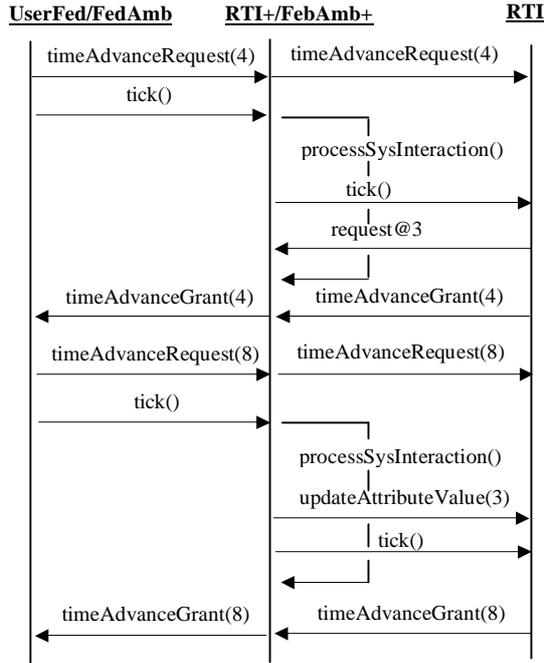


Figure 7: Semantics of RTI APIs

method. In the `RTI+:processSysInteraction` method, the request from the reader at simulation time 3 is processed. This results in a TSO object update event time-stamped at 3 being sent out to the reader.

However, sending out the TSO event time-stamped at 3 will immediately cause the RTI to generate an “*InvalidFedTime*” exception. The reason for this is that after the middleware invokes the method `RTI:timeAdvanceRequest` with request time 8, it “may not generate TSO events whose time stamps are less than the requested time plus the federate’s lookahead” (DMSO 2002). Thus, the request at simulation time 3 should not have been sent out whilst the `RTI:timeAdvanceRequest` with request time 8 is in progress.

The correct approach to implementing the `RTI:tick` method would be to reverse the order of executing the two tasks. In the example above, executing `RTI:tick` before `RTI+:processSysInteraction` would allow RTI to deliver the request at time 3 to the owner federate. This request would be processed by the `RTI+:processSysInteraction` that executes next. The middleware at the owner federate will generate an object update at simulation time 3 and send it to the reader before the owner federate is granted an advance to simulation time 4.

#### 4.4 Late Arriving Federates and Fossil Collection

Other implementation issues that have to be addressed are the problem of late arriving federates and fossil

collection. We will describe how these two issues are resolved in this section.

In the initial implementation of the *PushRO* and *PushROTG* solutions, an assumption is that the reader will receive all updates from the owner federate. The only situation in which a reader actively requests values of a shared variable from the owner is when the reader runs ahead of the owner and all the cached values of the shared variable in the reader’s future list have time-stamp (or time-guarantee) smaller than its simulation time.

However, the RTI also allows federates to join at any time an existing running federation. Such federates are termed as late arriving federates. Suppose one of the reader federates is a late arriving federate which joins the federation at simulation time  $t$ . If the owner federate is at simulation time  $t_1$  ( $t_1 \geq t$ ), some of the updates sent from the owner between  $t$  and  $t_1$  may be “lost” to the reader federate.

To handle this problem, an owner federate employing the *PushRO* or *PushROTG* scheme must also keep a history list for all its shared variables. A new reader federate which joins the federation midway will request for the values of shared variables using the same approach as in the case of *PullRO* and *PullROTG*.

The entries for each update to a shared variable are kept in a history list at an owner federate or in a future list at a reader federate. The memory used by the outdated entries has to be fossil collected periodically. While an owner federate has to keep those entries with time-stamp greater than the smallest federate time of all its readers (in all four RO solutions), a reader federate needs only to keep those entries with time-stamp greater than its own federate time. The federate times of individual reader federates can be obtained by accessing services provided by the Management Object Model (Fullford and Wetzel 1999) in order to determine the reader federate with the smallest time-stamp.

## 5 EXPERIMENTAL RESULTS

Experiments were carried out to compare the performance of the four solutions using RO messages against the *PullTSO* approach. The experiments were carried out using the DMSO RTI 1.3-NG version 5. The simulation model consists of two federates: an owner federate and a reader federate. The owner federate updates a shared variable every 100 time units, while the reader federate accesses the shared variable every  $100r$  time units, where  $r$  is the request ratio. We experimented with different request ratios  $r = (0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0)$ . We also experimented with different lookaheads between the two federates. The lookahead is modelled by sending a TSO user

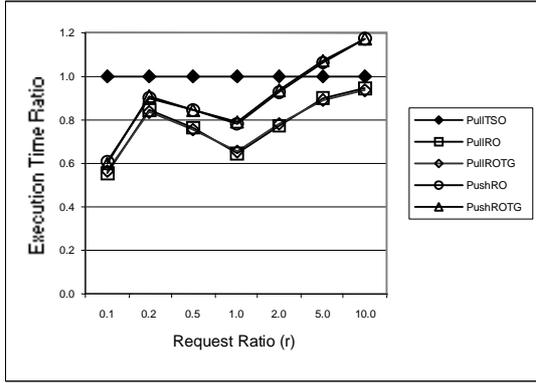


Figure 8: Execution time ratio vs Request Ratio (Lookahead = 10)

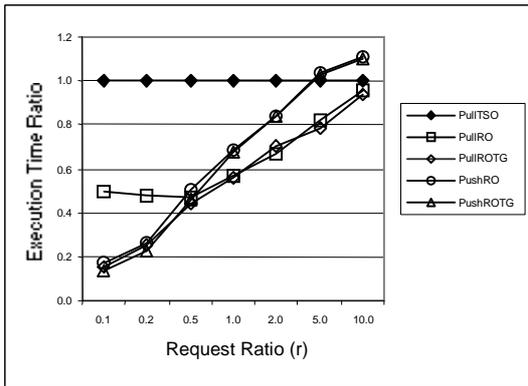


Figure 9: Execution time ratio vs Request Ratio (Lookahead = 100)

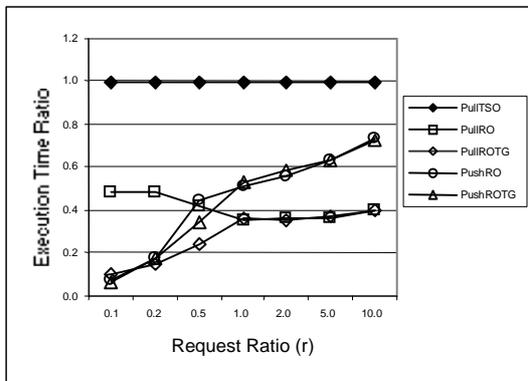


Figure 10: Execution time ratio vs Request Ratio (Lookahead = 1000)

interaction from the owner to the reader each time the owner updates the shared variable. We experimented with different values of lookaheads  $l = (0, 10, 100, 1000)$ . If the owner updates the shared variable at simulation time  $t$ , then the user interaction is time-stamped with the simulation time  $t+l$ .

Figures 8, 9 and 10 show the execution time ratios using the four solutions for different values of lookaheads. The execution time ratio is the ratio between the execution time of the respective RO solutions against the execution time using *PullTSO*. Hence the *PullTSO* version is depicted with an

	Lookahead=100		
Request Ratio	0.1	0.2	0.5
<i>PullRO</i>	99999	49999	19999
<i>PullROTG</i>	19744	21041	17307
<i>PushRO</i>	25122	23720	19900
<i>PushROTG</i>	13972	13607	10698
	Lookahead=1000		
Request Ratio	0.1	0.2	0.5
<i>PullRO</i>	99999	49999	19999
<i>PullROTG</i>	10000	10000	10000
<i>PushRO</i>	34	12620	19730
<i>PushROTG</i>	35	10000	10000

Table 1: No. of Requests Received by the Owner Federate

execution time ratio of 1.0. Table 1 shows the number of requests received by the owner federate for request ratios 0.1, 0.2 and 0.5.

From the three graphs, we see that the execution times of the four RO solutions generally decrease compared to the *PullTSO* approach as the minimum lookahead of the system is increased from 10 to 1000. This shows that the RO solutions free the owner and reader federates from one source of zero lookahead caused by the sending of a TSO request-reply message and allows them to regulate the federation with the lookahead imposed by the user interaction sent from the owner to the reader federate.

For small request ratios, the reader federate requests the values of the shared variable more frequently than the updates by the owner federate. This results in a large number of request messages being sent from the reader to the owner if *PullRO* is used. However, this scenario favors the *PushRO* solution since updates only need to be sent out infrequently. Most of the accesses to the shared variable by the reader federate can be fulfilled by the middleware using entries in the future list without the need to request new values from the owner federate. The graphs also show that the *PullROTG* solution is also able to deliver performance comparable to that of the *PushRO*. The time-guarantee provided by the owner federate is used effectively to satisfy requests from the reader federate without the need to request updates from the owner federate.

Table 1 confirms that the number of requests received by the owner federate is significantly fewer using the *PushRO* version compared to the *PullRO* version for small request ratios. The number of requests using the *PullROTG* solution is also comparable to that of the *PushRO* version for lookahead=1000. The additional time-guarantee provided by the *PushROTG* solution further reduces the number of requests received by the owner federate.

For large request ratios, the performance of both *PullRO* and *PullROTG* solutions are consistently better than the *PushRO* and *PushROTG* solutions. In fact, for

the runs with lookaheads = 10 and 100 and request ratio  $r \geq 5.0$ , both *PushRO* and *PushROTG* yield worse performance compared to the *PullTSO* solution. In this case, the owner updates the value of the shared variable more often than the reader accesses it. Thus, most of the updates from the owner received by the reader in the *PushRO* or *PushROTG* versions are redundant.

## 6 CONCLUSION

In this paper, a middleware approach to support shared state in an HLA-based distributed simulation has been described. Detailed discussions for some issues in implementing the four RO solutions in the middleware are also presented. Our experiments show that the four RO solutions proposed and implemented in the RTI+ middleware are indeed much more efficient compared to the *PullTSO* approach. The experimental results also show that the additional time-guarantee provided by the *PullROTG* solution allows further performance improvement compared to the *PullRO* solution when the request ratio is small.

This middleware approach for shared state will also be tested on more realistic simulation models to better evaluate the effectiveness of the four solutions. Further work will also be carried out to explore the possibility of applying similar techniques to address the issue of remote write.

## REFERENCES

- DMSO. 2002. RTI 1.3-Next Generation Programmer's Guide Version 5, DoD, DMSO, Feb 2002.
- Fullford D. and D. Wetzel. 1999. "A Federation Management Tool: Using the Management Object Model (MOM) to Manage, Control, and Monitor a Federation". In *Proceedings of the 1999 Spring Simulation Interoperability Workshop*, 99S-SIW-196.
- Gan B.P.; M.Y.H. Low; J.-H. Wei; X.-G. Wang; S.J. Turner and W.-T. Cai. 2003. "Synchronization and Management of Shared State in HLA-Based Distributed Simulation". To appear in the 2003 Winter Simulation Conference.
- HLA-CSPIF 2002. 3<sup>rd</sup> Meeting of HLA-CSPIF Forum, 20-21 November 2002, Savill Court Hotel, London. <http://www.cspif.com>.
- Kuhl F.; R. Weatherly and J. Dahmann. 1999. "Creating computer simulation systems: An introduction to the High Level Architecture". Prentice Hall PTR.
- Lim C.C.; Y.H. Low; B.P. Gan and S. Jain. 1998. "Implementation of Dispatch Rules in Parallel Manufacturing Simulation". In *Proceedings of the 1998 Winter Simulation Conference*, 1591-1597.
- Mehl H. and S. Hammes. 1993. "Shared Variables in Distributed Simulation". In *Proceedings of 7th Workshop on Parallel and Distributed Simulation*, 68-75.

## AUTHOR BIOGRAPHIES

**MALCOLM YOKE HEAN LOW** is a Research Engineer with the Production and Logistics Planning Group at the Singapore Institute of Manufacturing Technology. He received his Bachelor and Master of Applied Science in Computer Engineering from Nanyang Technological University, Singapore in 1996 and 1997 respectively, and a D.Phil. in Computer Science from Oxford University in 2002. His research interests are in the areas of adaptive tuning and load-balancing for parallel and distributed simulation systems, and the application of multi-agent technology in supply chain logistics coordination. His email address is [yhlow@SIMTech.a-star.edu.sg](mailto:yhlow@SIMTech.a-star.edu.sg).

**BOON PING GAN** is a Research Engineer with the Production and Logistics Planning Group at Singapore Institute of Manufacturing Technology (formerly known as Gintic Institute of Manufacturing Technology). He is currently leading a research project that attempts to apply distributed simulation technology for supply chain simulation. He received a Bachelor of Applied Science in Computer Engineering and Master of Applied Science from Nanyang Technological University of Singapore in 1995 and 1998 respectively. His research interests are parallel and distributed simulation, parallel programs scheduling, and application of genetic algorithms. His email address is [bpgan@SIMTech.a-star.edu.sg](mailto:bpgan@SIMTech.a-star.edu.sg).

**JUNHU WEI** is working with Nanyang Technological University (Singapore) as a Research Fellow. He received his BE in Automatic Control and ME in System Engineering and PhD in Control Engineering from Xi'an Jiaotong University (China). His current research interests include parallel and distributed simulation, Simulation, Planning and Scheduling of Manufacturing. His email address is [asjhw@ntu.edu.sg](mailto:asjhw@ntu.edu.sg).

**XIAO GUANG WANG** is currently a Ph.D student at School of Computer Engineering (SCE), Nanyang Technological University, Singapore. She received her B.Sc in Computer Science from Nanjing University of Aeronautics and Astronautics, China in 1997. Her research interests lie in Distributed Simulation and High Level Architecture, which is also her Ph.D topic currently being developed. Her email address is [PG02355670@ntu.edu.sg](mailto:PG02355670@ntu.edu.sg).

**STEPHEN J. TURNER** joined Nanyang Technological University (Singapore) in 1999 and is currently an Associate Professor in the School of Computer Engineering and Director of the Parallel and Distributed Computing Centre. Previously, he was a Senior Lecturer in Computer Science at Exeter University (UK). He received his MA in Mathematics and Computer Science from Cambridge University (UK) and his MSc and PhD in Computer Science from

Manchester University (UK). His current research interests include: parallel and distributed simulation, distributed virtual environments, grid computing and multi-agent systems. His email address is [assjturner@ntu.edu.sg](mailto:assjturner@ntu.edu.sg).

**WENTONG CAI** is currently an associate professor and Head of Software System Division at School of Computer Engineering (SCE), Nanyang Technological University (Singapore). He received his B.Sc. in Computer Science from Nankai University (P. R. China) and Ph.D. also in Computer Science from University of Exeter (U.K.). He was a Post-doctoral Research Fellow at Queen's University (Canada) from Feb 1991 to Jan 1993, and joined SCE as a lecturer in Feb 1993. Dr. Cai is a member of IEEE and his current research interests are mainly in the areas of parallel and distributed computing, particularly, Parallel & Distributed Simulation and Grid Computing. His email address is [aswtcai@ntu.edu.sg](mailto:aswtcai@ntu.edu.sg).

# USE-DRIVEN PRODUCT CONCEPTUALIZATION BASED ON NUCLEUS MODELING AND SIMULATION WITH SCENARIOS

Wilfred van der Vegte  
Imre Horváth  
Faculty of Industrial Design Engineering  
Delft University of Technology  
Landbergstraat 15  
2628 CE Delft, The Netherlands  
E-mail: {w.f.vandervegte, i.horvath}@io.tudelft.nl

## 1 KEYWORDS

conceptual product design, use process

## 2 ABSTRACT

Conventionally, simulation of product behaviour is employed as a pre-realization type of assessment at the end of the design process, making only late feedback for improvement possible. Enabling the start of optimization in the conceptualization is expected to have significant influence on design efficiency. However, the available information at that stage is uncertain, incomplete, multifold and imprecise, which calls for new simulation techniques. This paper proposes nucleus-based modelling and simulation as a solution. A nucleus is a modelling entity to capture the relationships between the lowest level metric elements of the product and to represent the physical effects governing the behaviour of the product. Tolerating uncertainty, incompleteness, modality and imprecision, a nucleus-based model is able to provide an integral model of the actors of the use process. Simulations are controlled by so-called scenarios that arrange a logical structure of feasible situations for the integral model. The paper describes the content of the nucleus-based integral model and presents an application case study to illustrate the potentials of this new approach.

## 3 INTRODUCTION

In the design process, simulations facilitate the anticipation of what happens with products during their life cycle, a crucial part of which is the use stage. However, typically, simulations do not offer a complete picture of the mutual interaction between a product, its user or users and its environment during use. Predicting operation from this broader perspective is considered to be beneficial especially in conceptual design. Regarding anticipation of use, concrete problems with the available geometry-oriented modelling environments and simulation tools boil down to the problem that current simulation packages cannot cope with interventions that naturally occur in use processes.

Furthermore, the pre-realization type of assessment during detail design for which simulation methods and techniques are typically intended, assumes that the main design process has been completed and the product model is

available in testable form. The results of the simulation are used to correct the model and to provide confirmative feedback to the designers at a late stage of product design, when the changes are costly and time-consuming. The major problem with the approach is the late feedback and the lack of in-process optimization of the functionality. Current efforts are towards starting the optimization of a product in the conceptualization, which is the design phase that has the most significant influence on the incurred costs and the value of the product. The information in the stage of conceptualization is however uncertain, incomplete, multifold and imprecise, which calls for new techniques in simulation of the behaviour.

To consider aspects such as product use in the conceptualization, new modelling and simulation approaches are needed. This paper proposes nucleus-based modelling and simulation as a solution. A nucleus is a modelling entity to capture the relationships between the lowest level metric elements of the product and to represent the physical effects that are governing the observable behaviour of the product. Tolerating uncertainty, incompleteness, modality and imprecision, a nucleus-based model is able to provide an integral model of the actors of the use process, that of the user ( $U$ ), the product ( $P$ ) and the environment ( $E$ ). The time history of the relationships implies elementary processes that are the basis of behavioural simulation. The simulation processes are controlled by so-called scenarios that prescribe typical use situations and arrange a logical structure of feasible situations for all elements included in the integral model. The paper describes the content of the nucleus-based integral model. We present the methodology that enables us to generate resource-integrated models and scenarios to deal with the use of products, and provide a template to specify the content of the models as well as a procedure to apply the methodology in conceptual design. The hypothesis is that by providing a homogenous representation for  $U$ ,  $P$  and  $E$ , a comprehensive model can be developed that allows not only modelling and simulating known use processes in various situations, but also predicting use processes in ad-hoc situations. Based on the investigation of the models, in particular of the forecasted behaviour, designers can improve products for use by devising the most appropriate design concepts and configurations. The validity of this hypothesis has been explored by performing tabletop research. A use-oriented conceptual model has

been realized in a commercially available system as a test-bed.

#### 4 STATE OF THE ART IN USE FORECASTING AND SIMULATION

Earlier, the authors presented a survey on the consideration of the use of products in computer-aided conceptual design (Van der Vegte and Horváth 2002). Highlighting the most important definitions and presenting the findings about the state of the art, this survey can provide the reader with additional relevant facts. Below we restrict ourselves to the core problems of modelling and simulating use processes in the course of product conceptualization and early simulation. The use of products can be defined as ‘employment or application to a purpose’ or more specifically ‘direct handling of technical aids to achieve a particular goal’, implying for the product working in service of, and having contact with the human body and the brain. Use is an interaction between the three actors, *U*, *P* and *E*, involving mutual exchange of matter, energy and information.

In approaches for simulating behaviour of the three actors in the use process, we can distinguish artefact simulation techniques for the behaviour of products and environments and human simulation techniques for the behaviour of users. In both areas we distinguish three categories: simulations based on equations or purely mathematical models, simulations based on discretized system representations and simulations based on artificial-intelligence (AI) techniques.

##### 4.1 Artefact-behaviour simulation: simulating behaviour of the product and the environment

In artefact-analysis models, the conventional approach to simulation is to devise a set of symbolic equations specifying a particular situation or a class of situations (Bryant et al. 2001). By solving the equations analytically in the time domain, the course of a process can be predicted. One frequent reason for unavailability of analytical solutions is complexity. Products, environments, and product-environment systems are usually complex and therefore difficult to simulate, even after idealization. A mathematical consequence of increasing complexity is that nonlinearities in the system behaviour can no longer be neglected. Research efforts are increasingly directed towards enhanced simulation techniques that can deal with non-linearity. With the increasing power of computers, *numerical methods* have gained popularity. The most straightforward numerical methods are typically purely mathematical recipes for solving particular types of ‘difficult’ equations (Riley et al. 1997).

Other numerical techniques do not predict the course of a process by solving equations for an idealized system, but based on a *discretized* representation of the system. Discretization takes place by building up artefacts from stereotypical solution elements. The elements carry knowledge about a certain behaviour. Usually, the behaviour knowledge is a linearized simplification of the actual physical behaviour. Some widely applied simulation techniques based on discretization are bond graphs (Red-

field & Krishnan 1992; Finger et al. 2001; Zeid & Overholt 1995), finite-element modelling (FEM) (Zienkiewicz & Taylor 2000; Bailey et al. 1998) and mass-spring modelling (Terzopoulos et al. 1987; Baraff & Witkin 1998; Jansson & Vergeest 2000).

The third approach to artefact-behaviour simulation lies in the application of AI techniques. Unlike the numerical techniques, most AI-based techniques are not yet widely applied in design. Part of the behaviour is controlled by rules stored in knowledge bases, making *qualitative* simulations possible as well. A well-known example is the application of qualitative reasoning (Forbus 1984). Other common AI concepts applied in artefact-behaviour modelling are agents (Mah et al. 1994), neural networks (Masini et al. 1999) and ontologies (Horváth et al. 1998).

##### 4.2 Human-behaviour simulation

Simulation techniques for human behaviour can be subdivided into the same categories that we identified in artefact simulation: (1) simulations based on equations or purely mathematical models, which are usually case-specific (Therrien & Bourassa, 1982); (2) simulations based on discretized system representations, such as FEM models (Koch et al. 1998), bond graph models (Pop et al. 1999) and mass-spring models (Porcher Nedel & Thalmann 2000) and (3) simulations based on AI techniques, such as neural networks (Martens 1998) and agents (Badler et al., 1993). We found that the simulation approaches could best be characterized by the aspects of human behaviour they cover, subdividing human acting into the behaviour types *perceptual*, *cognitive*, *control*, *active physical* and *passive physical* (note: *passive behaviour* means that a body is deformed or moved by external impact only; *active behaviour* means that a body is deformed or moved by internal muscular activity). The result of this characterization is shown in Table 1.

Table 1: Coverage of Human-Behaviour Types by Simulation Approaches from Investigated Literature

		perceptual behaviour	cognitive behaviour	control behaviour	active physical behaviour	passive physical behaviour
Equation-based simulations		X			X	X
Simulations based on structural / numerical models	bond graphs				X	X
	finite-element models				X	X
	mass-spring models				X	
AI-based simulations	neural network-based models		X	X	X	
	agent-based models	X	X	X	X	

##### 4.3 Applicability in use-process prediction

The review of actor-simulating techniques made it clear that a broad range of behaviours determining the interaction between the user, the product and the environment is covered by existing simulations, but there is no technique that covers all relevant aspects. Thus, a valid question would be, if integrating all those simulations into an overall use-process simulation technique can be the most auspicious way to realize use-process forecasting. After all, there are obvious tendencies towards more integrated

forms of simulation already, for instance multiphysics (Mahoney 2000). However, if we want to integrate simulation techniques, we need to take care of the models first, because all simulations are imposed on models of the actors, and the problem with these models is that they typically focus on a specific aspect. From section 5, this issue will be investigated more specifically.

Apart from the modelling issue the drawbacks of commonly used simulation techniques are: (1) the simulations are orientated towards the behaviour of artefacts ( $P$  and  $E$ ), but if  $P$  and  $E$  appear together with  $U$ , they typically only include passive behaviour of  $U$ ; (2) unlike phenomena describing the pure physical behaviour of  $P$  and  $E$ , phenomena that rule the active behaviour of  $U$  cannot straightforwardly be embedded in geometry; (3) associating a product  $P$  with different  $U$ s and different  $E$ s is not supported; (4) simulations tend to be restricted to behaviour that is completely determined by one initial state. The bottle-neck appears to be with simulating humans. Where simulation techniques represent active human behaviour, they do so through deterministic algorithms. In reality, the active behaviour of humans is controlled by mental processes, which make it non-deterministic.

A related weakness of simulation techniques is that they cannot handle multiple scenarios that have to be dealt with because of (1) the possible multiple outcomes of non-deterministic human behaviour, (2) multiple users, and (3) multiple environments. It does not seem feasible to consider all the possibilities but for many products, a considerable amount of such knowledge can be gathered, for instance, from historical data (from existing products). In (Van der Vegte et al. 2002), we presented an approach to handle this knowledge and make it available in conceptual design. From section 8, we elaborate on handling scenarios and the subsequent application of simulation techniques to investigate the more or less deterministic behaviour.

## 5 MODELLING ISSUES

The expansion of CAD/E systems to conceptual design introduces problems in terms of the modelling entities. When we take into account the modelling approaches that follow the mental processes and the thinking of designers, and, in addition, reflect the way the majority of designers would prefer to enjoy computer support, the current solutions are far from optimal. Just consider, whatever it involves, computer support of conceptual design. The overwhelming majority of the currently used systems have been developed to support detailed design and downstream application oriented modelling with geometry in the centre, to enable analyses and simulations. Research systems offer specific approaches to specific problems of conceptual design based on dedicated theories, but they are typically not connected to, and difficult to integrate with, the above mentioned systems due to the high level of abstractions in the models. Although many researchers believe it is totally in line with the nature of conceptual design, other solutions can also be thought of. Actually, this is the primary objective of the nucleus-based approach presented here. With computer aided

conceptual design in the centre, we sketch up a new way of thinking about modelling, which lends itself to a more evocative formation to models, following the way of thinking of designers.

The requested increase in the capabilities of CAD systems assumes ‘smarter’ modelling entities to be shared in modelling, analysis and simulation. The focus of our research into new modelling entities is on conceptual design. Conceptual design works with design concepts that are typically abstract, incomplete and vague. Detail design is for a comprehensive specification of the geometric features and mechanical attributes of the parts and the assembly. Whilst early behavioural simulations provide information about the expected behaviour mainly by qualitative reasoning, advanced behavioural simulations are to qualitatively investigate the behaviour of a product and of the components of it in both the space and time domains.

On the level of functional and methodical requirements, we envisage computer-aided conceptual design (CACD) systems to have the capabilities to handle incompleteness, vagueness and impreciseness of models and information, to be able to provide fast simulations of the physical behaviour of the product during conceptual design, involving the related humans and the environment and to support in-process physical modelling. The CACD systems fulfilling these requirements will operate as front ends of the conventional CAD/E systems, facilitating detail design and numerical analysis of parts, assembly design and behavioural simulation of products.

Application feature modelling is the current paradigm for detailed geometry, assembly and manufacturing modelling as well as for downstream activities (Noort 2002). The major shortcoming with respect to behavioural simulation is that feature technology is confined to handling permanence rather than changes. Practically each natural and artificial system is of a transitory nature that manifests in observable behaviour that is realized by the interactions of function carriers of different mechanical components. Conventional feature representations are application dependent and intend to capture morphological aspects rather than the semantics of functions and the manifestations of operation/behaviour.

Being aware of the potential of feature technology, the objective of our research has been to find possible answers to questions such as: What modelling entity concept comes in product modelling when the feature paradigm is exhausted? What information and/or knowledge have to be conveyed by these entities in order to be able to support conceptual modelling/simulation and detail modelling/simulation equally well? In the next section we propose the nucleus theory as a basis of next generation product modelling, explaining the innovative concept and showing that it results in a family of modelling entities that dramatically extends the functionality of current feature entities. The major difference relative to feature-based modelling is that the notion of geometric entities as fundamental building blocks is abandoned in favour to relations that actually govern the formation of geometry.

## 6 PRESENTING THE NUCLEUS AS A NEW MODELLING ENTITY

It is presumed that any new modelling entities should support feature-based design and processing, i.e., it has to support feature technology in general. In addition, the introduction of some new modelling entities should lead to knowledge-intensive conceptual models offering new functionalities for the designers to conceptualise products. We hypothesized that a new modelling entity has to focus on design concepts that are intuitively or systematically generated by the designers and to make it possible to represent their elements and entirety. It implies the need for a deeper understanding of the nature of design concepts and the possible ways of formalization without destroying creative power. It is especially important with respect to the inherent intuitiveness, incompleteness and uncertainty of design concepts and the heuristic nature of conceptualisation. Obviously, the modelling entities have to be of a very high level (or complex) to be capable to incorporate sufficient amount of knowledge for concurrent modelling of components, assemblies and systems. It amounts to saying that the current systems are somewhat limited in these capabilities.

We developed the nucleus theory as a foundational theory of a new product modelling methodology, and studied the feasibility and applicability. Below we explain the fundamental concepts and clarify the specific notions. From investigation of various engineering products we found that they all can ultimately be decomposed to a purposeful composition of physically coupled pairs. Any physically coupled pair can be abstracted as a composition of - typically two - interacting objects and multiple physical relations between the objects that may appear in various situations. Actually, this abstract construct gave the idea of the nucleus, which is understood as a generic modelling pattern that can be specialized to describe the constituents of a design concept or its entirety. From a programming point of view, the nucleus is a complex data and relation structure that covers geometric, structural, morphological, material and physical aspects. From a modelling point of view, this is the lowest level entity that carries both morphological and functional information to applications through the embedded structure of objects, relations and conditions.

## 7 FORMALIZATION OF THE NUCLEUS-BASED MODEL

As mentioned above, our intention has been to represent design concepts by a purposeful set and configuration of nuclei. With symbolic terms, we formalized a design concept as  $DC = \{O, \phi, S, C, A, D, P\}$ , where  $O = \{(o_i, o_j)\}$  the set of pairs of objects,  $A$  = attributes of objects,  $\phi$  = physical relations,  $P$  = parameters describing the relations,  $S$  = situation in space and time,  $D$  = descriptors of situation,  $C$  = constraints on attributes, parameters and descriptors. Design concepts can be decomposed but not beyond any limit. If the objects, relations and situations are missing, the abstraction becomes meaningless. Actu-

ally, this is another reason to call the  $N = \{O, \phi, S\}$  triplet the nucleus of a design concept (Figure 1). A semantics driven decomposition of design concepts results in nuclei that represent ultimate constituents. Representation of a most elementary design concept requires at least one nucleus. Compound design concepts however need a purposeful composition of a finite number of nuclei. A situation arranges the objects in a set of relations, or, in other words, creates a given structure of elementary processes described by the mathematical formulas. A situated nucleus lends itself to computable behaviour, that is, to temporal changes in the parameter values as governed by mathematical formulas and constraints.

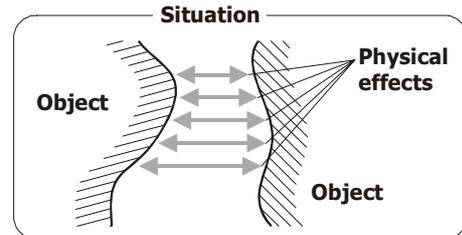


Figure 1: Ontological Conceptualisation of a Nucleus

The objects incorporated in a nucleus are metric entities, which are characterized for their shape and volume. The shape of the objects is represented by half spaces (HS). Actually, a region of these infinite half spaces is used in model building. The finite regions correspond to the natural surface patches of a mechanical part of a product, and lend themselves to effect carrying surface patches. Some of the effect carrying patches will be in contact with surface patches of other mechanical parts. The surface patches are positioned in the model by reference points and may have multiple other reference points for the physical relations assigned to them. For the reason that the geometry of these surface patches is always defined by the geometry of the describing half spaces, in the further discussion we replace the abstract objects in a nucleus with half spaces. Thus,  $N = (HS_n, HS_c, \phi, S)$ , where  $HS_n$  is called a native half space,  $HS_c$  is called a complement half space, and  $\phi$  and  $S$  are as above. A half space indicates the material domain of an object. Native half space is the term used to identify those half spaces that jointly define the boundary of a mechanical part. Complement half spaces are half spaces defining the boundary of other mechanical parts that are in logical, geometric, positional or physical relation with some native half spaces of a particular mechanical part. Our interpretation allows an object to exist in the nucleus without half space definition. In this case the object is logically identified, but geometrically not specified. This is a substantial assumption that enables incomplete modelling in conceptual design. If the half spaces included in a nucleus are geometrically specified, explicit and implicit analytic surface patches, finite parametric surface patches, or finite discrete point or particle clouds can be used as representations. From the aspects of physical modelling, an arbitrary number of relations can be specified between the pairs of half spaces. For a nucleus to operate, at least one half space must be geometrically specified, but, in

this case, only reflexive physical relations can be assigned. Represented by half spaces, the objects acting as ‘environment’ must have at least one reflexive relation to result in a non-limitless system.

The physical relations imply processes that boil down to the behaviour of a nucleus, or a design concept. Actually, the time-dependent changes described by the physical relations will lend themselves to some observable operation, or behaviour, of a nucleus,  $B$ , in some situations:  $B(N) = \Gamma \{S_k (o_i \phi_{ij} o_j)\}$ , where  $o_i, o_j \in O$ ,  $\phi_{ij}$  and  $S_k$  are as above, and  $\Gamma$  is a behaviour generator function, which takes into consideration the interaction of various nuclei and the influences on each other’s behaviour. The introduction of  $\Gamma$  is necessary, since the observable operation of a modelled design concept, DC, is an aggregation of the elementary operations of the nuclei. For the reason that all nuclei might interact in a composition, this aggregation can be represented as a Descartian product rather than as a Boolean union of the observable elementary operations, that is,  $B(DC) = B(N_i) \times B(N_j)$ , or  $B(DC) = \Pi (B(N_i), B(N_j))$ , where  $\Pi$  denotes a mathematical product. The arrangement of situations, or in other words, the operation and interaction of the nuclei, are governed by so called scenarios. A scenario,  $\Sigma$ , prescribes a sequence of situations, in which the observable operation delivered by a nucleus or a configuration of nuclei incorporated in a design concept happens. That is,  $\Sigma = \cup (S_k)$ . With these, the behaviour of a DC is:  $B(DC) = \Gamma (\Sigma \{N_i\})$ , or, on the level of relations,  $B(DC) = \Gamma (\cup (S_k (o_i \phi_{ij} o_j)))$ . Specification of the physical relations includes definition of the parameters, the mathematical formulas (equations and rules) that relate the parameters to each other, and the constraints and value domains. Thus, a nucleus is a primitive system in itself, since its data structure contains all pieces of information that is needed to simulate its behaviour. Based on the above terminology, we call our approach a nucleus-based conceptual modelling of engineering products. At this point we might revisit our previous observation, namely, that engineering products can be modelled in terms of physically coupled pairs (PCP) (Roth 1982). We may say that a PCP is a concrete manifestation of a nucleus, which is able to operate in situations. Examples for such PCP in different situations due to the different arrangement of the objects and the manifestation of physical effects are shown in Figure 2.

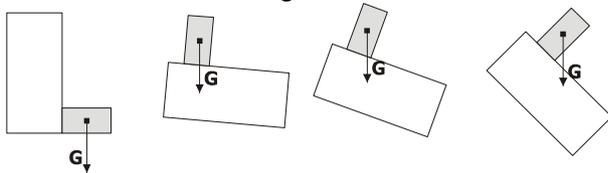


Figure 2: Examples of Situations for PCPs, From Left to Right: Falling; Not Sliding (Static Friction); Sliding; Turning Over

In simple words, relations express the ways in which objects can stand with regard to one another or themselves. Let  $O$  be a set of objects and  $\phi$  a set of relations. The domain of  $\phi$  is the set of objects  $o_1 \dots o_n \in O$  for which

there is at least one  $o_i$  such that  $\phi_i \in \phi$  holds. The converse domain of  $\phi$  is the set of entities  $o_1 \dots o_n \in O$  for which there is at least one  $o_i$  such that  $\phi_i \in \phi$  holds. The logical sum of the domain and the converse domain is the field of relations  $\phi$ . A universal relation contains both  $o_i$  and  $o_j$  as arguments. A universal relation is symmetric if  $o_i \phi o_j$  and  $o_j \phi o_i$  hold. A set of reflexive relations contains  $o_i$  as argument such that  $o_i \phi o_i$ . The square of a set of relations  $\phi$  is  $\phi | \phi$ . A set of relations is transitive if each relation contains its square, that is, if  $o_i \phi o_j$  and  $o_j \phi o_k$  hold, then  $o_i \phi o_k$ . A relation can be seen as a special sort of objects that connects other objects but is numerically distinct and ontologically independent from the connected objects. If  $o_i$  stands in relation  $\phi$  to  $o_j$ , but neither its identity nor its nature depends upon  $o_j$ , the relation is external. If the opposite is true, then  $\phi$  is internal. There are two dimensions of thinking about relations. The first one is the context of the relations; the second is the kind of relations. Various types of relations can be considered in various contexts. As contexts of specification of relations we identified mechanical part, assembly and system design (Figure 3). A mechanical part level relation exists in between pairs of native half spaces; therefore, it is called internal relation. If it brings two close neighbour (intersecting) half spaces in spatial relationship, then it is called direct internal relation (DIR). If it concerns two far neighbour half spaces, then it is an indirect internal relation (IDIR). A mechanical assembly relation exists between one-one native half spaces of two mechanical parts, which represent a native-complement construct. The assembly relations are called external relations, and based on the analogy of internal relations, they can also be direct (in contact, DER) or indirect (not in direct contact, IDER). Finally, system-level relations describe interactions with elements of the nuclei representing the physical environments. System level relations offer themselves to the representation of, for example, product-user-environment configurations, as it will be shown from section 8.

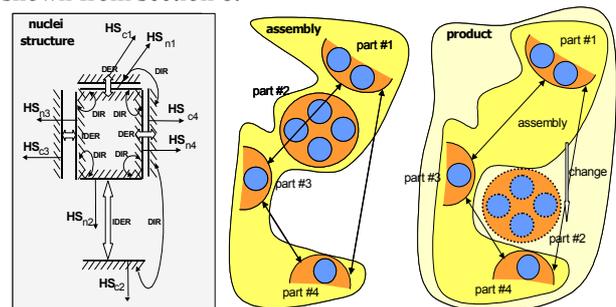


Figure 3: Relations on Mechanical Part, Assembly and System Levels

The type of relations depends on the semantics of the relations. We introduced (a) ontological, (b) connectivity, (c) morphological, (d) positional, and (e) physical relations (Horváth et al. 1998). An ontological relation indicates the existence of an object or any higher-level construct; therefore, it is reflexive. Connectivity relations define the topography of relations between objects and, as discussed above, they are used to define either me-

chanical parts or assemblies. Reflexive morphological relations define the geometry of the half space describing the metric of an object. Associating morphological relations define the relationship between two half spaces of different objects. Positional relations specify the rotations and translations between the half spaces of a nucleus or any two higher-level constructs. Finally, physical relations formulate physics-based relationships between half spaces of a nucleus to transfer physical effects. They can be reflexive (such as mass) or non-reflexive (such as a force). The relations are described by means of parameters and mathematical formula. The geometric aspect and the effect aspect are brought into synergy through reference points or spots. Based on the nucleus concept, a conceptual modelling system is able to know about and manage a complementing object when a native object is defined. The system is also able to automatically apply all of the default relations for any pair of objects and to let the designer activate only the necessary ones. Based on activating an internal relationship, the system can be aware of the fact that a mechanical part is being formed, and activating an external relationship means that an assembly is generated.

The system can not only monitor these steps of conceptualisation, but also can control the processes and check for validity, completeness and consistence. In system programming, the nucleus concept lends itself to the internal modelling scheme of a CACD system. In fact, it is observable only in the prevailing modelling methodology that focuses on the relations and handling the changes in the relations of objects in various situations. Activation of a nucleus offers a generic modelling entity for the designer that can be further specified according to the design concepts to be applied to solve the design problem. Should a nucleus be activated, the designer is given a set of relations that are specified in terms of attributes, parameters and descriptors. In principle, an infinite number of relations can be specified between two objects, but in practice only those will be instantiated that are important for a given modelling or simulation task (Kitamura et al. 2002).

Parameters representing flow quantities and cross quantities are referred to specific points on the half spaces, which are called ports. In the case of an incomplete part or assembly model, indication of the integrity is a remarkable problem. As a simple solution, fictitious connection lines are generated and visualized between the reference points of the half spaces being in internal positional relations. This leads us to a physically based skeleton model, which is one of the alternative realizations of the nucleus concept as a practical modelling methodology. Naturally, designers do not face these abstract concepts and terms when they are using a nucleus-based system in conceptual design. The design concepts are expressed in terms of an arrangement of nuclei, e.g., in application features, which are represented as functionally related surface patches in given situations. A nucleus can be placed into different situations, which means instantiation of the interacting processes in different forms. Not only complex design concepts, but also design features

can be defined in the same manner and used to express design concepts in a semantics-intensive way. Solid mechanics offers the means to treat the four main observable phenomena: motion, collision, deformation and fracture. Phenomena relating thermodynamics, fluid dynamics, gas dynamics, and so forth can also be considered in relations. It is a fact however that there exists no single predictive model that is capable to incorporate all phenomena and interrelated changes, not even theoretically.

## 8 MODELLING AND SIMULATING PRODUCT USE

In the workflow for modelling and simulating product use, three basic activities are involved: (1) modelling the actor triplet  $U$ ,  $P$  and  $E$ , (2) modelling a scenario and (3) performing a simulation.

To model the actor triplet, instantiations of nuclei serve as building blocks for the actors  $U$ ,  $P$  and  $E$ . The fact that nuclei can represent the physical characteristics of the actors in addition to their geometric and structural characteristics makes them attractive for modelling and simulation of use cases.

A scenario is an arrangement of situations that can be used as input for simulations,  $\Sigma = \cup (S_k)$ . A situation is a state of the actors that allows the description of different circumstances. By describing a particular configuration of the actors  $U$ ,  $P$  and  $E$ , each situation defines the physical processes to be simulated as well as the initial state of the system, from which simulation algorithms can calculate the course of physical processes. In simulation, the scenarios are the formalized means to treat the circumstances. They can be seen as a kind of program: the simulation engine works according to the control that comes from the scenario. The scenario connects to the mental part of the designer as a means to formalize happenings the designer expects in terms of the three actors. In this context, the scenario serves as a formalization of the design intent: it is a connection between the designer and the triplet that allows the designer to play with the simulated triplet. At the same time, scenarios allow the designer to include the effects of mental processes of the user that cannot be covered by the deterministic algorithms of common simulation engines. In that context, the scenario can be seen as a use pattern for a product.

To overcome the problem that simulations cannot cope with multiple use processes, produced by a multitude of users, user behaviours, and environments, scenarios can also be applied to generate and manage multiple simulations.

## 9 INFORMATION CONTENT OF THE PRODUCT-USE MODEL

The conceptual model consists of (1) an object-type model based on the nucleus principle and (2) scenarios. The nucleus model also incorporates a relevant set of relations that make it possible to define the associations between  $U$ ,  $P$  and  $E$  and simulate the use through a series of situations. To make the abstract concepts more tangible, we arrange the discussion around a practical exam-

ple.

## 9.1 The actor triplet

Figure 4 shows a schematic representation of a simple  $U$ - $P$ - $E$  model. The illustrative ‘product’,  $P$ , is a foot-operated lever that can be used to lift objects. The objects that are not part of the product are considered to be part of the environment  $E$ . As the figure shows, the interactions between the actors and between parts of actors take place on the specified regions of the contact surfaces, which are represented by finite surface patches.

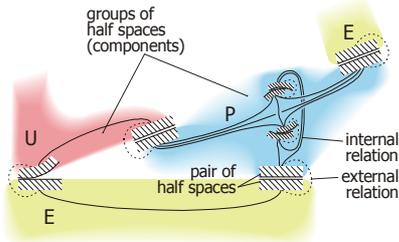


Figure 4: Nucleus-based  $U$ - $P$ - $E$  Model with Internal and External Relations

Point-oriented relations are assigned to the reference points of these surface patches. The surface patches on the half-spaces belonging to the same part of the same actor are connected through internal relationships. Figure 5 shows the topography of the relation structure between patches in an abstract form. With the edges associated as they are representing internal relationships (forming a component) or external relationships (forming an assembly), it serves as a conceptual scheme to organize the computer-internal database.

## 9.2 Scenarios

It is important to note that, while Figure 5 represents a generic situation, Figure 4 concerns a particular situation. A particular situation assumes a given configuration of the contact surface patches and a given manifestation of the physical effects in the presented situation: the foot presses the lever, friction and gravity impede the rotation of the lever and the lever takes a definite spatial position. Other situations could be when the foot releases the lever upward, or situations in which the object or the foot is absent, or in which they are swapped. A *scenario* contains at least one situation, for there is at least one initial state from which the physical processes can be launched. Other states that cannot straightforwardly be derived from these processes, i.e., not from the associated simulation, must be defined in other situations within the scenario. A typical use scenario for this lever would consist of a series of situations that can be qualitatively described as: (1) no foot present, the right end of the lever is down and there is an object placed on it, (2) the foot pushes the lever to lift the object, (3) the object is removed at a certain height and the foot releases the lever to make the right of the lever end come down. Note that the situations that are introduced in (2) and (3) depend on decisions from the human user, and that they cannot be calculated by a deterministic simulation algorithm starting from (1).

Elaborating the information content of scenarios, we will take a closer look at situations first. Practically, situations define how and where  $U$ ,  $P$  and  $E$  interface/interact with each other, and which initial configuration the individual parts of  $U$ ,  $P$  and  $E$  are supposed to be in. In case of the user, this configuration refers to the posture that is governed by degrees of freedom of the joints and the skeleton.  $P$  and  $E$  can also be assumed to be in various configurations based on degrees of freedom, which always implies different situations. These configurations, or degrees of freedom, typically appear as *simulation parameters*, i.e., parameters that are variable within a simulation. They have to be distinguished from *design parameters*, i.e., parameters that can be chosen by the designer. For instance, the mass of a component is typically a design parameter because the designer can change it, but it normally does not change during a physical process in use. Conversely, an arbitrary intermediate angle that the lever in Figure 5 can assume is a typical simulation parameter: it does not make sense to define it as a design specification. On the other hand, the maximum and minimum angles for the lever can be considered both design parameters and simulation parameters.

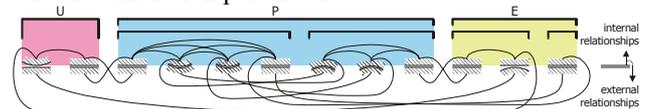


Figure 5: Internal Representation of the Nucleus-based  $U$ - $P$ - $E$  Model

Through the situations, the scenario imposes simulation parameters over the modeled environment. We can distinguish input, throughput and output simulation parameters. The throughput and output parameters receive values by inheritance from the input parameters with the contribution of the design parameters. The output reflects the change caused by whatever happens in the system. The connection between the various situations that a scenario feeds to the simulation is made through conditions described in terms of variables. If a certain condition occurs during the simulation, the scenario prescribes that a new situation comes into effect and the simulation has to continue from there. The condition can be a value of a simulation parameter, or the time elapsed from the latest specified situation.

## 10 APPLICATION EXAMPLE

To investigate the applicability of resource-integrated modelling and simulation in conceptual design, we have developed a nucleus-based model of an existing product, a pedal bin. The level of detailing of the object-type models of  $U$ ,  $P$  and  $E$  corresponds to what we presumed to be appropriate in conceptual design. We generated a qualitative description of a simple use scenario, disposal of a piece of garbage, which specifies the situations and the initial conditions for a simulation. The actual simulation was performed with Working Model<sup>®</sup> 2D (WM2D), a product of MSC Software Corporation (Wang 2001). This package was also used to create most of the nucleus-based models of  $U$ ,  $P$  and  $E$ .

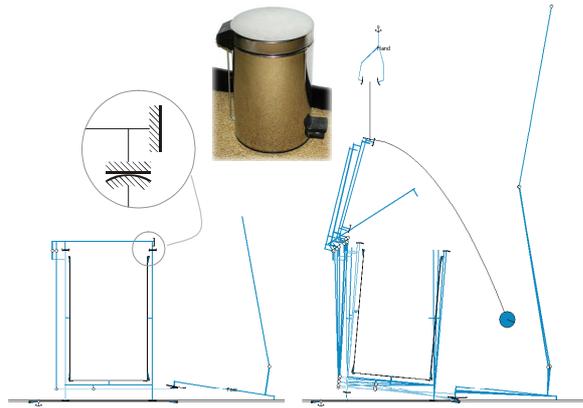


Figure 6. (a) The Nucleus-Based Conceptual Model of the Pedal-Bin (Left);  
(b) the Result of Applying a Use Scenario (Right)

The reason why we chose WM2D is its distinctive capability to support situations that do not only depend on one initial state, but may include predefined interventions afterwards. In case of the pedal-bin, the user's hand can drop the object at any given time, or the time of dropping can depend on the position of the lid. Likewise, the moment when the pedal is released can depend on the position of the dropped object. Many commercial simulation packages cannot directly include such interventions. Figure 6a shows the initial resource-integrated conceptual model of the pedal bin for investigation of use. Only those parts of the user's body have been modelled that are concerned in the use scenario: a hand and a foot (note: in this pilot study, it was not our primary objective to come up with a correct anatomic representation of the human body, or to provide exhaustive forecasts for real-life design process). The model of the product consists of four moving parts, and the environment consists of the floor and the garbage object. The grey rods represent skeleton elements, and the half-spaces are indicated by the black outlines. Note that for graphical reasons, we used the common representation of a dot in a circle to represent joints rather than the half-space representation depicted in Figure 5. Half-spaces are graphically represented at those locations where components interact at  $t=0$ , or where interaction can be conceived during the situations defined in the scenario. The simulation is based on a scenario arranging two situations starting from the following two states: (1) the foot exerts a constant force  $F_1$  on the pedal (i.e., to operate the lid) and (2) if the condition  $t=1s$  is met, the reaction force that keeps the garbage object in the hand is set to  $F_2=0$  (i.e., the object is released), while  $F_1$  remains unchanged.

Figure 6b shows the results of the simulation: when the pedal is pressed, the bin starts to tumble and the lid tends to oscillate around its highest position. If the object is launched from the shown position, this behaviour of the bin prevents proper disposal.

Figure 7 shows an improved conceptual design with a counterweight connected to one side of the lid. It ensures a more determined movement of the bin and the lid. The simulation proved that the object launched at the same

time and from the same location as in Figure 6, now lands successfully inside the bin. As an all-embracing definition of the use scenario, a third situation was added and investigated in the simulation: (3) from  $t=1.5s$ ,  $F_1=0$ , i.e., the foot releases the pedal to close the lid.

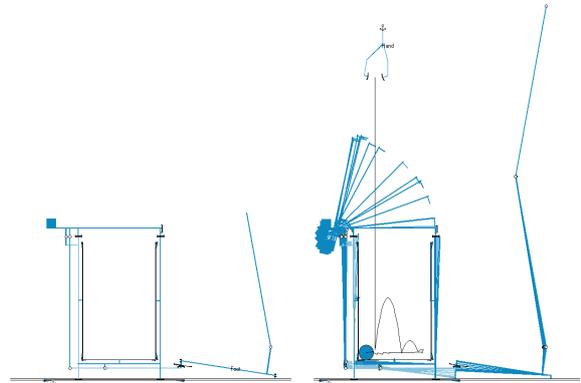


Figure 7: Simulation Result of a Use Scenario with Improved Conceptual Design of the Pedal Bin

## 11 DISCUSSION

Despite the experienced restrictions, the results revealed attractive prospects for the application of resource-integrated models to represent use-processes in conceptual design. Designers can anticipate the use process without having to switch between object-type models and process-type models (including functional models). That is, processes involving simulation-based forecasting can be seamlessly included in the modelling environment, and even intervention-type interactions can be studied.

The proposed nucleus concept offers relation-oriented modelling rather than entity-centred modelling. It places the pairs of objects into a multitude of relations, which are not restricted to be in the same aspect or context. By doing so, it mimics the working of the human mind as it builds associations between neutral entities in a creative conceptualisation. It also tries to resolve the known problem of linking different views or jumping between aspects.

By making the entity relationships more explicit and knowledge intensive, a nucleus-based conceptual design system converts the paradigm of 'doing what you know' to the paradigm of 'knowing what you're doing'. It allows the designers to describe design concepts as an aggregation of nuclei, to define and use application features, to construct parts, assemblies and systems, and to investigate the physical behaviour of all these constructs based on space- and time-dependent evaluation of the specified relations. It involves validity management, consistency management and multi-view management. An obvious advantage of the nucleus concept is that it does not force the designer to define the part geometries first. He may alternate between structure, component and system definition, leaving the geometry to appear as a by-product of the conceptualisation process.

The nucleus concept vindicates that models can be incomplete on part, assembly and system levels. Models can gradually be extended and refined as knowledge becomes available for the designed product. Extension and

refinement may take place in terms of the morphological and physical relations. This way, the evolving model that integrates both artefact/actor representation and process representation adapts to the progress of conceptualisation. This model is referred to as a multi-resolution model.

Current research deals with extensional relations only, and considers them as n-ary relations that can be traced back to dyadic relations. The used prepositional functions do not extend to intentional relations.

Note that we still face some sort of 'metaphysical' limitations in terms of being able to define any ideal modelling entity for the reason that an exact scientific understanding related to the following issues is still missing: (a) mapping requirements onto a system of functions or potential operations, (b) mapping target functions to first principles and physical processes, (c) mapping functions or structures to forms and embodiments, (d) deriving structures from first principles and physical phenomena, and (e) identification of the necessary constituents from physical processes.

Our application case study, on the one hand, demonstrated the method and issues of using nucleus-based modelling in conceptual design. On the other hand it made it possible for us to see the advantages and disadvantages with respect to the given application case.

The homogenous representation of  $U$ ,  $P$  and  $E$  utilized in the use-oriented modelling and simulation of the concept product, i.e., the pedal bin, enabled us (1) to model the known use processes in the form of common scenarios and (2) to predict ad-hoc use processes based on simulations.

As a result of the investigations and, in particular, of forecasting the behaviour, an improved concept product could be realized on the level of detail that is typical for conceptual design. Thus, our hypothesis seems to be proven at least for the presented application example. It is likely that the same can be claimed for products of a similar (low) complexity and of resembling use processes. Nevertheless, we have to validate it for a wider range of products and use processes. In this respect it has to be made clear that in terms of an exhaustive validation of the hypothesis, the set-up of this tabletop research was inherently limited by the capabilities of the simulation package. WM2D does not support object-type models and processes of high complexity, since (1) it cannot deal with three-dimensional representations, (2) it has difficulties in dealing with statically undetermined structures, and (3) it has been developed for rigid-body dynamics. Moreover, it has difficulties in dealing with conceptual modelling entities that do not necessarily correspond to actual geometries with corresponding weight distributions, such as the skeleton elements and of surface patches of half-spaces that we applied. This indicates that there is a need to develop a dedicated simulation environment for resource-integrated models.

## 12 CONCLUSIONS AND FUTURE WORK

For a typical, however simple, application, we have shown that a nucleus-based model that offers a homoge-

nous representation for the product, the user and the environment can support conceptual design of products. This comprehensive, resource-integrated model allows a designer to consider known use processes in various situations, but also to obtain predictions of ad-hoc use processes by means of simulation. The results of these behavioural simulations can be utilized in conceptual design to improve products for use. To make the resource-integrated models and the forecasting of use processes applicable to a wider range of products and use processes, further work is needed in particular in (1) further refinement of the fundamentals and methodology of modelling and simulation based on scenarios prescribing the use of products, and (2) development of a dedicated simulation environment that can benefit from resource-integrated conceptual models. It is expected that a full-featured system can be developed based on these future achievements to assist designers in optimizing products for use in the early stages of development.

## 13 REFERENCES

- Badler N.I, C.B. Phillips, B.W. Webber. 1993. *Simulating humans – computer graphic animations and control*. Oxford University Press, New York.
- Bailey, C., G.A. Taylor, M. Cross, P. Chow. 1999. "Discretisation procedures for multi-physics phenomena." *Journal of Computational and Applied Mathematics* 103, 3-17.
- Baraff, D., Witkin, A. 1998. "Large steps in cloth simulation." *Proceedings of SIGGRAPH*, 43-54.
- Bryant, C.R., M.A. Kurfman, R.B. Stone, D.A. McAdams. 2001. "Creating equation handbooks to model design performance parameters." *Proceedings of ICED*, Glasgow, 501-508.
- Finger, S., X. Chan, R. Lan, B. Cahn. 2001. "Creating virtual prototypes - integrating design and simulation." *Proceedings of ICED*, Glasgow, 485-492.
- Forbus, K.D. 1984. "Qualitative Process Theory." *Artificial Intelligence*, Vol. 24, 85-168.
- Horváth, I., G. Kuczogi, J.S.M. Vergeest. 1998. "Development and application of design concept ontologies for contextual conceptualization." *Proceedings of ASME DETC*, Atlanta.
- Jansson, J., J.S.M. Vergeest. 2000. "A General Mechanics Model for Systems of Deformable Solids." *Proceedings of TMCE*, Delft.
- Kitamura, Y., T. Sano, K. Namba, R. Mizoguchi. 2002. "A functional concept ontology and its application to automatic identification of functional structures." *Advanced Engineering Informatics*, Vol. 16, No. 2, 145-163.
- Koch, R.M., M.H. Gross, A.A. Bosshard. 1998. "Emotion editing using finite elements." *Eurographics*, Vol. 17, No. 3, 295-302.
- Mah, S., T.W. Calvert, W. Havens. 1994. "A constraint-based reasoning framework for behavioural animation." *Computer Graphics Forum*, 13 (3), 315-324.
- Mahoney, D.P. 2000. "Multiphysics analysis." *Computer Graphics World*, 23 (6), 44-46, 50, 52.
- Martens, D. 1998. "Neural networks as a tool for the assessment of human pilot behaviour in wind shear." *Aerospace science and technology*, Vol. 1. 39-48.
- Masini, R., Padovani, E., Ricotti, M.E., Zio, E. 1998. "Dynamic simulation of a steam generator by neural networks." *Nuclear Engineering and Design*, Vol. 187. 197-213.
- Noort, A. 2002. *Multiple-view feature modeling with model adjustment*. Ph.D. Thesis, Delft University of Technology,

Delft.

- Park J. and D.S. Fussell. 1997. "Forward dynamics based realistic animation of rigid bodies." *Computers and Graphics*, Vol. 16, 483-496.
- Pop, C., A. Khajepour, J.P. Huisson, A.E. Patla. 1999. "Application of Bondgraphs to Human Locomotion Modeling." *Proceedings of HKK Conference*, Waterloo, Canada, June, 85-90.
- Porcher Nedel, L., D. Thalmann. 2000. "Anatomic modeling of deformable human bodies." *The Visual Computer*. Vol. 16, 306-321.
- Redfield, R.C., S. Krishnan. 1992. "Towards automated conceptual design of physical dynamic systems." *Journal of engineering design*, Vol. 3, No. 3. 187-204.
- Riley, K.F., M.P. Hobson, S.J. Bence. 1997. *Mathematical methods for physics and engineering*. Cambridge Press, Cambridge.
- Roth, K.-H. 1982. *Konstruieren mit Konstruktionskatalogen*. Springer Verlag, Berlin.
- Terzopoulos, D., J. Platt, A. Barr, K. Fleischer. 1987. "Elastically deformable models." *Computer Graphics*, Vol. 21, No. 4, 205-214.
- Therrien, R.G., P.A. Bourassa. 1982. "Mechanics application to sports equipment: protective helmets, hockey sticks, and jogging shoes." In: *Human body dynamics: impact, occupational and athletic aspects*, Ghista, D.N. (Ed.). Clarendon Press, Oxford.
- Van der Vegte W.F. and I. Horváth. 2002. "Consideration and modeling of use processes in computer-aided conceptual design: a state of the art review." *Transactions of the SDPS*, Vol. 6, No. 2, 25-59.
- Van der Vegte W.F., Y. Kitamura, R. Mizoguchi, I. Horváth. 2002. "Ontology-based modeling of product functionality and use – part 2: considering use and unintended behavior." *Proceedings of EDIProD*, Łagów, 115-124.
- Wang, S.-L. 2001. "Motion simulation with working model 2D and MSC.visualNastran 4D." *Journal of Computing and Information Science in Engineering*. Vol. 1, No. 2., 193-196.
- Zeid, A.A., Overholt, J.L. 1995. "Singularly perturbed formulation: explicit modeling of multibody systems", *Journal of the Franklin Institute*, Vol. 332B No. 1, 21-45.
- Zienkiewicz, O.C, R.L. Taylor. 2000. *The finite element method*. Vol. 1,2,3. Butterworth-Heinemann, Oxford.

## AUTHOR BIOGRAPHIES



Wilfred van der Vegte (1963) received M. Sc. in mechanical engineering (1989) at Twente University and MTD (Master of Technological Design) in industrial design engineering (1992) at Delft University of Technology. He worked as a designer and project manager at the TNO Institute of Industrial Technology from 1991-1998. In 1998 he started in his current position as an assistant professor of Computer Aided Design and Manufacturing at the Faculty of Industrial Design Engineering, Delft University of Technology. His primary research interests are computer-support of use processes in conceptual design and knowledge-intensive support of design. His current educational contributions are in knowledge-intensive support of design and in coaching students in design projects commissioned by companies.



Imre Horváth (1954) received M.Sc. in mechanical engineering (1978) and in engineering education (1980) from the Technical University of Budapest. He was working for the Hungarian Shipyards and Crane Factory between 1978 and 1984. He has had faculty positions at the Technical University of Budapest between 1985 and 1997. He earned a dr. univ. title (1987) and a Ph.D. title (1994) from the TU Budapest, and a C.D.S. title from the Hungarian Academy of Sciences (1993). Since 1997 he is a full professor of Computer Aided Design and Manufacturing at the Faculty of Industrial Design Engineering, Delft University of Technology. He initiated the series of TMCE Symposiums. His primary research interests are in computer support of conceptual design, vague shape modelling, multi-physics-based behavioural simulation, free-form prototyping and ontology-based formalisation of design knowledge. He is member of several editorial boards. As educator he is currently interested in computer application in conceptual design, integrating research into design education, and teleconferencing-based active learning.

# PROMETHEE-i SELECTING THE BEST SIMULATION MODEL CONFIGURATION BASED ON MULTIPLE PERFORMANCE MEASURES

H. Pastijn, F. Van Utterbeeck, and R. Van Loock  
Royal Military Academy  
Renaissance Avenue, 30  
B-1000 Brussels  
Belgium  
E-mail: Hugo.Pastijn@rma.ac.be

## KEYWORDS

Discrete event simulation, system configuration comparison, outranking methods, multicriteria decision making, performance measures.

## ABSTRACT

We introduce the use of a variant of the original multicriteria decision making method Promethee, in order to select the best simulation model configuration among a finite set of alternatives. For each alternative configuration a number of replications of terminating simulation runs is performed. At the end of each replication, and for each configuration, the result of a number of performance measures is obtained. In the selection problem, these performance measures are typically conflicting criteria for which the alternative configurations have been assessed by a number of computer simulation replications. We submit these data to an interval version of the Promethee outranking method, in order to select the best model configuration. We illustrate this by means of an incident management model for a call centre.

## INTRODUCTION

Comparing alternative system configurations based on stochastic computer simulation output has been extensively studied in the literature (Law and Kelton, 1991) (Kleijnen, 1975). More generally, there is an impressive bibliography available about stochastic ordering (Mosler and Scarsini, 1993). However, most of the effort is done when only one performance measure is involved. In reality various performance measures are simultaneously involved in assessing the behaviour of a system. Different system configurations will typically improve some performance measures and deteriorate some other ones. The selection of the best system configuration among a finite set of alternatives assessed for a finite set of criteria (performance measures) is a typical multicriteria decision making problem. For these problems there is a wide variety of methods available nowadays. The original Promethee methods (Brans et al., 1986), belonging to the

outranking category of multicriteria decision making tools, are appropriate to handle these problems in the deterministic case. In order to take into account the stochastic character of the computer simulation runs, these methods have to be adapted (Mareschal, 1986). We propose to use an interval version: Promethee-i (Mareschal and Le Teno, 1992).

First we describe the main features of Promethee-i.

Secondly we describe a simulation model of a call centre which was implemented in ARENA<sup>®</sup> (Kelton et al., 1989). Finally we compare alternative designs of this model based on traditional performance measures: waiting times in queues, productivity, cost and service level. Therefore we run a number of replications of terminating simulation runs and we apply Promethee-i. The selection result is studied as a function of the number of replications of the simulation runs.

## PROMETHEE-i

For the original Promethee method we refer to Brans et al. (1986). This method is based on crisp data assessing the  $n$  alternatives on  $k$  criteria (performance measures).

For each criterion a pairwise comparison (difference of assessment) of alternatives  $a$  and  $b$  is translated by a preference indicator  $P_j(a,b)$  on the interval  $[0,1]$ . These  $P_j(a,b)$  are aggregated over the set of all criteria by :

$\pi(a,b) = \sum_j \omega_j P_j(a,b)$ , with  $\omega_j$  in  $[0,1]$  being the weight of criterion  $j$ . Then are calculated for each alternative  $a$ :

the strength  $\phi^+(a) = (1/(n-1)) \cdot \sum_x \pi(a,x)$ ,

the weakness  $\phi^-(a) = (1/(n-1)) \cdot \sum_x \pi(x,a)$  and

the net dominance  $\phi(a) = \phi^+(a) - \phi^-(a)$ . The best alternative is the one with the highest net dominance.

When we have  $m$  replications of computer runs, then we obtain for each alternative configuration  $m$  assessments for each criterion (performance measure). These  $m$  assessments can be represented for alternative  $a$  by an interval  $[a^l, a^u]$ , which we take here either as the inter-quartile interval, or as the 99.9% confidence interval on the mean (results with both methods will be compared). All the arithmetic of Promethee is now extended, keeping intervals all along the calculations, by means of the following definitions:

$$[a^l, a^u] + [b^l, b^u] = [a^l + b^l, a^u + b^u] \text{ and } [a^l, a^u] - [b^l, b^u] = [a^l - b^u, a^u - b^l].$$

We obtain consecutively

$$P_j(a, b) = [P_j^l(a, b), P_j^u(a, b)],$$

$$\pi(a, b) = [\pi^l(a, b), \pi^u(a, b)],$$

with  $\pi^l(a, b) = \sum_j \omega_j P_j^l(a, b)$  and

$$\pi^u(a, b) = \sum_j \omega_j P_j^u(a, b),$$

$$\phi^+(a) = [\phi^{+l}(a), \phi^{+u}(a)],$$

with  $\phi^{+l}(a) = (1/(n-1)) \cdot \sum_x \pi^l(x, a)$  and

$$\phi^{+u}(a) = (1/(n-1)) \cdot \sum_x \pi^u(x, a),$$

$$\phi^-(a) = [\phi^{-l}(a), \phi^{-u}(a)],$$

with  $\phi^{-l}(a) = (1/(n-1)) \cdot \sum_x \pi^l(x, a)$  and

$$\phi^{-u}(a) = (1/(n-1)) \cdot \sum_x \pi^u(x, a),$$

$$\phi(a) = \phi^+(a) - \phi^-(a) = [\phi^l(a), \phi^u(a)].$$

In addition the original Promethee method is applied by taking into account all the worst bounds of the assessment intervals  $[a^l, a^u]$  and another time by taking all the best bounds of these assessment intervals  $[a^l, a^u]$  for all alternatives on all criteria. This yields for each alternative another interval  $[\phi^l(a), \phi^u(a)]$ .

Finally this Promethee-i procedure is resulting into a trapezoidal fuzzy number  $[\phi^l(a), \phi^l(a), \phi^u(a), \phi^u(a)]$  for each alternative  $a$ .

On these fuzzy numbers we apply the Yager operator  $\Psi$  (Yager, 1981)(Detyniecki et al., 2001), and the best alternative corresponds to the highest value for this Yager operator  $\Psi$ .

## INCIDENT MANAGEMENT MODEL FOR A CALL CENTRE

The simulation model we consider is the incident management process of a call centre. We will describe the elements of the model (the incidents, the resources and the skills matrix) and the process flow (Fig.1). (Van Loock et al., 2003)

The *incidents* are initiated by the customers of the call center. These incidents are represented by the calls that arrive at the centre. These *incoming calls* follow a stochastic arrival pattern. The calls are subdivided into 2 categories and 6 subcategories, depending on the area of expertise required by the customer. Each category has a specified probability of occurrence, while the subcategories within a certain category are assumed to be equiprobable.

The *resources* in our model are the dispatcher(s) and the system engineers. Each resource has its own weekly working schedule, an hourly cost (based on the number of skills known) and a FIFO queue associated with it. Incoming calls will wait in the FIFO queue if the resource is busy. A call will be *rejected* (and leaves the system immediately) if the time spent waiting in a FIFO queue exceeds a certain fixed threshold.

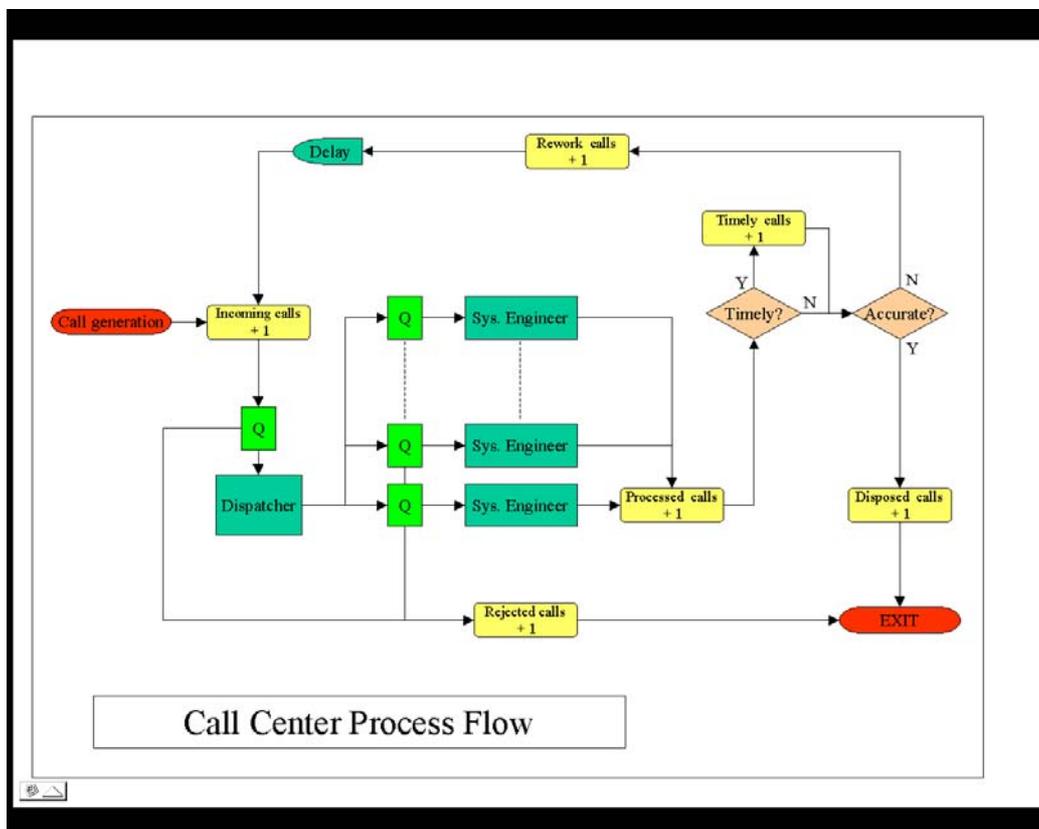


Figure 1: Call Center Process Flow

Every system engineer has his own areas of expertise, which are specified in the *skills matrix*. Every line in the matrix represents a subcategory, while every column represents a system engineer.

The *process flow* used in our model can be summarized as follows. Every incoming call must pass through a dispatcher. The dispatcher will rout the call to a system engineer whose area of expertise covers the category and subcategory of the call. If multiple system engineers are eligible, the dispatcher will rout the call to the resource with the shortest queue. Ties are broken in favour of the resource located the most to the left in the skills matrix. The *dispatching time* (the time needed by the dispatcher to decide on the routing of the call) follows a triangular distribution.

The *processing time* (the time the system engineer needs to handle a call) follows an exponential distribution, regardless of the subcategory. For every *processed call*, there is a fixed probability that the customer is not completely satisfied with the assistance provided. These customers will call back after a stochastic delay. These subsequent *rework calls* will result in a decrease in the performance of the call centre. If the customer is satisfied with the assistance provided, the call is *disposed* and leaves the system.

## PERFORMANCE MEASURES

We use four performance measures: waiting times in queues, resource utilisation or productivity, service level and system cost. Waiting times in queues and resource utilisation are average values obtained from standard ARENA® statistics. Service level is expressed as the percentage of arriving calls which are finally disposed after a successful handling by the available resources (and as a consequence were not ejected from the system). The cost of a system engineer depends on his degree of polyvalence (number of skills). The overall system cost is a stochastic entity due to the fact that the resources continue to work at the end of their daily schedule until all calls waiting in their queue at the end of the working day have been processed.

## SIMULATION

Ten different configurations of the system were simulated, and the observed performance indicators were compared. Reconfiguration of the system is simulated through changes in either the number of resources (additional system engineers or additional dispatchers), or through changes in the skills matrix. More radical changes, like the implementation of a frontoffice-backoffice strategy, were not considered. We restricted the eligible configurations by imposing a maximum allowable weekly cost and a minimum acceptable servicelevel. The OptQuest for Arena software was then used to heuristically identify eligible

scenarios that optimise one of the four selected performance indicators. Finally we selected the four optimal configurations identified by OptQuest, as well as six variations of those as the ten “likely candidates” for our simulation study.

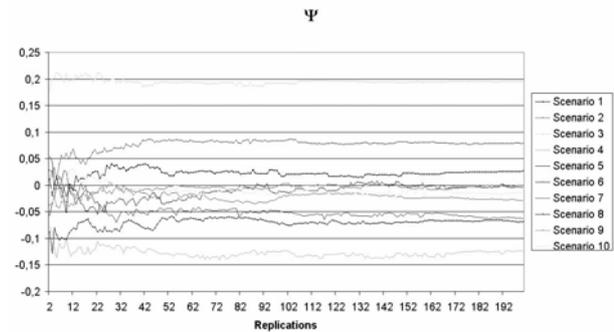


Figure 2:  $\Psi$  Operator Based on Interquartile Intervals

Figure 2 shows the evolution of the  $\Psi$  operator based on the inter-quartile intervals, during 200 replications for the 10 system configurations (scenarios).

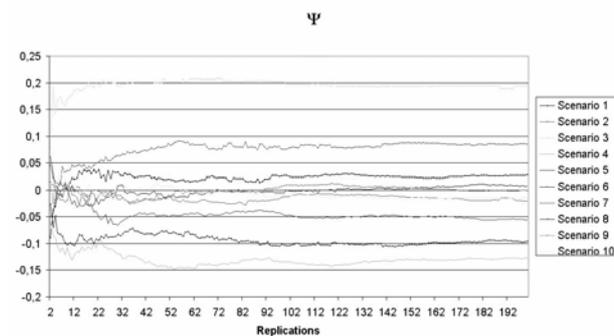


Figure 3:  $\Psi$  Operator Based on Confidence Intervals

Figure 3 shows the evolution of the  $\Psi$  operator based on the 99.9% confidence intervals of the mean performance measures, during 200 replications for the same 10 system configurations (scenarios).

We remind that the 99.9% confidence intervals become of course narrower as the number of replications increases. As a result, this method is expected to converge towards the crisp version of Promethee, by using the average performance measures as an input for the computations at each replication. This is confirmed by figure 4 which is showing the evolution of the net dominance  $\phi$  of Promethee during 200 replications for the 10 system configurations.

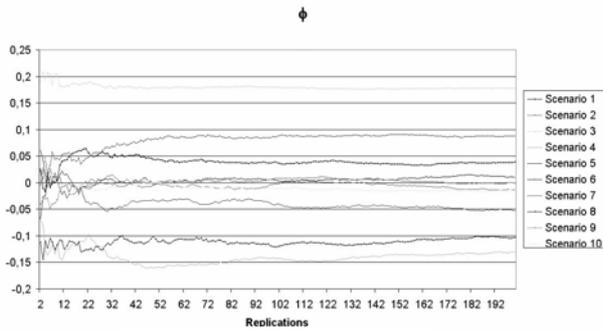


Figure 4: Net Dominance

We notice that the ranking of the alternative system configurations is quite stable for the three methods for a number of replications exceeding about 30 in this experiment. Rank inversions occur even at a high replication number for the standard (crisp) Promethee method and also for the Promethee-i method using 99.9% confidence intervals on the mean performance measures. These ranking inversions occur especially when the  $\Psi$  operator values or the net dominances  $\phi$  are very close to each other. We see however in this experiment that the discrimination in ranking between system configurations which are quite distant in terms of  $\Psi$  operator values and net dominances  $\phi$  after 200 replications, are identified earlier (after fewer replications) when we use the interquartile intervals in the Promethee-i method. This is clearly illustrated by a zoom-in on the evolution of the  $\Psi$  operator value and net dominance  $\phi$  during the earliest replications for 5 system configurations which happen to become those at the top and the bottom of the ranking, and are sufficiently distant to avoid late (after many replications) rank inversions. Comparison of figure 5 ( $\Psi$  operator based on interquartile intervals), figure 6 ( $\Psi$  operator based on 99.9% confidence intervals) and figure 7 (net dominance  $\phi$  with the standard crisp Promethee method), illustrate this observation.

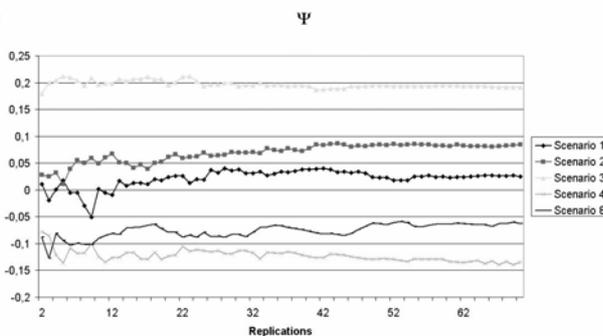


Figure 5:  $\Psi$  Operator Based on Interquartile Intervals

It seems indeed that the use of more information about the variability of the performance measures through the entire replication scheme by means of the interquartile Promethee-i method, is sooner discriminating between system configurations. The final ranking becomes the same by the other methods, but it is stable only after a larger amount of replications.

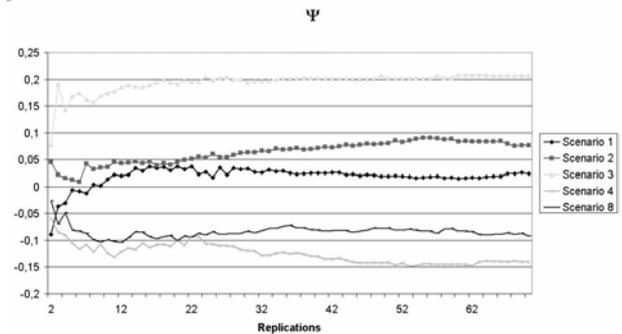


Figure 6:  $\Psi$  Operator Based on Confidence Intervals

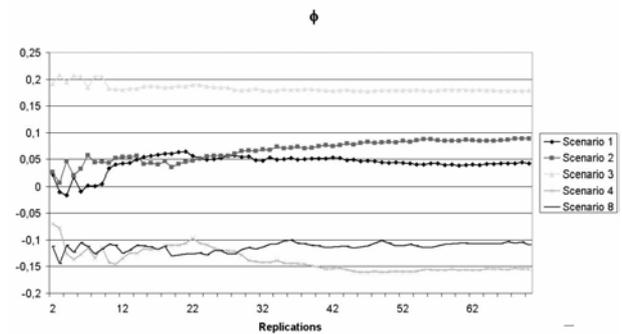


Figure 7: Net Dominance

The same observations were made with the same experiments but by replacing the exponential distribution of system engineers' service times by triangular distributions with a lower variability of the input data in the model. Discrimination between the three computational schemes was then not so obvious. It is indeed more relevant to use the Promethee-i method when the variability of stochastic elements in the model becomes higher.

## CONCLUDING REMARKS

The results for the different computational schemes using the Promethee-i method, compared with the one replacing the assessment intervals by the observed mean values and then applying the original Promethee method, show that the results for all methods are similar if the number of replications increases. However, it seems that the interquartile assessments of performance measures, combined with the Promethee-i method is discriminating very fast between system configurations.

This method seems to be very promising when the number of replications should be kept low (for instance in an almost real-time environment for crisis management decision support).

No assumptions about independence of criteria (performance measures) assessments are of course necessary for applying Promethee-i. No assumptions are made about the probability distribution of assessments.

It is of course not evident to evaluate the probability of correctly selecting system configurations. More experimental research might shed some light on this issue.

Further research about the use of alternative multicriteria methods could be inspired by Fodor and Roubens (1994) or by Pastijn and Leysen (1989).

## REFERENCES

- Brans, J.P., Vincke P. and Mareschal, B. 1986. *How to select and how to rank projects: the promethee method*. European Journal of Operational Research 1986(24), pp. 228-238.
- Detyniecki, M. and Yager, R.R. 2001, *Ranking fuzzy numbers using  $\alpha$ -weighted valuations*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol.8(5), 2001, pp. 573-592.
- Fodor, J. and Roubens, M. 1994. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers.
- Kelton, W.D., Sadowsky, R.P., and Sadowsky, D.A. 1998. *Simulation with Arena*, WCB/McGraw-Hill.
- Kleijnen, J.P.C. 1975. *Statistical Techniques in Simulation – Part II*, Marcel Dekker, Inc.
- Law, A.M. and Kelton, W.D. 1991. *Simulation Modeling and Analysis*. 2<sup>nd</sup> ed, McGraw-Hill, New York.
- Mareschal, B. 1986. *Stochastic multicriteria decision making and uncertainty*. European Journal of Operational Research, 1986 (26), pp. 58-64.
- Mosler, K. and Scarsini, M. 1993. *Stochastic Orders and Applications – A Classified Bibliography*. Lecture Notes in Economics and Mathematical Systems (Vol 401), Springer-Verlag.
- Mareschal, B. and Le Teno, J.F., 1992. *An Interval version of Promethee for the comparison of building products design with ill-defined data on environmental quality*, European Journal of Operational Research, 109, pp.522-529.
- Pastijn, H. and Leysen, J. 1989. *Constructing an outranking relation with ORESTE*. Mathematical and Computer Modelling 1989 (12), pp. 1255-1268.
- Van Loock, R. and Pastijn, H. 2003. *Performance measures in simulating incident management processes of a call centre*. Orbel 17, Brussels Jan 23-24 2003
- Van Loock, R., Van Utterbeeck, F. and Pastijn, H. 2003. *TAPE Performance measures implemented in an incident management model*, Valencia (Spain) ISC2003
- Yager, R. R. 1981. *A Procedure for ordering fuzzy subsets of the unit interval*. Information Science 1981 (24), pp. 143-161.

# Simulation of a Distributed Mutual Exclusion Algorithm Using Multicast Communication

Jonathan Pearlin and Robert Signorile\*  
Boston College  
Fulton Hall 460  
Computer Science Department  
Chestnut Hill, MA 02056  
\*Email: signoril@bc.edu  
\*Phone: 617-552-3936

## KEYWORDS

Distributed Coordination, Distributed Mutual Exclusion

## ABSTRACT

The development of a distributed mutual exclusion algorithm operating via multicast communication is considered necessary to ensure the proper performance of distributed systems. We developed an algorithm that takes advantage of the characteristics of a multicast network and various synchronization mechanisms, such as a timestamp and the election of a central arbitrator from the multicast group. These synchronization safeguards not only enforce mutual exclusion, but also aid in the enforcement of reliable communication between nodes in a distributed system. The implementation of a dynamic rotating server selected from among the members of the multicast group, in combination with a timestamp applied to all messages sent between nodes in the system, forces the ordering of messages sent within the system. This adds a level of predictability and stability to a distributed system and its communication mechanism.

## INTRODUCTION

Distributed systems utilize a network of computers in order to share the workload of an application evenly amongst the members of the network. Such a system must not only coordinate between processes in the system, but must also provide the same functionality and assurances that one finds in a non-distributed program. These assurances include the mutual exclusion of access to shared resources among processes vying for entry into a critical section. Mutual exclusion consists of the prevention of deadlock between processes and the prevention of the starvation of a process attempting to acquire entry to the critical section. Without these assurances, one cannot rely on a system where processes communicate via messages in order to gain entrance to a critical section to be safe. Mutual exclusion in a

distributed system is therefore required to not only ensure that the work is evenly distributed between processes, but that resources available to the system are also shared evenly and fairly.

## PREVIOUS RESEARCH

Providing mutual exclusion to a distributed system requires the consideration of several important factors, including the way in which processes communicate with each other, the type of network on which the system is based and the coordination of processes in the system. Distributed mutual exclusion algorithms achieve this coordination by using techniques such as message passing or token passing to communicate between nodes (Ricart et al. 1981, Suzuki et al. 1985). Furthermore, these algorithms are often based on point-to-point communication in which each node must communicate directly and separately with any other node in the network it wishes to address. Because of this behavior, the focus of such algorithms has been on the reduction of the number of messages that are required to maintain mutual exclusion while preserving synchronization between the processes in the system.

The problem of synchronization in a distributed system is often addressed by utilizing a logical clock shared between all members of the system. The use of a logical clock in the form of a sequence number or timestamp, as proposed by Lamport, imposes ordering on the events that occur in the system (Lamport 1978).

While coordination of the processes in a distributed system is an important consideration in the design of a distributed mutual exclusion algorithm, the method of communication between processes is crucial to the effectiveness of the algorithm. Much of the work in the field of developing distributed mutual exclusion algorithms have been based around the problem of reducing the number of messages necessary to ensure a safe entry into the critical section. Ricart and Agrawala proposed a "message passing" algorithm that requires 2

( $N - 1$ ) messages in order to achieve distributed mutual exclusion in a point-to-point communication-based network (Ricart et al. 1981). In their algorithm, “sequence numbers” are used to create a total ordering of events in the system. While assuring mutual exclusion, the algorithm has a large overhead in terms of the messages needed to achieve mutual exclusion.

Our algorithm promises to obtain mutual exclusion in a constant number (one or two) of messages per request for entrance into the critical section: one message to contact the rotating “server” with a request and one message back to entire system informing the group of the server’s decision. It can be shown that this algorithm is distributed and that it provides mutual exclusion to the system (the term “distributed” is used to describe the equal number of turns that each node takes as the central arbitrator in the system).

### THE PROPOSED ALGORITHM

The proposed algorithm is based on a combination of message passing and election based distributed mutual exclusion algorithms. At all times, the algorithm has a node that has been “elected” to serve as the central arbitrator for resource requests for a fixed term. When a node receives an incoming message, its first course of action (whether it is currently the server or not) is to determine if it is its turn to be the server by examining the timestamp value contained in the message. Server status rotates from one node to the next, giving all nodes in the system a turn to act as the server and handle an equally distributed amount of work. Any node wishing to request a resource or enter the group does so by issuing its request to the entire multicast group. The server (who is just another node in the group) listens for such requests and processes and responds to them appropriately. Once the server node has made a decision regarding the resource request, it broadcasts its response to the entire group. The requesting node, upon receipt of this response, enters the critical section if it has been determined by the server that it is safe to do so. All nodes (other than the node that has been granted access to the resource) simply mark the resource as in use and wait for the node with access to the critical section to broadcast a release command before attempting to request the resource again.

The distributed mutual exclusion algorithms described earlier in this paper all use some form of synchronization to ensure that events are processed in the correct order of occurrence. However, these algorithms deal with fixed networks of computers using point-to-point communication. The use of multicast communication adds an extra level of complexity to the notion of synchronizing events in a distributed system. Here,

nodes do not communicate directly with one another and cannot be sure that their message has reached every other member in the group (this is the nature of multicast communication and must be assumed for the purposes of creating synchronization). This means that all nodes in the group cannot be allowed to control a logical clock apparatus. Instead, there must be a central source that regulates the clock in order to force synchronization upon all of the nodes in the system. Naturally, the server node was chosen to be in charge of synchronizing the logical clock created for the system (specifically, the algorithm uses a simple timestamp that acts like a real clock – when it equals a prescribed limit, the clock is reset to zero). By designating the server as the only node with authority to change the value of the clock, the algorithm ensures that all clocks in the system will be synchronized and that the workload of the system stays distributed equally.

The management of the synchronization mechanism is crucial to the performance of the proposed algorithm. The current server node increments the timestamp when it receives a message that pertains to a resource or membership request. The following diagram displays this process:

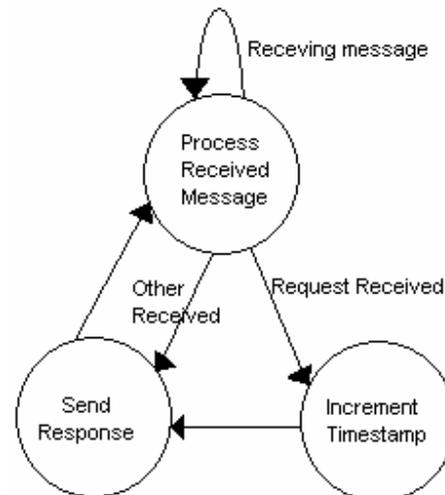


Figure E: State diagram showing when the server node increments the timestamp value.

As we can see from the diagram, the timestamp is increased after this message is received and right before a response is sent out to ensure that subsequent incoming messages are not discarded. The server has the sole responsibility and authority to change the value of the timestamp, which derives from its role as the central arbitrator in the system. This is the reason that synchronization can be achieved even with the use of multicast communication. By subjugating all other

nodes under the authority of the server node, the system is assured that the synchronization mechanism will remain stable.

While the inclusion of a timestamp is necessary to ensure that ordering of events in the system is possible, an issue does arise regarding the rotating server status. The server designation (as discussed earlier) is based on a dynamic calculation, which takes into consideration the number of members currently in the group, the current value of the timestamp, a node's relative position in the group based on its logical identification number and a constant which represents the number of turns each node will serve as the server. Therefore, because a node's status as server is not fixed and changes dynamically based on its relative position in the group (numerical ordering based on identification number), the server node must be consciously aware of the current timestamp and when its turn comes to an end. A potential problem can arise when the server is on its last turn and receives a request which increments the timestamp. In order to deal with this potential synchronization problem, a "SYNC" message was introduced to the algorithm to force synchronization on a server's last turn. The following diagram represents this addition to Figure E:

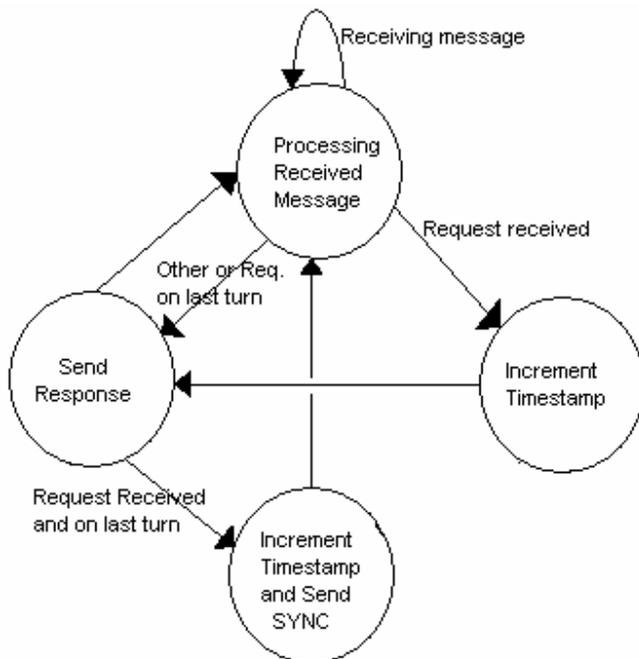


Figure F: State diagram showing the addition of the SYNC message to server module

This gives the server the power to respond to the current resource request and then increment the timestamp value without having to worry about creating the situation described in detail above. If the server is currently on its

last turn, it sends its response, then updates the timestamp value (effectively ending its turn as server), and sends out the "SYNC" message with the new timestamp value. All nodes accept the timestamp value contained in a "SYNC" message, regardless of its value. This is because the value is guaranteed to be correct, as the only node that can issue such a message is the acting server. This necessary safeguard is the result of the server apparatus and its implementation. By developing a system based on the dynamic designation of a node to act as the server (a modification on election-based mutual exclusion algorithms), the addition of a special type of message to ensure synchronization and to prevent a node from taking on the role of the server prematurely is essential. This added safety feature protects the synchronization of the system's logical clock and as a result, protects the distributed nature of the algorithm. In order for a distributed system to achieve mutual exclusion, no two nodes should have access to the same shared resource at the same time. This means that a node must exit the critical section before any other node can gain access to the same critical section.

Assertion: The proposed algorithm achieves mutual exclusion

Proof: In order to prove that the proposed algorithm achieves mutual exclusion, we must assume the contrary condition: two nodes can both have access to the same shared resource at the same time. In order for two nodes to have access to the same critical section at the same time, one of two cases must be possible:

1. Node A must have already been in the critical section at the time in which Node B entered the same critical section.
2. Node A and B simultaneously entered the critical section.

In the first case, node B is granted access to the critical section that is already occupied by node A. This means that the server node has made two grant responses for the same critical section without receiving a release from node A. This is impossible with the proposed algorithm. Upon receiving a request for the critical section from node A, the server node would see that the critical section is still available. It would lock the critical section (mark it as in use by node A) and issue a broadcast message to the group stating that node A now has permission to use the critical section. While this is occurring, assume that node B also issues a request for the same resource. While waiting for its response, node B receives the message from the server indicating that node A has been given the critical section. Node B would therefore mark the critical section as being in use by node A. This also means that node B would not be

able to enter the critical section until node A broadcasts its release message to the entire group (this is because the server node cannot issue permission to the same resource until it is released). Therefore, the first case cannot occur and mutual exclusion is preserved.

In the second presented case, the server node receives two different requests for the same resource simultaneously. The server node cannot issue two different responses for the same resource. Instead, the server will issue one response indicating which node has been granted access to the critical section. Because of the “passive” nature of the non-server nodes, the node that is not granted access will accept the decision and mark the resource as in use. When the node that is granted access is done with the resource and broadcasts its release message to the group, the node that was denied in its request will again be able to issue a request for the resource. The use of the central arbitrator to determine access to shared resources ensures that no two nodes can enter the critical section by way of simultaneous resource requests. Thus, mutual exclusion is preserved in the proposed algorithm.

Deadlock occurs in a distributed system when no node can enter the critical section despite requests being issued for entrance. This is usually caused by outstanding requests for the critical section or a failure in the permission granting authority.

Assertion: Deadlock cannot occur in the proposed algorithm.

Proof: In order to prove that the proposed algorithm avoids deadlock, we must assume the contrary condition: deadlock can occur among nodes wishing to access a critical section. This would mean that all of the nodes that have made a request for the critical section are waiting indefinitely for a response. In other words, the central arbitrator in the system has failed to respond to a request for a resource. However, this cannot be the case. A node cannot issue a request for a resource that is already in use, as there is no mechanism to block on a resource. Instead, a node checks its list of system resources to see if a particular resource is available. If it is available, then the node is free to issue a request for the resource. If it is not available, the node periodically checks to see if the resource’s user has freed the resource before requesting that resource again. Either way, a node receives a broadcasted response from the server to the entire group informing all nodes as to which node has secured the right to use the resource. Therefore, the node cannot be in a state of constantly waiting for a response on a resource (this could occur if the server node only communicated directly with the node that is being granted access to the critical section and left other

requesting nodes in the dark about its decision). This also means that a node receives information about the availability of a resource from the server when any node in the system requests a resource. Because of this, a node will not issue a request for a resource it already views as being in use.

On the surface, this “passive” nature, combined with the periodic checking of a resource’s availability, could appear to cause a race condition, in which nodes are constantly contacting the server asynchronously to find out about the status of a resource. This is not what is meant by “periodically” checking to see if the resource is free. Instead, the node checks its own list of the system resources to see if a resource is available. As soon as the node views the resource as free, it may then reapply for the resource. Because all nodes receive a broadcasted response from the node that possesses the resource when the resource is released, the process is synchronized in terms of when all of the nodes view the resource as available for use. This synchronization therefore avoids a race condition. Thus, deadlock is avoided by using the broadcast nature of multicast communication in conjunction with the “passive” state of a node requesting a resource.

It is also possible for deadlock to occur if no node has taken on the role of the system’s central arbitrator. This situation could arise if the node that is to become the server never receives the “SYNC” message sent out by the current server during its last turn. Because the current server sends out the “SYNC” message and then relinquishes its turn as server, neither it nor the next server believe that they are the active server if the message is lost (which is possible with multicast communication). The developed algorithm has been constructed to avoid permanent deadlock caused by such a scenario. Every node checks the timestamp of each incoming message and determines if it is its turn to be the server before processing the message. This means that even though the correct timestamp value never reached the node that is to be the next server, the correct value is present in all of the other nodes that received the message. Since the timestamp value is included in every packet sent out, those who send out packets with an invalid timestamp (this would be any node that did not receive the “SYNC” message) will have their messages discarded by those nodes that received the “SYNC” message and have the correct timestamp value. This preserves the correct state of the system. Furthermore, the next time any of the nodes (including the former server) that have the correct timestamp value send out a packet, all of the nodes with the wrong timestamp value (including the node that is supposed to be the server) will finally receive the correct timestamp value. This, in effect, would cause the node that is supposed to be the

server to notice that it is its turn to be the server. During this period without a server, the system would remain stable because only a server can make system-changing decisions (such as granting access to a critical section or adding members to the group) and increment the timestamp. This ensures that the system stays in the same state while it is attempting to rectify the lost communication. Such synchronization gives the system some margin of error in terms of the rotating server apparatus. While it may be true that a message or two directed to the server may be lost if this scenario occurs, it does not result in permanent deadlock of the system. This is because of the “passive” nature of requesting nodes, which do not block while waiting for a response. Multicast communication is unreliable and therefore extra safeguards must be put into place in the system in order to allow it to recover from lost packets.

Finally, deadlock could result in the system if a node that has been granted the critical section stops responding to the system. This would mean that a resource could be indefinitely held by one of the nodes and therefore would prevent other nodes from every acquiring that resource (this could also cause starvation, as a node would not be able to get the resource that it needs). However, the node acting as the server in the developed algorithm has the means to deal with this situation. In order to determine if a node is still responding to the group, the server uses a “ping” apparatus. If, at the end of its turn, the server determines that a node that is currently holding a resource is not responding to the group, the server can reclaim the resource for the group by broadcasting the resource’s availability to the entire group. This frees the resource and allows any node that wishes to use the resource next to issue a request for that resource. Therefore, the “ping” apparatus serves the dual purpose of preventing the server status from falling to a non-responding node, as well as reclaiming resources from failing nodes. Thus, deadlock (and starvation, as discussed above) is avoided.

Starvation occurs in a distributed system when a node is continuously prevented from accessing the critical section while other nodes are allowed to access the critical section. This can occur in systems where unfair weighting is used to process requests for the critical section.

Assertion: Starvation cannot occur in the proposed algorithm.

Proof: In order to prove that the proposed algorithm avoids starvation, we must assume the contrary condition: starvation can occur, preventing a node wishing to access a critical section from ever being granted access. This would mean that the node continuously has its request denied by the central

arbitrator, possibly as a side effect of network traffic (the node’s messages are constantly arriving too late to be granted access). However, as mentioned earlier in the discussion of the algorithm, the server node has priority to enter the critical section because it is the node that grants access to the critical section. Therefore, a node may not have a chance to enter the critical section while it is not the server, but upon becoming the server, its requests supersede any requests received at the same time (the server node acts just like a regular node when requesting resources – it does not block on an outstanding resource request). This means that any starving node will not starve forever – it will eventually become the access granting authority (due to the rotating server apparatus) with the ability to grant its own access to the critical section, if it is available. Furthermore, as mentioned previously, the server node has the ability to determine if a node that currently holds a resource has stopped responding to messages from the group. If it finds this to be the case, it has the authorization to reclaim the resource for the group. Thus, starvation is prevented in the proposed algorithm.

## SIMULATIONS

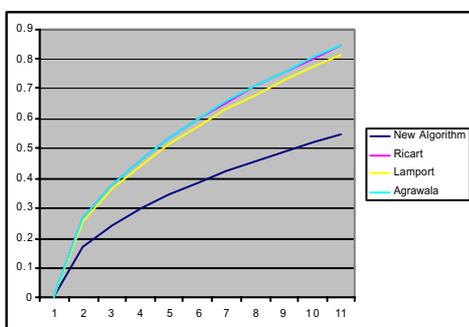
We simulated this algorithm as part of a distributed system represented  $N$  processes running on  $M$  machines. The number of processes and machines were dynamic in that they could be altered during the simulation (i.e. destroyed or created). The resources were fixed in size; the critical section was entered/exited many times per process. We also implemented Lamport’s and Ricart and Agrawala algorithms as comparisons.

Algorithms that provide mutual exclusion to a distributed system are evaluated primarily on the bandwidth utilized to communicate with the other computers in the system, the effect the algorithm has on the throughput of the system, and the effect of synchronizing the computers in the system. The current body of work in this field is primarily focused on enforcing mutual exclusion in a distributed system based on using a point-to-point protocol for means of inter-node communication. Algorithms that meet this condition include message-passing algorithms, election algorithms and token-based algorithms.

The goal for all of these algorithms is ultimately the same: the reduction of the number of messages that is required to be exchanged between processes in order to enforce mutual exclusion. By reducing the number of messages needed to ensure mutual exclusion, these algorithms attempt to improve the optimality of the overall system in terms of bandwidth and throughput. (Lamport 1978, Suzuki et al. 1982, Garcia-Molina 1982)

The proposed algorithm achieves mutual exclusion in a constant number (one or two) of message exchanges. This is the result of the combination of four characteristics of the developed distributed system: the rotating server apparatus, the logical identification numbering of nodes, the “passive” disposition of non-server nodes and the broadcast nature of multicast communication. The deployment of a node to act as the central arbitrator in the system means that all resource requests must be directed to the server node. Therefore, any node wishing to request a resource must only issue one message, asking the server node (whomever it is – the requesting node does not need to have direct knowledge of the server to have its request heard) for entrance into the critical section. For the server’s part, it needs to only send one response to the entire group (as a result of the “passive” non-server nodes and the broadcast protocol), instead of contacting each node individually to notify them of its decision. Each node accepts the incoming response regardless of whether or not it has requested a resource. This whole transaction requires an exchange of two messages to obtain mutual exclusion. If the server wishes to enter the critical section and it is open, then the number of messages required to secure the critical section is reduced to one (the broadcasted message to the entire group informing the group that the resource has been secured). This constant number of messages exchanged is the minimum number of messages required to ensure mutual exclusion in regards to the proposed algorithm. Here, we can fully see the benefit of using multicast communication in order to reduce the number of messages required to provide mutual exclusion to a distributed system. The proposed algorithm takes full advantage of the nature of multicast communication in order to provide mutual exclusion through the exchange of a constant number of messages.

In our simulations, we compared the time to converge (i.e. the time to notify all processes) that resources were needed and released. The below table is normalized to the new algorithm



## CONCLUSIONS

This paper presents an algorithm that provides mutual exclusion using multicast communication to a distributed system. The proposed algorithm combines the effectiveness of message passing and election based distributed mutual exclusion algorithms in order to facilitate mutual exclusion. We have also implemented this algorithm as part of a distributed 3D game, with great success.

## REFERENCES

- Lamport, Leslie. “Time, Clocks, and the Ordering of Events in a Distributed System.” Communications of the ACM. Volume 21, Number 7. July 1978. 558-565.
- Ricart, Glenn and Agrawala, Ashok K. “An Optimal Algorithm for Mutual Exclusion in Computer Networks.” Communications of the ACM. Volume 24, Number 1. January 1981. 9-17.
- Suzuki, Ichiro and Kasami, Tadao. “An Optimality Theory for Mutual Exclusion Algorithms in Computer Networks.” Proceedings of The 3<sup>rd</sup> International Conference on Distributed Computing Systems. October 18-22, 1982. 365-370.
- Maekawa, Mamoru. “A vN Algorithm for Mutual Exclusion in Decentralized Systems.” ACM Transactions on Computer Systems. Volume 3, Number 2. May 1985. 145-159.
- Suzuki, Ichiro and Kasami, Tadao. “A Distributed Mutual Exclusion Algorithm.” ACM Transactions on Computer Systems. Volume 3, Number 4. November 1985. 344-349.
- Garcia-Molina, Hector. “Elections in Distributed Computer Systems.” IEEE Transactions on Computers. Volume C-31, Number 1. 1982. 48-59.

## BIOGRAPHY

Robert Signorile is an Associate Professor in the Computer Science Department of Boston College. His research interests include multimodal simulation, simulation in business, networks and distributed computing. He has published regularly in applied simulation, simulation methodology, distributed systems and networks.

Jonathan Pearlin is a recent graduate of the Boston College Computer Science department. His work revolves around operating systems support for distributed gaming.

# IMPROVEMENT OF THE STATISTICAL ACCURACY FOR THE THREE-DIMENSIONAL MONTE CARLO SIMULATION OF ION IMPLANTATION

Robert Wittmann  
Andreas Hössinger  
Siegfried Selberherr  
Institute for Microelectronics  
Technical University Vienna  
Gusshausstr. 27–29, A-1040 Vienna, Austria  
E-mail: Wittmann@iue.tuwien.ac.at

## KEYWORDS

Simulation of Ion Implantation, Smoothing Monte Carlo Simulation Results, Statistical Fluctuation of Doping Profiles, TCAD Tool Development.

## ABSTRACT

Statistical fluctuations of three-dimensional Monte Carlo simulation results require a sophisticated post-processing of the obtained data. We present an advanced algorithm for smoothing Monte Carlo results of ion implantation and translating them from the internal ortho-grid of the simulator to an unstructured grid which is suitable for subsequent process simulation steps. This algorithm allows a more accurate prediction of doping profiles, which is essential for an implantation treatment of deep-submicrometer device structures. Basically, the ion concentration value on a grid point of the unstructured grid is approximated by means of Bernstein polynomials, evaluated in a fast way only in the middle point of that cell which contains the new grid point. The key idea is to estimate also the concentration difference between the new point and the middle point by calculating the scalar product of the concentration gradient and the distance vector. In that way, effects based on the cell discretization can also be significantly reduced, which leads to more realistic doping profiles. The impact of the advanced smoothing procedure on the statistical accuracy for three-dimensional implantation applications is demonstrated.

## INTRODUCTION

Ion implantation is the most important doping technique for electronic device fabrication, in particular for ultra large scale integration (ULSI) circuits. The ongoing trend of scaling device feature sizes down to the deep-submicrometer regime requires TCAD tools which provide a more accurate prediction of doping profiles and a full three-dimensional implantation treatment.

These requirements drive the improvement and optimization of existing three-dimensional Monte Carlo simulation tools in the field of ion implantation.

The Monte Carlo method is based on applying random behavior at an atomistic level (Hobler and Selberherr 1989), (Ziegler et al. 1995). Particularly, the position where an ion hits the crystalline target is calculated using random numbers. Furthermore, the lattice atoms of the target are in permanent movement due to thermal vibrations. Thus, the actual positions of the vibrating atoms in the target are also simulated using random numbers. In this model the ion implantation process is accurately simulated by computing a large number  $N$  of individual ion trajectories through a semiconductor material. The trajectory of each implanted ion is determined by the interactions with the atoms and electrons of the target material. The final position of an implanted ion is reached where it has lost its kinetic energy. The crystalline model of silicon allows to simulate the channeling effect during ion implantation. Additionally, point defect distributions generated as results of ion implantation can also be calculated by the Monte Carlo approach. The Monte Carlo method to model ion implantation has the advantage of being a physically based method and therefore it is easily extendable for new technological conditions without the need for additional extensive calibration. On the other hand, long computing times prevent the standard use of Monte Carlo simulation tools in technology optimization.

Based on random numbers, the results obtained with the Monte Carlo method are never exact but rigorous in a statistical sense. The results converge to the used model characteristics by increasing the number  $N$  of simulated ions. As with experiments, the errors on Monte Carlo results are divided into two classes: statistical errors and systematic errors.

The statistical error is basically determined by the number  $N$  of calculated trajectories. The slow convergence rate of the Monte Carlo method leads to long simulation runs and produces ion concentration

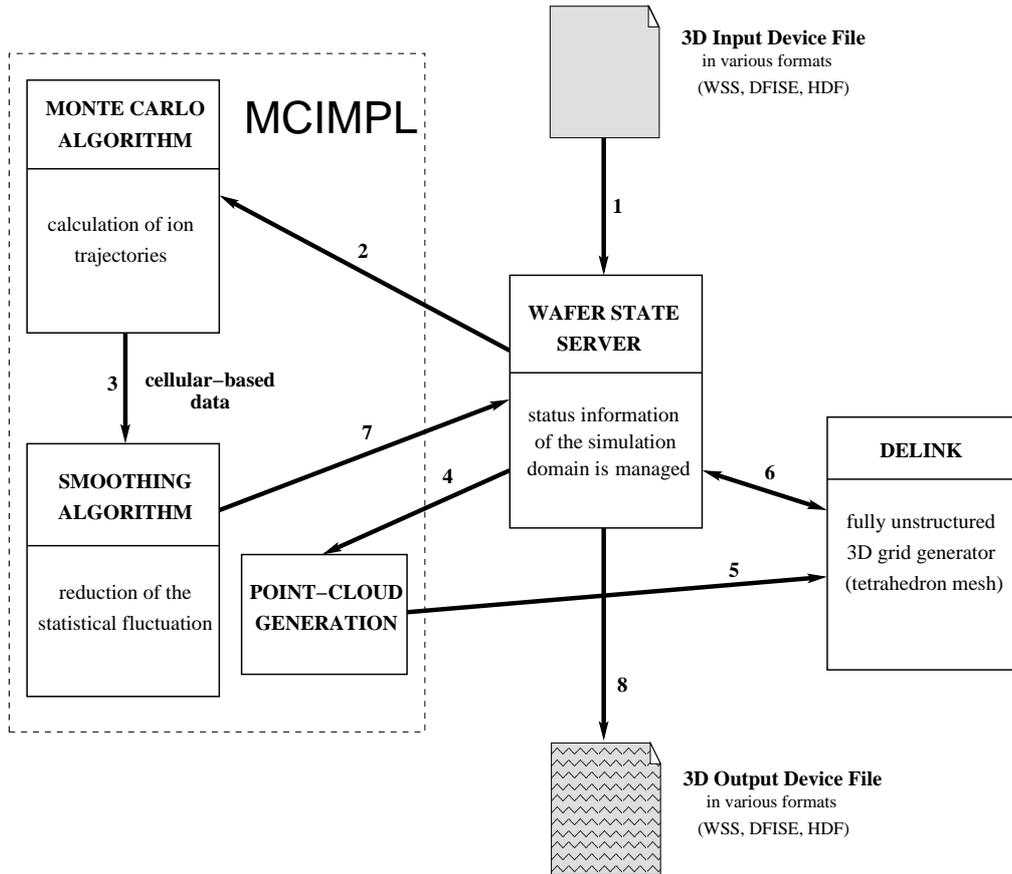


Figure 1: Data Flow and Involved Process Simulation Tools

estimates which tend to have high variances. The obvious way to reduce the statistical error is by increasing the number  $N$  of simulated ions. This error vanishes for  $N \rightarrow \infty$ . Speed-up techniques like the trajectory split method (Bohmayr et al. 1995) or trajectory reuse method (Hössinger and Selberherr 1999) help to increase  $N$  by holding the additional computational effort within acceptable limits. On the other hand, systematic errors arise due to model limitations or insufficient calibration of the simulator. Model validation or software verification will not be covered in this work.

Paticularly in three-dimensional applications a worse statistical representation arises in regions with a dopant concentration several orders of magnitudes smaller than the maximum (projected range). The raw Monte Carlo results are smoothed through a post-processing step in order to achieve results with an acceptable accuracy even in deep regions with a poor statistical representation of dopants. It turned out that fluctuations of the original data can effectively be reduced by using an approximation with Bernstein polynomials. We have developed an advanced smoothing algorithm which extends the Bernstein approximation by calculating an additional linear approximation also in a fast manner. This algorithm

helps to significantly reduce the statistical error of three-dimensional Monte Carlo simulation results.

## THE SIMULATOR

All Monte Carlo simulation experiments were performed with the object-oriented, multi-dimensional ion implantation simulator MCIMPL-II. The simulator is based on a binary collision algorithm and can handle arbitrary three-dimensional device structures consisting of several amorphous materials and crystalline silicon. In order to optimize the performance, the simulator uses cells arranged on an ortho-grid to count the number of implanted ions and of generated point defects. The final concentration values are smoothed and translated from the internal ortho-grid to an unstructured grid suitable for subsequent process simulation steps, like finite element simulations for annealing processes.

Figure 1 shows the data flow during the simulation of ion implantation. The simulator MCIMPL-II is embedded in a process simulation environment by using the object-oriented WAFER STATE SERVER library (Binder and Selberherr 2003), (Binder et al. 2003).

The WAFER STATE SERVER has been developed

in order to integrate several three-dimensional process simulation tools used for topography, ion implantation, and annealing simulations. It holds the complete information describing the simulation domain in a volume mesh discretized format, and it provides convenient methods to access these data. The idea is that simulators make use of these access methods to initialize their internal data structures, and that the simulators report their modifications of the wafer structure to the WAFER STATE SERVER. Thereby a consistent status of the wafer structure can be sustained during the whole process flow.

The meshing strategy of DELINK follows the concept of advancing front Delaunay methods and produces tetrahedral grid elements (Fleischmann and Selberherr 2002).

### ADVANCED SMOOTHING ALGORITHM

The smoothing of the raw Monte Carlo result is performed by approximating the concentration value on a grid point of the unstructured grid by means of Bernstein polynomials defined in a cubical surrounding space (Heitzinger et al. 2003). The Bernstein polynomial  $B_{f,n,n,n}(x_1, x_2, x_3)$  approximates a function  $f$  of 3 variables, where  $n \in \mathbf{N}$  are the concentration sample points in each dimension. The Bernstein approximation  $B_{f,n,n,n}$  is specified by  $n^3$  sample points whereby  $B_{f,n,n,n}$  does not run through the sample points, but each of them affects the approximated function. Such sample points are usually called control points, since they enforce the function progression.

The statistical accuracy of the Monte Carlo result is determined by the number of counted ions per cell. More and more empty cells at increasing penetration depth downgrade the statistics dramatically. The calculation of the Bernstein approximation needs a reasonable information in all sample points. In order to fulfill this requirement, the concentration value at a sample point in an empty cell is calculated by averaging over the values of surrounding cells.

It is not necessary to calculate the Bernstein polynomial explicitly, since each polynomial is only evaluated in the middle point  $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$  of the domain  $[0, 1]^3$ . In this case, the approximating polynomial of order  $(n^3 - 1)$  can be simplified to the formula according to (1), which enables a fast calculation of the approximated value.

$$B_{f,n,n,n}\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) = \sum_{k_1=0}^n \sum_{k_2=0}^n \sum_{k_3=0}^n f_{k_1,k_2,k_3} \binom{n}{k_1} \binom{n}{k_2} \binom{n}{k_3} \left(\frac{1}{2}\right)^{3n} \quad (1)$$

For the original smoothing algorithm the approximation of the concentration values for the grid points of

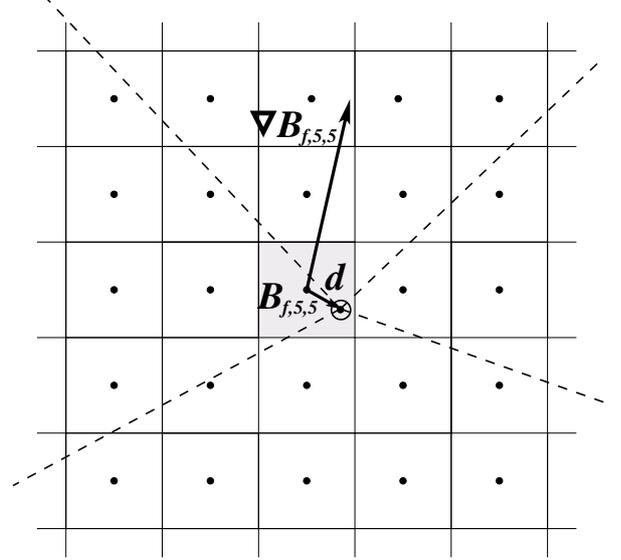


Figure 2: This sketch demonstrates the calculations performed for one point of the new unstructured grid in a two-dimensional example. It can be applied to the third dimension in an analogous manner. The thin orthogonal lines confine the cells of the internal grid. The four dashed lines denote the unstructured grid, where the new point is marked by a small circle. In the first step the  $5^2$  sample points are used to calculate the concentration value  $B_{f,5,5}$  at the middle of the central grey cell. Then the scalar product of the gradient and the distance vector  $\mathbf{d}$  is calculated to produce a delta concentration value.

the unstructured grid was evaluated only in the middle point of that cell which contains the grid point. The drawback is that two grid points contained in the same cell get an equal concentration value. This approximation can be improved if the distance of the new point from the middle point is also taken into consideration as it is depicted in Figure 2. According to (2) the concentration difference  $\Delta B_{f,n,n,n}$  between the new point and the middle point can be approximated by the scalar product of the concentration gradient and the distance vector. (3) points out that it is possible to calculate the required three partial derivatives also in a fast way.

$$\Delta B_{f,n,n,n} = \nabla B_{f,n,n,n} \cdot \mathbf{d} \quad (2)$$

$$\begin{aligned} \frac{\partial}{\partial x_i} B_{f,n,n,n} \Big|_{\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)} &= \\ &= \sum_{k_1=0}^n \sum_{k_2=0}^n \sum_{k_3=0}^n f_{k_1,k_2,k_3} \binom{n}{k_1} \binom{n}{k_2} \binom{n}{k_3} \cdot \left(\frac{1}{2}\right)^{3n} (4k_i - 2n) \end{aligned} \quad (3)$$

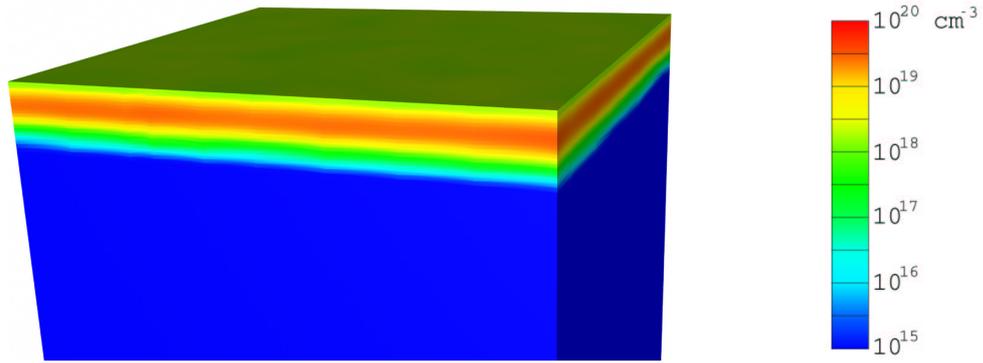


Figure 3: Accurate Monte Carlo Simulation Result of Phosphorus Implantation in Silicon with  $N = 4 \cdot 10^6$  Simulated Ions, an Energy of 25 keV, and a Dose of  $10^{14} \text{ cm}^{-2}$

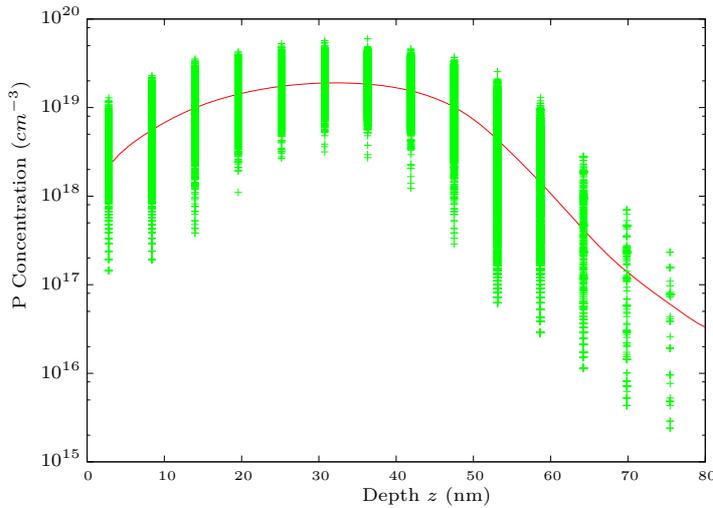


Figure 4: Fluctuation of the Raw Monte Carlo Result

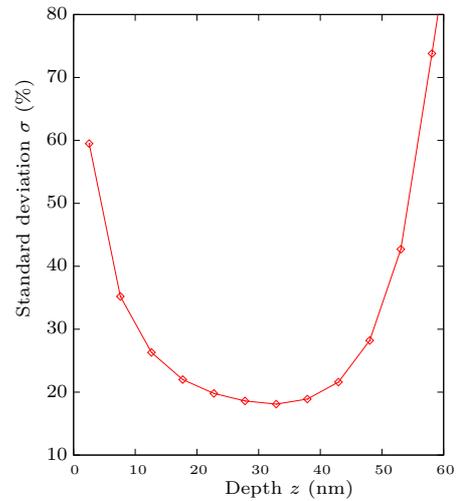


Figure 5: Raw Result

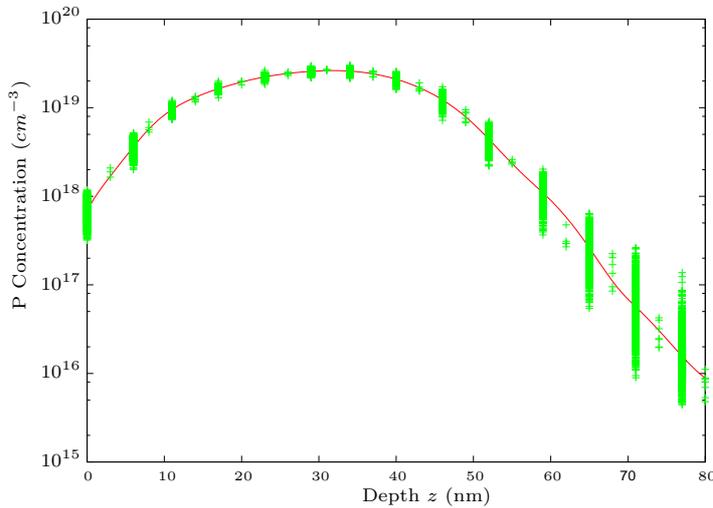


Figure 6: Fluctuation of the Smoothed Simulation Result

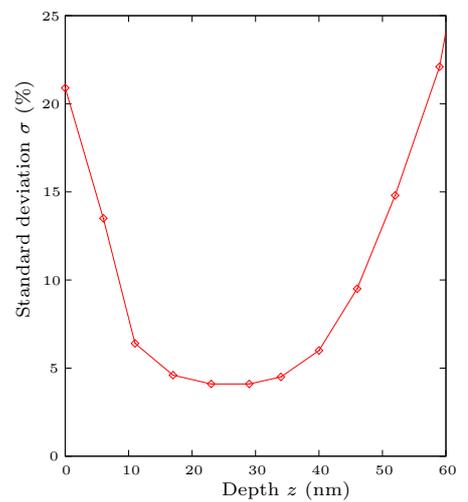


Figure 7: Smoothed Result

For the analysis of the implemented smoothing algorithm, numerical experiments were performed with the simulator MCIMPL-II on a three-dimensional structure equivalent to one-dimensional problems. We assume that all simulated ions are statistically inde-

pendent. Figure 3 shows the three-dimensional result for the implantation of phosphorus ions into a crystalline silicon substrate with  $N = 4 \cdot 10^6$  simulated ions. We extracted  $z$  coordinates and phosphorus concentration values  $C$  (vertical direction) from

all  $120 \times 112 \times 20$  cells of the simulation area. This leads to Figure 4 which shows a significant statistical fluctuation of the ion concentration at equal penetration depth  $z$  for  $N = 4 \cdot 10^6$  ions. The relative standard deviation  $\sigma$  of the impurity concentration in a plane  $z = \text{const}$  is a measure for the simulation error of three-dimensional results compared to one-dimensional results (Figure 5). The mean impurity concentration  $\bar{C}(n)$  of  $n$  ortho-grid points at equal location  $z$  forms the one-dimensional doping profile. The standard deviation  $S(n)$  of a sample defined by the concentration values of  $n$  grid points in a plane  $z = \text{const}$  is given by

$$S(n) = \sqrt{\frac{\sum_{i=1}^n [C_i - \bar{C}(n)]^2}{n-1}} \quad (4)$$

$$\sigma = \frac{S(n)}{\bar{C}(n)} \quad (5)$$

The relative standard deviation  $\sigma$  according to (5) is calculated in order to evaluate the three-dimensional raw Monte Carlo result before smoothing. Additionally, we extracted all  $z$  coordinates and smoothed phosphorus concentration values from the unstructured grid. This leads to Figure 6 which shows a clearly reduced fluctuation of the phosphorus concentration compared to Figure 4. Most of the ions come close to the mean projected range  $R_p$  to rest, causing a smaller variance there. Figure 7 shows the corresponding relative standard deviation  $\sigma$  of the final simulation result.

As measure of the improvement, the ratio of the maximum of the standard deviation  $\sigma_{\text{max}}$  within the range  $2 \cdot \Delta R_p$  (twice the straggling at the mean projected range) of the doping profile, after and before smoothing, can be used. In our case  $\sigma_{\text{max,after}}/\sigma_{\text{max,before}} = 0.25$  for  $2 \cdot \Delta R_p = 22$  nm at  $R_p = 30$  nm.

$$\sigma_{\text{max}} = \text{const} \cdot \frac{1}{\sqrt{N}} \quad (6)$$

The theoretical simulation error  $\sigma_{\text{max}}$  according to (6) follows from the Central Limit Theorem (Law and Kelton 2000). It has been expectedly verified by simulation experiments with different  $N$  (Wittmann et al. 2003). In order to reduce  $\sigma_{\text{max}}$  by 1/4 only by increasing  $N$ , one has to increase  $N$  by a factor of 16. But as demonstrated in Figure 4–7 also a significant improvement of the statistical accuracy can be achieved by sophisticated postprocessing, in particular through the filtering effect of the Bernstein polynomials, which eliminates high-frequency fluctuations from the original data.

The linear approximation of the delta concentration value between the grid point of the unstructured grid

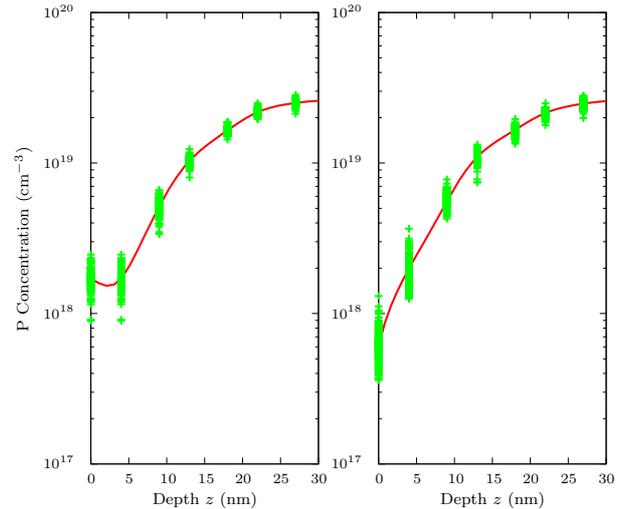


Figure 8: Effect of the Linear Approximation

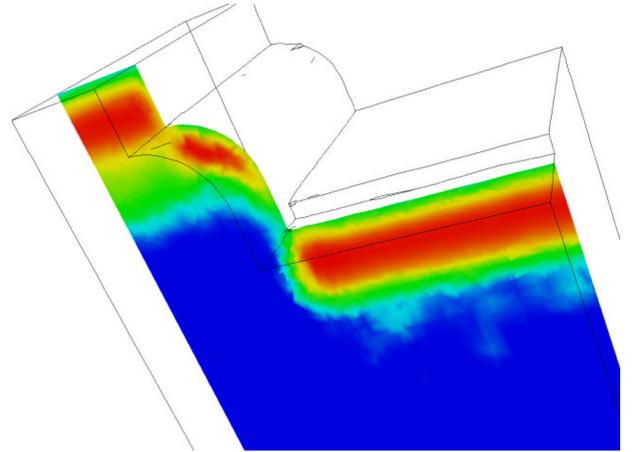


Figure 9: Monte Carlo Implantation Application

and the associated cell middle point significantly reduces the effect of the cell discretization. A rough cell dimension (5 nm) is used for the Monte Carlo simulation to count the number of implanted ions in a computationally efficient way. The left part of Figure 8 shows the doping profile produced by a Bernstein approximation without performing an additional linear approximation in the smoothing procedure. The internal cell dimension of 5 nm produces equal concentration values for the first two samples. The right part of this picture shows the doping profile obtained after advanced smoothing. As demonstrated in this comparison, the advanced smoothing algorithm leads to a smoother and therefore to a more realistic doping profile. For the year 2003 the International Technology Roadmap for Semiconductors (Semiconductor Industry Association 2001) predicted the necessity of a simulation accuracy of 5% (5 nm) for vertical and lateral junction depths. Longer computing times or worse statistics would arise if the cell dimension of the simulator would be further reduced under 5 nm

in order to enhance the accuracy. In contrast to that the used smoothing algorithm can help to improve the accuracy of Monte Carlo results in a computationally efficient way.

Figure 9 shows the result of an implantation performed with the simulator MCIMPL-II. In this application a real-world device structure for processing a MOS transistor was used as input of the simulation. Arsenic ions with an energy of 70 keV and a dose of  $3 \cdot 10^{15} \text{ cm}^{-2}$  were implanted. In this example only 2000000 initial ions were used to demonstrate the fluctuation of the doping profile. It is recommended to use at least 2000000 ions/ $\mu\text{m}^2$ , otherwise the simulation result is inacceptably inaccurate due to the statistical fluctuation.

## CONCLUSION

Three-dimensional Monte Carlo simulation results of ion implantation tend to inherent high variances in particular in regions with a bad statistical representation of dopants. The goal of this work is to achieve a more accurate prediction of doping profiles with the aid of advanced smoothing of Monte Carlo simulation results in a computationally efficient way. The extended algorithm has an impact on the accuracy of the predicted doping profile in various ways. High-frequency fluctuations from the original data are eliminated through the filter effect of the Bernstein polynomials in the first approximation step of the final result. Next, a linear approximation step is performed to reduce effects which arise through the cell discretization of the simulator. The analysis of the results produced by the advanced algorithm demonstrates the gained improvement of the doping profile which can immediately be used in a subsequent process simulation step.

## REFERENCES

- Binder T. and Selberherr S. 2003. "Rigorous Integration of Semiconductor Process and Device Simulators". *IEEE Transactions on CAD*, 99–102.
- Binder T., Hössinger A., and Selberherr S. 2003. "Rigorous Integration of Semiconductor Process and Device Simulators". *IEEE Transactions on CAD*.
- Bohmayr W., Burenkov A., Jürgen L., Heiner R., and Selberherr S. 1995. "Trajectory Split Method for Monte Carlo Simulation of Ion Implantation". *IEEE Transactions on Semiconductor Manufacturing*, Vol. 8, No. 4, 402–407.
- Fleischmann P. and Selberherr S. 2002. "Enhanced Advancing Front Delaunay Meshing in TCAD". *SISPAD 2002*, 99–102.
- Heitzinger C., Hössinger A., and Selberherr S. 2003. "An Algorithm for Extracting and Smoothing Three-Dimensional Monte Carlo Simulation Results". *IEEE Transactions on CAD*, Vol. 22, No. 7, 879–883.
- Hobler G. and Selberherr S. 1989. "Monte Carlo Simulation of Ion Implantation into Two- and Three-Dimensional Structures". *IEEE Transactions on CAD*, Vol. 8, No. 5, 450–489.
- Hössinger A., and Selberherr S. 1999. "Accurate Three-Dimensional Simulation of Damage Caused by Ion Implantation". *Int. Conf. on Modeling and Simulation of Microsystems*, 363–366.
- Semiconductor Industry Association. 2001. *The International Technology Roadmap for Semiconductors 2001 Edition, Modeling and Simulation*.
- Law A.M. and Kelton W.D. 2000. "Simulation Modeling and Analysis". McGraw-Hill, USA, 254.
- Wittmann R., Hössinger A., and Selberherr S. 2003. "Statistical Analysis for the Three-Dimensional Monte Carlo Simulation of Ion Implantation", *Industrial Simulation Conference Proceedings 2003*, 159–163.
- Ziegler J.F., Biersack J.P., and Littmark U. 1995. "The Stopping Range of Ions in Solids". Pergamon Press.

## AUTHOR BIOGRAPHY



**ROBERT WITTMANN** was born in Vienna, Austria, in 1966. He worked from 1989–1997 as an engineer in the Development Department at the company Kapsch, Vienna. After that he studied computer engineering at the 'Technische Universität Wien', where he received the degree of 'Diplom-Ingenieur' in January 2002 with focus on CORBA applications, with distinction. He joined the 'Institut für Mikroelektronik' in June 2002, where he is currently working on his doctoral degree. His scientific interests include simulation of ion implantation, statistical analysis of Monte Carlo simulations, parallel processing, and software technology. His e-mail address is: [wittmann@iue.tuwien.ac.at](mailto:wittmann@iue.tuwien.ac.at).



**ANDREAS HÖSSINGER** was born St. Pölten, Austria, in 1969. He studied technical physics at the 'Technische Universität Wien', where he received the degree of 'Diplom-Ingenieur' in January 1996. He joined the 'Institut für Mikroelektronik' in June 1996. In 1998 he held a visiting research position at Sony in Atsugi, Japan. In September 2000 he finished his Ph.D. degree at the 'Institut für Mikroelektronik' where he is currently enrolled as a post-doctoral researcher. In 2001 he held a position as visiting researcher at LSI Logic in Santa Clara, CA, USA within the scope of a cooperate research project on three-dimensional process simulation. In 2001 he also received a grant from Austrian Academy of Science within the scope of the Austrian Program for Advanced Research and Technology for his work on three-dimensional process simulation. His e-mail address is: [hoessinger@iue.tuwien.ac.at](mailto:hoessinger@iue.tuwien.ac.at).



**SIEGFRIED SELBERHERR** was born in Klosterneuburg, Austria, in 1955. He received the degree of 'Diplom-Ingenieur' in electrical engineering and the doctoral degree in technical science from the 'Technische Universität Wien' in 1978 and 1981, respectively. Dr. Selberherr has been holding the 'venia docendi' on 'Computer-Aided-Design' since 1984. Since 1988 he has been the head of the 'Institut für Mikroelektronik', and since 1999 dean of the 'Fakultät für Elektrotechnik und Informationstechnik'. His current topics are modeling and simulation of problems for microelectronics engineering. His e-mail address is: [selberherr@iue.tuwien.ac.at](mailto:selberherr@iue.tuwien.ac.at).

# ANISOTROPIC MESH ADAPTION GOVERNED BY A HESSIAN MATRIX METRIC

W. Wessner, H. Ceric, C. Heitzinger, A. Hössinger, and S. Selberherr  
Institute for Microelectronics  
Technical University of Vienna  
Gusshausstr. 27-29, A-1040 Vienna, Austria  
E-mail: Wessner@iue.tuwien.ac.at

## KEYWORDS

Diffusion Simulation, Anisotropic Mesh Refinement, Error Estimation, Finite Elements

## ABSTRACT

An essential task for any finite element method is to provide appropriate resolution of the mesh to resolve the initial solution. We present a computational method for anisotropic tetrahedral mesh refinement according to an adjustable discretization error. The initial attribute profile is given by an analytical function which is twice continuously differentiable. Anisotropy is taken into account to reduce the amount of elements compared to pure isotropic meshes. By the proposed method the spatial resolution in three-dimensional unstructured tetrahedral meshes used for diffusion simulation is locally increased and the accuracy of the discretization improved.

## INTRODUCTION

The generation of locally adapted tetrahedral meshes which carry the initial attribute profile is an important task of many modern algorithms in the finite element solution of partial differential equations, particularly for diffusion problems. The goal is to create and adapt a mesh which matches the initial attribute profile appropriately. This can only be done efficiently by anisotropic meshes.

Strict isotropic three dimensional regular meshes are not practicable for most realistic simulation structures especially in the field of semiconductor process simulation, which is our major application, because the resulting amount of tetrahedral elements required for a discretization with isotropic meshes is not practicable. The demand of calculation time and the limitation of memory require anisotropic adapted meshes which are more manageable.

The generation of tetrahedral meshes has to be enhanced by adaptation techniques where anisotropy plays a central position. The question then is how to identify these regions and how to obtain a good balance between the refined and unrefined regions such that the overall accuracy is optimal using a (nearly) minimal number of

grid points. These considerations clearly show the need for error estimators which can be extracted a posteriori from the computed numerical solution and the given data of the problem. A combination of error estimation and refinement mechanism is necessary to deliver higher accuracy, if needed, by increasing the spatial resolution.

## ANISOTROPIC REFINEMENT

One of the main methods for improving the spatial resolution is tetrahedral bisection which is well investigated by, e.g. (Arnold et al. 2000). When bisecting a tetrahedron, a particular edge – called the *refinement edge* – is selected and split into two edges by a new vertex, cf. Fig. 1. As new tetrahedra are constructed by refinement, their refinement edges must be selected carefully to take anisotropy into account without producing degenerately shaped elements. Bisecting a particular edge always influences the whole batch. To avoid ill shaped elements, the longest edge of the refinement tetrahedron is used as refinement edge.

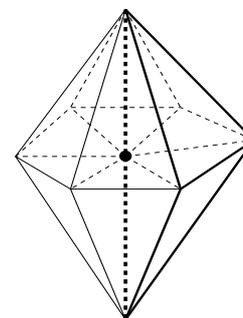


Figure 1: Bisecting.

It is obvious that the longest edge of one tetrahedron is not necessarily the longest edge of all attached tetrahedra. To bear down this problem, all new tetrahedra are directly tested and included into the refinement procedure.

A recursive approach for local mesh refinement which was suggested, e.g. in (Kossaczky 1994), cannot be applied to our situation due to the fact that anisotropy has to be taken into account.

To embrace anisotropy the basic idea of our refinement strategy is to calculate the length of an edge in a certain metric space (Lo 2001), i.e., the strain of the space varies from point to point with the consequence that the length of an edge depends on its position in the space. In case the *anisotropic edge length* is larger than an adjustable value, the edge is cut in the middle.

A set  $S$  with a global distance function (the metric  $g$ ) which for every two points  $x, y$  in  $S$  gives the distance between them as a nonnegative real number  $g(x, y)$  is called a metric space. A metric space must also satisfy

$$\begin{aligned} g(x, y) &= 0 \Leftrightarrow x = y \\ g(x, y) &= g(y, x) \\ g(x, y) + g(y, z) &\geq g(x, z). \end{aligned} \quad (1)$$

Calculating the length of an edge in a metric space can be seen as calculating a line integral. In general an arc length  $\ell_C$  is defined as the length along a curve  $C$ :  $\ell_C = \int_C ds$ . We define a symmetric and positive definite tensor  $\mathbf{M} = \mathbf{M}(x, y, z)$  over the entire domain, representing a Riemannian metric (Lo 2001). Roughly speaking, the metric tensor  $\mathbf{m}_{ij}$  determines how to compute the distance between any two points in a given space. Its components can be viewed as multiplication factors which must be placed in front of the differential displacements  $dx_i$  in a generalized Pythagorean theorem  $ds^2 = g_{11}dx_1^2 + g_{12}dx_1dx_2 + g_{22}dx_2^2 + \dots$ . A metric tensor at a point of the three-dimensional domain  $\Omega$  can be represented by a  $3 \times 3$  matrix  $\mathbf{M}$ . The length of a line segment  $\overline{PQ}$  in a metric space is calculated by (Borouchaki et al. 1997)

$$\ell_{PQ} = \int_0^1 \sqrt{\overline{PQ}^T \cdot \mathbf{M}(P + t \cdot \overline{PQ}) \cdot \overline{PQ}} dt \quad (2)$$

where  $\mathbf{M}(P + t \cdot \overline{PQ})$  is the metric at point  $P + t \cdot \overline{PQ}$ ,  $t \in [0, 1]$ .

In (Yamakawa and Shimada 2000) anisotropy is defined by three orthogonal principal directions and an aspect ratio in each direction. The three principal directions are represented by three unit vectors  $\vec{\xi}$ ,  $\vec{\eta}$ , and  $\vec{\zeta}$ , and in these directions the amounts of stretching of a mesh element are represented by three scalar values  $\lambda_\xi$ ,  $\lambda_\eta$ ,  $\lambda_\zeta$ , respectively. Using  $(\vec{\xi}, \vec{\eta}, \vec{\zeta})$  and  $(\lambda_\xi, \lambda_\eta, \lambda_\zeta)$  we define two matrices  $\mathbf{R}$  and  $\mathbf{S}$  by

$$\mathbf{R} := \begin{pmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{pmatrix} \text{ and } \mathbf{S} := \begin{pmatrix} \lambda_\xi & 0 & 0 \\ 0 & \lambda_\eta & 0 \\ 0 & 0 & \lambda_\zeta \end{pmatrix}. \quad (3)$$

By combining matrices  $\mathbf{R}$  and  $\mathbf{S}$ , we obtain a  $3 \times 3$  positive definite matrix  $\mathbf{M}$

$$\mathbf{M} := \mathbf{R}\mathbf{S}\mathbf{R}^T \quad (4)$$

that describes the three-dimensional anisotropy.

The crux of the matter is to find a suitable anisotropic tensor function which describes the stretching factors for a specific diffusion problem. Another problem is to find an error estimation which detects those regions where a higher spatial resolution is needed. An answer to these questions can be found when looking at the characteristics of the diffusion problem.

## DIFFUSION

Diffusion can be viewed as the transport of matter caused by a gradient of the chemical potential. This mechanism is responsible for the redistribution of dopant atoms in a semiconductor during a high-temperature processing step. The underlying ideas can be categorized into two major approaches, namely, the continuum theory of FICK's diffusion equation and the atomistic theory (Nishi and Doering 2000). We are using the continuum theory approach which describes the diffusion phenomenon by

$$\vec{J} = -D \cdot \text{grad}(C). \quad (5)$$

$\vec{J}$  denotes the diffusion flux,  $D$  is the *diffusion coefficient* or *diffusivity*, and  $C$  is the concentration of the dopant atoms. In general, the diffusion models used in semiconductor process simulation are strongly nonlinear, because the diffusion coefficients depend, e.g., on the impurity concentration and the point defects distribution (Kosik et al. 2000). These dependences result in coupled equation systems for impurities and point defects. Additionally, chemical reactions and convection problems have to be considered in the models. However, for better understanding of our refinement method we use the linear parabolic diffusion problem which is given by (5) for the following analysis.

There are mainly two discretization schemes for PDEs in complex domains namely the *finite element* method and the *finite volume* (finite box) method. In our diffusion simulator we use the Galerkin approach of the finite element method with linear shape functions and with backward Euler time discretization (Putti and Cordes 1998).

## GRADIENT FIELD

The gradient  $\nabla C = \text{grad}(C)$  of a scalar field  $C = C(x, y, z)$  in Cartesian coordinates is given by

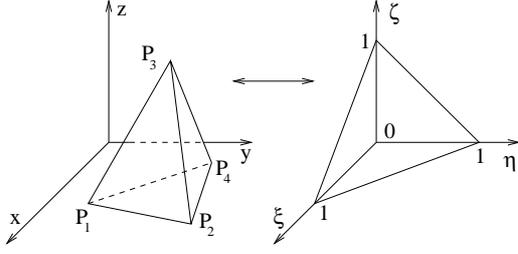
$$\nabla C = \frac{\partial C(x, y, z)}{\partial x} \vec{i} + \frac{\partial C(x, y, z)}{\partial y} \vec{j} + \frac{\partial C(x, y, z)}{\partial z} \vec{k}. \quad (6)$$

We are looking for (6) expressed through local coordinates on the three-dimensional unit simplex  $T$ . The gradient of a tetrahedral discretization can be calculated by using linear basis functions (Zienkiewicz and Taylor 1989) applied to the three-dimensional unit. The coordinate transformation

$$\begin{aligned} x &= x_1 + (x_2 - x_1)\zeta + (x_3 - x_1)\eta + (x_4 - x_1)\zeta \\ y &= y_1 + (y_2 - y_1)\zeta + (y_3 - y_1)\eta + (y_4 - y_1)\zeta \\ z &= z_1 + (z_2 - z_1)\zeta + (z_3 - z_1)\eta + (z_4 - z_1)\zeta \end{aligned} \quad (7)$$

allows to map an arbitrary tetrahedron at global coordinates  $(x, y, z)$  to the unit simplex  $T$  (cf. Fig. 2) with local element coordinates  $(\xi, \eta, \zeta)$ . In matrix notation this can be written as

$$\vec{r} - \vec{r}_1 = \mathbf{J} \cdot \vec{\delta}, \quad (8)$$



**Figure 2:** Coordinates transformation.

where  $\vec{r} = (x, y, z)^T$ ,  $\vec{r}_1 = (x_1, y_1, z_1)$ ,  $\vec{\delta} = (\xi, \eta, \zeta)^T$ , and  $\mathbf{J}$  denotes the JACOBIAN matrix

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \quad (9)$$

which applied to (7) results in

$$\mathbf{J} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{pmatrix}. \quad (10)$$

Using linear basis functions on the three-dimensional unit simplex (Bauer 1994), which are given by

$$\begin{aligned} N_1 &= 1 - \xi - \eta - \zeta, & N_2 &= \xi, \\ N_3 &= \eta, & N_4 &= \zeta, \end{aligned} \quad (11)$$

allows a linear approximation of the scalar field over the element in the form

$$C_T(\xi, \eta, \zeta) = \sum_{k=1}^4 N_k(\xi, \eta, \zeta) C_k, \quad (12)$$

where  $C_k$  denotes the scalar value of the solution on vertex  $k$  of the three-dimensional unit simplex  $T$ .

Applying (6) to the linear approximation, given by (12), results in

$$\nabla C_T(\xi, \eta, \zeta) = \begin{pmatrix} -C_1 + C_2 \\ -C_1 + C_3 \\ -C_1 + C_4 \end{pmatrix} \quad (13)$$

for the gradient of the spatial discretization element. Using the inverse of the transposed JACOBIAN matrix the gradient in global coordinates can now be expressed by :

$$\nabla C_T(x, y, z) = (\mathbf{J}^T)^{-1} \cdot \nabla C_T(\xi, \eta, \zeta). \quad (14)$$

It is in the nature of this approach that the gradient  $\nabla C_T(x, y, z)$  (14) is constant over an element  $T$  and forms a piecewise constant gradient field which gives a granular approximation of the proper gradient field given by (6).

Since the vector field (14) is piecewise constant, it is obvious that strong variations of the gradient from one element to an adjacent one yield an approximation error when compared to the proper continuous gradient field. This gradient approximation error causes a diffusion

flux error which gives rise to a violation of the law of mass conservation. The approximation gets worst if the changes of the gradient field are too high, i.e. the derivatives of the gradient field should be slight, or from another point of view, refinement should take place at spatial regions with high second derivatives of the initial scalar field.

## HESSIAN MATRIX

The basic idea of our refinement strategy is to use the HESSIAN matrix of the given (initial) solution for the anisotropic metric. The HESSIAN  $\mathbf{H}$  is given by

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f(x,y,z)}{\partial x^2} & \frac{\partial^2 f(x,y,z)}{\partial x \partial y} & \frac{\partial^2 f(x,y,z)}{\partial x \partial z} \\ \frac{\partial^2 f(x,y,z)}{\partial y \partial x} & \frac{\partial^2 f(x,y,z)}{\partial y^2} & \frac{\partial^2 f(x,y,z)}{\partial y \partial z} \\ \frac{\partial^2 f(x,y,z)}{\partial z \partial x} & \frac{\partial^2 f(x,y,z)}{\partial z \partial y} & \frac{\partial^2 f(x,y,z)}{\partial z^2} \end{pmatrix}. \quad (15)$$

In general the entries of the HESSIAN matrix are possibly negative. To use the HESSIAN as metric for the refinement strategy, a transformation has to be performed. This transformation is done simply with the norm of all function derivatives. The corresponding metric can be written as  $k \cdot \mathbf{m}_{ij} = |\mathbf{h}_{ij}|$  ( $i, j = 1, 2, 3$ ). To scale the metric a scalar factor  $k$  is used. This factor describes the maximum of the edge length in a static non-biased metric space where  $\mathbf{M} = \mathbf{I}$ , the identity matrix. The advantage of using the HESSIAN matrix is that it excellently reflects the curvature of the dopant profile and guarantees a good approximation in regions with high second derivatives (Gray 1998).

In our investigation of the method the HESSIAN must be given analytically which requires a twice continuously differentiable initial attribute profile. It can be shown that a large class of realistic profiles can be produced by linear combinations of twice differentiable functions.

## ERROR ESTIMATION

In order to measure the quality of a given three-dimensional discretization an error approximation is needed. According to the discussion of an interpolation error caused by using linear weighting functions (Johnson 1987), we calculate the linear approximation error

$$E_T = \left| \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} u(\xi, \eta, \zeta) d\xi d\eta d\zeta - \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} u_h(\xi, \eta, \zeta) d\xi d\eta d\zeta \right| \quad (16)$$

on a three-dimensional simplex  $T$ , where  $u(\xi, \eta, \zeta)$  denotes the given analytical profile and  $u_h(\xi, \eta, \zeta)$  the linear approximation over the three dimensional unit simplex  $T$ . For the diffusion problem the approximation error calculated by (16) can be seen as the diffusion particle difference over a tetrahedron. The difference is

caused by using piecewise linear functions to approximate the proper attribute profile. This approach is a usual one and suitable for sufficiently flat functions. For more strongly curved profiles the error of the approximation increases and a higher spatial resolution is needed.

### EXAMPLE

Diffusion, in the sense of an IC processing step, refers to the controlled forced migration of dopants into the substrate or adjacent material. The resulting doping profile which plays a major role in the performance of the integrated circuit, is affected by temperature and time as well as the temperature-time relationship during processing. Dopant atoms can be introduced into silicon in many ways. The most commonly used methods are (1) ion implantation and subsequent annealing or drive-in diffusion, (2) diffusion from a chemical source in vapor form at high temperatures, and (3) diffusion from a doped-oxide source (SMS1988). Since ion-implantation provides very precise control of the implanted profile, it is used to replace the chemical and doped-oxide sources wherever possible and is extensively applied in VLSI device fabrication.

The most common class of functions for the approximation of ion-implantation or drive-in diffusion profiles are GAUSSIAN probability distributions which are given by

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (17)$$

for a variate  $X$  with mean  $\mu$  and variance  $\sigma^2$  (Kenney and Keeping 1951).  $P(x) dx$  gives the probability that a variate with a GAUSSIAN distribution takes on a value in the range  $[x, x + dx]$ .

The HESSIAN matrix (15) can then be built easily from the dopant profile approximation given by (17) and used for the anisotropy metric (4).

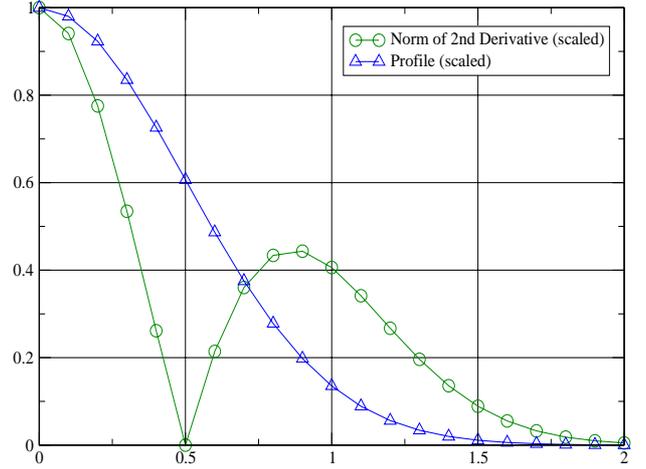
Such profiles can occur, e.g., after a diffusion with a constant dopant atoms dose. For a one-dimensional case this can be written as

$$N_d = \int_0^\infty C(x, t) dx = \text{const.} \quad (18)$$

This diffusion condition is referred to as *drive-in diffusion* (Shewmon 1989).

To see the essential impact of our refinement strategy we use a three-dimensional test structure. The underlying initial mesh (see Fig. 4(a)) is a coarse isotropic mesh which carries a normalized GAUSSIAN profile (see Fig. 3) which could occur, e.g., after diffusion with a constant dopant atoms dose.

Note that after a drive in diffusion process step the gradient of the concentration  $C$  vanishes at the surface (left end of the test structure, see Fig. 4(a)),



**Figure 3:** Test profile and norm of the second derivation (scaled to one) along the  $x$  direction of the mesh structure.

$\nabla C = \text{grad}(C) = 0$ , and so does the diffusion flux  $\vec{J}$  (5). Therefore the initial dopant concentration has its maximum at the left of the structure.

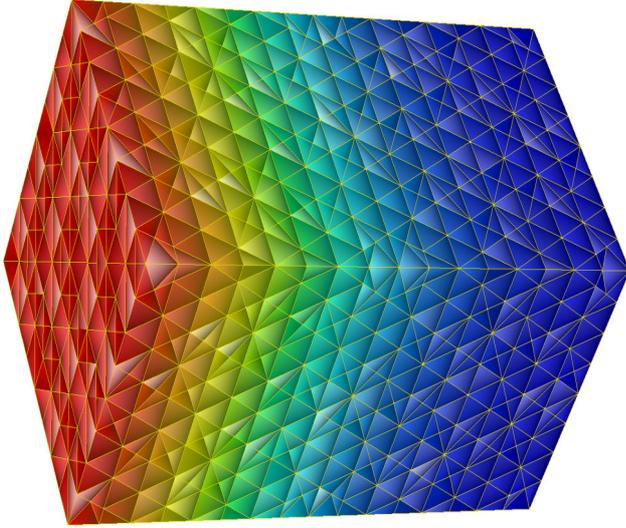
In this case the profile is one-dimensional and has therefore only one non-zero HESSIAN entry  $\frac{\partial^2 f(x, y, z)}{\partial x^2}$ , cf. (15). The corresponding scaled shape of the curve can be seen in Fig. 3. Using the HESSIAN matrix as input for the refinement procedure it is guaranteed that the refinement takes place where the initial solution shows a strong curvature and leaves regions with low curvature untouched.

Fig. 4(a) shows the initial coarse mesh. The tetrahedral structure is strict isotropic and mostly regular. The color of the structure corresponds to the one-dimensional initial solution which is shown in Fig. 3.

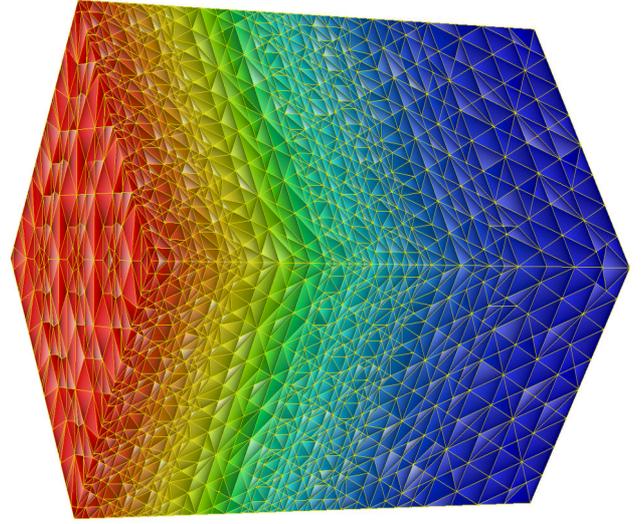
After the anisotropic refinement which is based on the HESSIAN matrix of the attribute-profile function, the refinement takes place only in regions of high curvature as shown in Fig. 4(b). The anisotropy is restricted to the  $x$ -direction of the test structure while others directions are not influenced.

Fig. 5 shows the mesh valuation according to the linear approximation error, cf. (16). We used a one-dimensional cut along the  $x$  direction through the test-structure and the error evaluation was performed along this one-dimensional cut.

At the initial mesh the error varied according to the curvature of the profile. In the regions where the profile shows a flat behavior the error is very small and therefore in this areas no refinement is needed. The shape of the error curve reflects the shape of the norm of the second derivative of the initial profile (see Fig. 3).

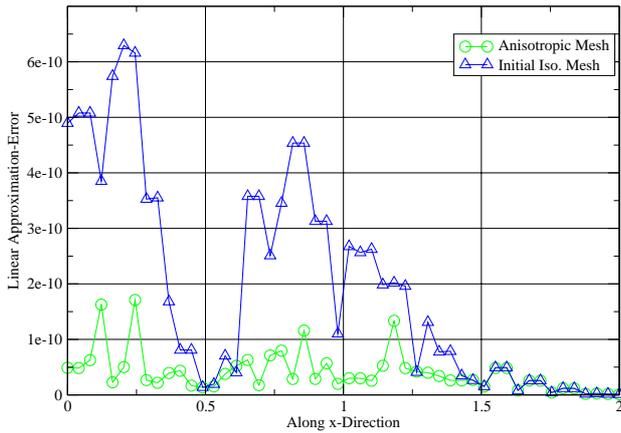


(a) Isotropic coarse mesh (initial mesh).



(b) Anisotropic fine mesh (after refinement).

**Figure 4:** Mesh test structure before and after anisotropic refinement.



**Figure 5:** Approximation error  $E_T$ : one-dimensional cut through initial mesh and refined anisotropic mesh test structure.

After the refinement a rigorous reduction of the error is obtained. The variation along the  $x$ -direction has vanished and a good initial solution approximation over the whole domain was reached.

## CONCLUSION

We present a computational method for locally conformal anisotropic tetrahedral mesh refinement according to an adjustable discretization error. The goal of our procedure is to create a mesh which matches an analytically given initial attribute profile appropriately. This is essential for

accurate three-dimensional diffusion process simulation which is a key process of semiconductor device manufacturing.

The basic refinement step is tetrahedral bisection which guarantees a conformal mesh during refinement and allows easily local mesh adaption. A special mathematical torsion of a metric space is used to take anisotropic structures into account. This enables the reduction of elements compared to strict isotropic refinement.

It is obvious that in regions with high curvature of the initial profile the approximation error also shows high values. The refinement procedure detects those regions and uses the HESSIAN matrix of the profile for the metric space torsion. This guarantees a target oriented local mesh refinement and keeps the amount of additional mesh points small.

In our refinement procedure the initial attribute profile has to be given by an analytically function which is twice continuously differentiable. At the first glance this is a loss of generality but a wide range of realistic diffusion and ion implantation profiles can be approximated with GAUSSIAN distribution functions. These function classes are continuously twice differentiable and therefore perfect to form the HESSIAN matrix.

Our algorithm shows good local behavior and reflects the curvature of the initial profile excellently. The error estimations shows that the accuracy can be improved drastically by target oriented refinement. To find a good balance between refined and unrefined regions such that the overall accuracy is optimal using a (nearly) minimal number of grid points, the error estimation must be more problem oriented and reflect the nature of finite elements.

## REFERENCES

- Arnold, Douglas N.; Arup Mukherjee; and Luc Pouly, 2000, "Locally Adapted Tetrahedral Meshes Using Bisection". *SIAM J. Sci. Comput.*, 22(2):431–448.
- Bauer, Robert, November 1994, *Numerische Berechnung von Kapazitäten in dreidimensionalen Verdrahtungsstrukturen*. Dissertation, Institut für Mikroelektronik, Technische Universität Wien. <http://www.iue.tuwien.ac.at/>.
- Borouchaki, Houman; Paul Louis George; Frederic Hecht; Patrick Laug; and Eric Saltel, 1997, "Delaunay Mesh Generation Governed by Metric Specifications. Part I. Algorithms". *ELSEVIER Finite Elements in Analysis and Design*, 25:61–83.
- Gray, Alfred, 1998, *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press LLC, 2000 Corporate Blvd., N.W., Boca Raton, Florida 33431, USA, first edition.
- Johnson, Claes, 1987, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 32 East 57th Street, New York, NY 10022, USA.
- Kenney, J.F. and E.S. Keeping, 1951, *Mathematics of Statistics*, volume 2. Van Nostrand Co., New York, NY, USA, second edition.
- Kosik, Robert; Peter Fleischmann; Bernhard Haindl; Paola Pietra; and Siegfried Selberherr, November 2000, "On the Interplay Between Meshing and Discretization in Three-Dimensional Diffusion Simulation". *IEEE Trans. Computer-Aided Design*, 19(11):1233–1240.
- Kossaczky, Igor, 1994, "A Recursive Approach to Local Mesh Refinement in Two and Three Dimensions". *ELSEVIER Journal of Computational and Applied Mathematics*, 55:275–288.
- Lo, S.H., 2001, "3D Anisotropic Mesh Refinement in Compliance with a General Metric Specification". *ELSEVIER Finite Elements in Analysis and Design*, 38:3–19.
- Nishi, Yoshio and Robert Doering, 2000, *Handbook of Semiconductor Manufacturing Technology*. Marcel Dekkar, Inc., 270 Madison Avenue, New York, NY 10016, first edition.
- Putti, M. and Ch. Cordes, 1998, "Finite Element Approximation of the Diffusion Operator on Tetrahedra". *SIAM J. Sci. Comput.*, 19(4):1154–1168.
- Shewmon, Paul, 1989, *Diffusion in Solids*. Minerals, Metals & Materials Society, 420 Commonwealth Drive, Warrendale, Pennsylvania 15086, USA, second edition.
- Simon M. Sze, editor, 1988, *VLSI Technology*. McGRAW-HILL Science/Engineering/Math, Shoppenhangers Road, Maidenhead, Berkshire, England, second edition.
- Yamakawa, Soji and Kenji Shimada, 2000, "High Quality Anisotropic Tetrahedral Mesh Generation via Ellipsoidal Bubble Packing". *Proceedings of 9th IMR*, pages 263–273.
- Zienkiewicz, O.C. and R.L. Taylor, 1989, *The Finite Element Method*, volume 1. McGRAW-HILL Book Company Europe, Shoppenhangers Road, Maidenhead, Berkshire, England, fourth edition.

## AUTHOR BIOGRAPHIES



**WILFRIED WESSNER** was born in Horn, Austria, in 1977. He studied computer engineering at the 'Technische Universität Wien', where he received the degree of 'Diplom-Ingenieur' (with honors) in 2002. He joined the 'Institut für Mikroelektronik' in summer 2002, where he is currently working on his doctoral degree. His scientific interests include three-dimensional mesh generation, anisotropic mesh adaption, computational geometry, and data visualization. His e-mail address is: [wessner@iue.tuwien.ac.at](mailto:wessner@iue.tuwien.ac.at).



**HAJDIN CERİĆ** was born in Sarajevo, Bosnia and Hercegovina, in 1970. He studied electrical engineering at the Electrotechnical Faculty of the University of Sarajevo and 'Technische Universität Wien', where he received the degree of 'Diplom-Ingenieur' in 2000. He joined the 'Institut für Mikroelektronik' in June 2000, where he is currently working on his doctoral degree. His scientific interests include interconnect and process simulation. His e-mail address is: [ceric@iue.tuwien.ac.at](mailto:ceric@iue.tuwien.ac.at).



**CLEMENS HEITZINGER** was born in Linz, Austria, in 1974. He studied Technical Mathematics at the 'Technische Universität Wien', where he received the degree of 'Diplom-Ingenieur' (with honors) in 1999. He joined the 'Institut für Mikroelektronik' in February 2000. From March to May 2001 he held a position as visiting researcher at the Sony Technology Center in Hon-Atsugi (Tokyo, Japan). He received the doctoral degree in technical sciences (with honors) from the 'Technische Universität Wien' in 2002. In January 2003 he was awarded an Erwin Schrödinger Fellowship by the Austrian Science Fund (FWF) for the project entitled *Mathematical Models for Nanoscale Semiconductor Device Engineering*. In March 2003 he held a position as visiting researcher at Cypress Semiconductor in San Jose (CA, USA). His scientific interests include applied mathematics for process and device simulation. His e-mail address is: [heitzinger@iue.tuwien.ac.at](mailto:heitzinger@iue.tuwien.ac.at).



**ANDREAS HÖSSINGER** was born St. Pölten, Austria, in 1969. He studied technical physics at the 'Technische Universität Wien', where he received the degree of 'Diplom-Ingenieur' in January 1996. He joined the 'Institut für Mikroelektronik' in June 1996. In 1998 he held a visiting research position at Sony in Atsugi, Japan. In September 2000 he finished his Ph.D. degree at the 'Institut für Mikroelektronik' where he is currently enrolled as a post-doctoral researcher. In 2001 he held a position as visiting researcher at LSI Logic in Santa Clara, CA, USA within the scope of a cooperate research project on three-dimensional process simulation. In 2001 he also received a grant from Austrian Academy of Science within the scope of the Austrian Program for Advanced Research and Technology for his work on three-dimensional process simulation. His e-mail address is: [hoessinger@iue.tuwien.ac.at](mailto:hoessinger@iue.tuwien.ac.at).



**SIEGFRIED SELBERHERR** was born in Klosterneuburg, Austria, in 1955. He received the degree of 'Diplom-Ingenieur' in electrical engineering and the doctoral degree in technical science from the 'Technische Universität Wien' in 1978 and 1981, respectively. Dr. Selberherr has been holding the 'venia doceni' on 'Computer-Aided-Design' since 1984. Since 1988 he has been the head of the 'Institut für Mikroelektronik', and since 1999 dean of the 'Fakultät für Elektrotechnik'. His current topics are modeling and simulation of problems for microelectronics engineering. His e-mail address is: [selberherr@iue.tuwien.ac.at](mailto:selberherr@iue.tuwien.ac.at).

# COLOR ANT POPULATIONS ALGORITHM FOR DYNAMIC DISTRIBUTION IN SIMULATIONS

Cyrille Bertelle, Antoine Dutot, Frédéric Guinand, Damien Olivier

Laboratoire d'Informatique du Havre  
Université du Havre  
25 rue Philippe Lebon  
76600 Le Havre

email:{Cyrille.Bertelle,Antoine.Dutot,Frederic.Guinand,Damien.Olivier}@univ-lehavre.fr

## KEYWORDS

Ant algorithms, dynamic computation, communication graph, clustering, auto-organization

## ABSTRACT

Most distributed simulations are applications composed of numerous mobile communicating entities that continuously evolve. Such entities are organized or organize themselves in groups or societies of cooperating agent or processes. To keep these simulations efficient, it may be necessary to migrate entities of highly communicating groups so that they remain close, ideally on the same processing unit, while at the same time, we need to maintain a reasonable load for each processor.

We have proposed (Bertelle et al., 2002b) a method to hint migration of entities making a trade-off between communication and load-balancing based on a variant of ant algorithms. We use a dynamic communication graph to model distributed simulations where vertices represent entities and edges communications. Several colonies of ants, each of a distinct color representing a computing resource, compete to mark vertices of the graph using colored pheromones. A color change on a vertex hints the entity it represents to migrate on the corresponding processor. In some cases, the solution obtained is not satisfying. We try to show in this paper the reasons and to give improvements to solve it.

## INTRODUCTION

Simulation are often used to study complex systems (ecosystems, traffic road ...). In such simulations there are a very large number of entities whose behavior and interactions describe the evolution of the system. Such models are implicitly distributed due to their heterogeneous swarm nature. This kind of problem does not allow a static placement, we are in front of a non-predictable aspect.

However, when trying to distribute such systems, communications flows and computing resource load may become problematic for correct execution of the simulation. In this article we propose a way to hint entities on locations that try to reduce communication impact on system execution by migrating entities taking care of load-balancing.

The paper is organized as follows: Section describes the graph model that underlies our system. Section provides some background informations on ant algorithms whereas section describe our specific colored ant system and some preliminary results. Section presents our implementation, some experiments, problems encountered with the initial colored ant system, and enhancements made to it. Finally we conclude with further expected improvements and perspectives.

## DYNAMIC COMMUNICATION GRAPH

### Model

We model the simulation by a graph  $G = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  is a set of vertices representing entities of the application and  $\mathcal{E} = \mathcal{V} \times \mathcal{V}$  is a set of edges  $e = (v_1, v_2)$  representing communications between entities  $v_1$  and  $v_2$ . We consider communications as being bidirectional and therefore the graph is undirected. Edges are labeled by communication volumes. Each vertex is assigned to an initial processing resource at start. This initial distribution is fixed by the application and for the model we consider it as random.

We call *actual communications*, communications between entities that are not located on the same processor. We try to limit the number of these actual communications by identifying clusters of highly communicating entities (organisations). However merely trying to avoid actual communications could simply migrate all entities on a single processor. Therefore we also need to do some load balancing. Colored pheromones have been introduced for this purpose.

## Dealing with a Dynamic Environment

The graph model we defined is dynamic, communication volumes may vary, vertices and edges may appear or disappear. These changes in both topology and valuation are one of the major motivation for using ant algorithms.

Indeed, a monotonic approach is one way to achieve clustering on a graph. It consists in regularly applying a computation on a frozen copy of the dynamic graph, then trying to use this information, though the real graph is still evolving. This approach is problematic: the graph can have changed during computation and results may not be usable any more, creating discrepancies between the real application state and calculated migration hints. Furthermore, it is not incremental, each time the algorithm is performed.

Another way is to use an anytime algorithm. The dynamic graph is considered as a changing environment for computing entities that travel on the graph, taking into account the changes as they appear, and storing the solution directly in the graph, as a side effect of their evolution. Ant algorithms are well suited for that task as it has been shown in (Dorigo et al., 1996).

## ANT ALGORITHMS

Our method is based on ant algorithms. These techniques are part of meta-heuristics approaches that yield near-optimal solutions to hard optimisation problems where algorithms that provide an exact solutions are not an issue. Ant algorithms maintain a population of agents, that exhibit a cooperative behaviour (Langton, 1987), by continuously foraging their territories to find food (Gordon, 1995) using optimal paths, creating bridges, constructing nests, etc.

Such self-organization appears from interactions that can be either direct or indirect. Direct communication is done via antennation, trophallaxis, any sort of contact (mandibular, visual, chemical). Indirect communications arise from individuals changing the environment and other responding to these changes: this is called *stigmergy*. For example, ants deposit signals named *pheromones* in the environment that influence others: the more pheromone on a path, the more ants tend to follow it. As pheromones evaporate, long paths tend to have less pheromone than short ones, and therefore are less used than others.

Such an approach is robust and well supports parameter changes in the problem. Besides, it is intrinsically distributed and scalable. It uses only local informations (required for a continuously changing environment), and find near-optimal solutions. Ant algorithms has been applied successfully to various combinatorial optimization problems like the Travelling Salesman Problem (Dorigo and Gambardella, 1997) or routing in networks (Caro and Dorigo, 1997), (White, 1997), but also to DNA sequencing (Bertelle et al., 2002a), graph partitioning (Kuntz

et al., 1997) and clustering (Faieta and Lumer, 1994).

## COLORED ANT SYSTEM

As shown above, we model large scale distributed applications by a dynamic graph  $G = (\mathcal{V}, \mathcal{E})$  where vertices represent entities of the simulations and edges communication between these entities. Edges are labeled by communication volume. The ant algorithm is used to detect clusters of highly communicating entities. To solve load balancing problems we introduce *colored ants* and *colored pheromones* that correspond to available processors. To support our algorithm we extend our graph definition:

### Definition 1 (Dynamic communication colored graph)

A dynamic communication colored graph is a weighted undirected graph  $G = (\mathcal{V}, \mathcal{E}, \mathcal{C})$  such that:

- $\mathcal{C}$  is a set of  $p$  colors where  $p$  is the number of processors of the distributed system.
- $\mathcal{V}$  is the set of vertices. Each vertex has a color belonging to  $\mathcal{C}$ .
- $\mathcal{E}$  is the set of edges. Each edge is labelled with a weight. A weight  $w(u, v) \in \mathbb{N}^+$  associated with an edge  $(u, v) \in \mathcal{V} \times \mathcal{V}$  corresponds to a volume of communications between the couple of entities corresponding to vertices  $u$  and  $v$ .

The figure 1 shows an example of a dynamic communication colored graph. The proposed method changes the color of vertices if this change can improve communications or processor load. The algorithm tries to color vertices of highly communicating clusters with the same colors. Therefore a vertex may change color several times, depending on the variations of data exchange between entities.

## The Colored Ant Algorithm

Our algorithm is inspired by the Ant System (Dorigo et al., 1996). We used it as a base for further improvements that we detail in the section . We consider a dynamic communication colored graph  $G = (\mathcal{V}, \mathcal{E}, \mathcal{C})$ .

1. Each processing resource is assigned to a color. Each vertex gets its initial color from the processing resource where it appears. For each processor of color  $c \in \mathcal{C}$  a number of ants proportional to the processor power is allocated and are uniformly placed on vertices of their color.  $\mathcal{F}$  denotes the set of all ants. An ant of color  $c$  deposits pheromones of color  $c$ .
2. The algorithm is based on an iterative process. Between steps  $t - 1$  and  $t$ , each ant crosses one edge and reaches a new vertex. During its move, it drops

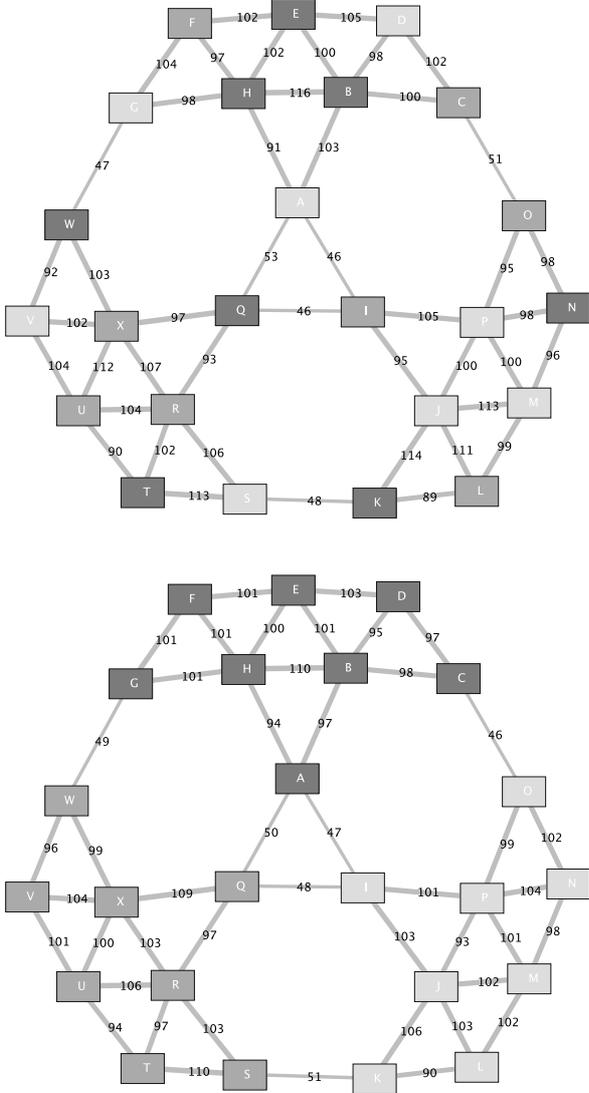


Figure 1: Example of a dynamic communication graph, at start ( $t = 0$ ), and clustered ( $t = 45$ ).

pheromone of its color on the crossed edge. Moreover, each ant has the ability to remember the vertex it comes from.

We define the following positive numbers:

- The quantity of pheromone of color  $c$  dropped by one ant  $x$  on the edge  $(u, v)$ , between the steps  $t - 1$  and  $t$  is noted  $\Delta_x^{(t)}(u, v, c)$ .
- The quantity of pheromone of color  $c$  dropped by the ants when they cross edge  $(u, v)$  between steps  $t - 1$  and  $t$  is noted:

$$\Delta^{(t)}(u, v, c) = \sum_{x \in \mathcal{F}} \Delta_x^{(t)}(u, v, c) \quad (1)$$

- The total quantity of pheromone of all colors dropped by ant on edge  $(u, v)$  between steps  $t - 1$  and  $t$  is noted:

$$\Delta^{(t)}(u, v) = \sum_{c \in \mathcal{C}} \Delta^{(t)}(u, v, c) \quad (2)$$

- If  $\Delta^{(t)}(u, v) \neq 0$ , the rate of pheromone of color  $c$  on the edge  $(u, v)$  between the steps  $t - 1$  and  $t$  is noted

$$K_c^{(t)}(u, v) = \frac{\Delta^{(t)}(u, v, c)}{\Delta^{(t)}(u, v)} \quad (3)$$

This rate verifies  $K_c^{(t)}(u, v) \in [0, 1]$ .

3. The current quantity of pheromone of color  $c$  present on the edge  $(u, v)$  at step  $t$  is denoted by  $\tau^{(t)}(u, v, c)$ . Its initial value (when  $t = 0$ ) is 0 and then is computed following the recurrent equation:

$$\tau^{(t)}(u, v, c) = \rho \tau^{(t-1)}(u, v, c) + \Delta^{(t)}(u, v, c)$$

$\rho \in [0, 1]$  represents the pheromone persistence due to its evaporation.

4. At this stage of the algorithm, we have computed the current quantity of pheromone,  $\tau^{(t)}(u, v, c)$ , in classically, as reinforcement factor for clustering formation based on colored paths. We need now to take into account the load balancing in this auto-organization process. For this purpose, we need to balance this reinforcement factor with  $K_c^{(t)}(u, v)$ , the relative importance of considered color with regard to all other colors. This corrected reinforcement factor is noted:

$$\omega^{(t)}(u, v, c) = K_c^{(t)}(u, v) \tau^{(t)}(u, v, c)$$

Unfortunately, this corrected reinforcement factor can generate too instable process. So we prefer to use a delay-based relative importance of considered color

with regard to all other colors. For a time range  $q \in \mathbb{N}^+$ , we define:

$$K_c^{(t,q)}(u, v) = \sum_{s=t-q}^t K_c^{(s)}(u, v). \quad (4)$$

According to this definition, we compute the new corrected reinforcement factor :

$$\Omega^{(t)}(u, v, c) = K_c^{(t,q)}(u, v) \tau^{(t)}(u, v, c) \quad (5)$$

5. Let us defines  $p(u, v_k, c)$  the transition probability of an edge  $(u, v_k)$  incident to vertex  $u$  for an ant of color  $c$  and whose communication volume is noted  $w(u, v_k)$ .

- At the initial step ( $t = 0$ ),

$$p(u, v_k, c) = \frac{w(u, v_k)}{\sum_{v \in \mathcal{V}_u} w(u, v)} \quad (6)$$

- After the initial step ( $t \neq 0$ ),

$$p(u, v_k, c) = \frac{(\Omega^{(t)}(u, v_k, c))^\alpha (w(u, v_k))^\beta}{\sum_{v_q \in \mathcal{V}_u} (\Omega^{(t)}(u, v_q, c))^\alpha (w(u, v_q))^\beta} \quad (7)$$

Where  $\mathcal{V}_u$  is the set of vertices adjacent to  $u$ .

The relative values of  $\alpha$  and  $\beta$  give the weighting between pheromone factor and communication volumes. We will see later that this weighting is a major factor in the way the algorithm achieves its goals.

The choice of the next edge crossed by an ant depends on the previous probabilities. However, to avoid the ant moves to oscillate between two vertices, we introduce in the formula a penalisation factor  $\eta \in [0, 1]$ . Given  $\bar{v}_x$  the last vertex visited by ant  $x$ , the new probability formula is:

$$p_x(u, v_k, c) = \frac{(\Omega^{(t)}(u, v_k, c))^\alpha (w(u, v_k))^\beta \eta_{x,k}}{\sum_{v_q \in \mathcal{V}_u} (\Omega^{(t)}(u, v_q, c))^\alpha (w(u, v_q))^\beta \eta_{x,q}} \quad (8)$$

Where

$$\eta_{x,q} = \begin{cases} 1 & \text{if } v_q \neq \bar{v}_x \\ \eta & \text{if } v_q = \bar{v}_x \end{cases} \quad (9)$$

6. The color of a vertex  $u$ , noted  $\xi(u)$  is obtained from the main color of its incident arcs:

$$\xi(u) = \arg \max_{c \in \mathcal{C}} \sum_{v \in \mathcal{V}_u} \tau^{(t)}(u, v, c) \quad (10)$$

## Solution Quality

It is necessary to have a measure of the quality of the solution, to know if we must continue to search a solution. They are two aspects to take into account :

- The global costs of communications;
- The load-balancing of the application.

They are antagonist. So, in order to evaluate our solution we defined two quality criterions  $r_1$  and  $r_2$ . The first criterion  $r_1$  allows to know between two solutions which has proportionally less actual communications. Thus we compute global communication costs, noted  $e$ , by summing actual communications on the graph (between entities located on distinct processors). Then we compute a ratio  $r_1$  among the total volume of communications, noted  $s$ , on the graph and we have:

$$r_1 = e/s$$

The more  $r_1$  is close to 0, the more actual communications are low, as expected. The second criterion  $r_2$  considers the load-balancing. For each color  $c$ , we have  $v_c$  the number of vertices having color  $c$  and  $p_c$  the power of processor affected to  $c$ . Then we have:

$$r_2 = \frac{\min \mathcal{E}}{\max \mathcal{E}} \quad \text{where } \mathcal{E} = \left\{ \frac{v_c}{p_c}; c \in C \right\}$$

The more  $r_2$  is close to 1, the more load-balancing is good. For example, we obtain on the two graphs (fig. 1)  $r_1 = 0.7$ ,  $r_2 = 1.0$  for  $t = 0$  and  $r_1 = 0.1$ ,  $r_2 = 1.0$  for  $t = 45$ .

These criterions are used to compare different solutions obtained during the computation, essentially to verify if we improve the solution during the steps. These criterions, enable us to store the best solution obtained so far.

We use also these criterions to compare communication graphs where clusters are already identified. These graphs are randomly perturbed in term of color allocation to fix an initial configuration. After the algorithm tries to find the initial allocation on the graph, as a solution or a solution where the criterions are closest.

## Ant Populations Algorithm

The original algorithm reveals several difficulties due to lack of control over the relations between the numerous parameters. In clusters of high communication, ants tend to follow privileged paths that form *loops*. This is due to the fact communications in such areas are mostly the same and pheromone take a too large importance in ant path choices. Such paths exclude some nodes that could be settled and leads to three problems: *grabs*, *starvation* and *overpopulation* as explained under and shown in figures 2, 3 and 4. In these figures, the graph representation

is as follows. Vertices are rectangles. Edges are shown with a pie chart in the middle that indicates relative levels of pheromones with the maximum pheromone level numbered. Vertices are labeled by their name at the top with under at the left the total number of ants they host and at the right a pie chart indicating the relative number of ants of each color present on this vertex. These figures are excerpt from the communication graph corresponding to figure 1.

**grabs** Small ants group of a given color are locked in an area of the graph by a larger colony of another color that surround them. In this case they cannot escape due to repulsion factors and cannot help consolidating correct clusters of their own color elsewhere (figure 2).

**starvation** Whole parts of the graph get less and less pheromone trails and are left unoccupied by ants. As pheromone evaporate, less and less ants are attracted by them (figure 3).

**overpopulation** Some parts of the graph get too much ants as the total number of ants on a vertex is not bounded. In the worse case, ants of all colors can appear in such high density populations without the repulsion factors to operate correctly (figure 4).

The base algorithm gives useful results if its parameters are well set, however fixing them is delicate. If they are not correctly configured the algorithm may well fall in local minima as shown in examples above: whether pheromones grow too quickly and overpopulation appears, or they evaporate too fast and starvation comes into play. These two cases leading to grabs.

Control on the ant population as been changed. Before, this control was used only to handle the dynamic graph (vertex and edge appearance or disappearance). Now we add some death and hatching mechanisms. We perturb the ants repartition generating small stable clusters which are the result of local minima. Furthermore this procedure makes senses since our algorithm runs continuously not to find a static solution as the standard Ant System, but to provide anytime solutions to a continuously changing environment.

Including death and hatching ask a question about population: for a given number of vertices is it constant or not? Indeed we want to avoid cases where all ants disappear or too many ants appear. Therefore, we resolved to make one hatch for one death.

Additions to the original Colored And Algorithm are:

1. We define the following positive numbers:
  - $\tau^{(t)}(u, c)$  is the quantity of pheromone of color  $c$  dropped on all edges connected to vertex  $u$ :

$$\tau^{(t)}(u, c) = \sum_{v_q \in \mathcal{V}_u} \tau^{(t)}(u, v_q, c) \quad (11)$$

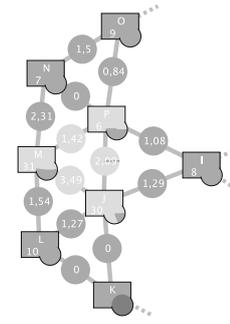


Figure 2: *Grab*: red ants are unable to move to the red cluster (not shown here), being hold in a small loop by blue ants. Both groups of ants are responsible: red ants are attracted by their own ever growing loop of pheromone, and blue ants repulse red ones forcing them to remain on the loop.

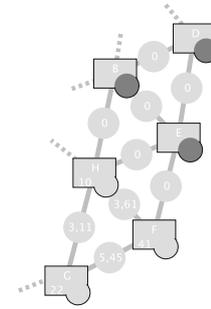


Figure 3: *Starvation*: red ants remain on a small loop at the bottom, leaving a whole part of a high communication group unoccupied.

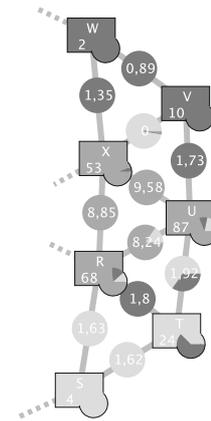


Figure 4: *Overpopulation*: the original distribution allocated 20 ants on each vertex. Here there are more than 50 ants on each blue vertex running in a loop. Several other parts (not shown here) of the graph are empty. This problem is due to bad parameters  $\alpha$  and  $\beta$ .

- $\tau^{(t)}(u)$  is the quantity of pheromone of all colors dropped on all edges connected to vertex  $u$ :

$$\tau^{(t)}(u) = \sum_{c \in \mathcal{C}} \tau^{(t)}(u, c) \quad (12)$$

- $\varphi_c(u) \in [0, 1]$ :

$$\varphi_c(u) = \frac{\tau^{(t)}(u, c)}{\tau^{(t)}(u)} \quad (13)$$

the relative importance of pheromones of color  $c$  compared to pheromones of all colors on edges leading to vertex  $u$ .

2. Then, at each step, before the ant chooses an arc to cross (equations (6) to (8)), we must choose whether the ant will die or not. We determine this using a threshold parameter  $\phi \in [0, 1]$  (preferably small, under 0.1) for an ant of color  $c$  on vertex  $u$ :

- if  $\varphi_c(u) < \phi$  we make the ant die and create a new ant choosing a new location for it as follows. We select randomly a set  $\mathcal{V}_n$  of  $n$  vertices. Let  $\text{card}(\mathcal{F}(v))$  be the number of ants on vertex  $v$ . Then we select a vertex  $u$  in  $\mathcal{V}_n$  using:

$$u = \arg \min_{v \in \mathcal{V}_n} (\text{card}(\mathcal{F}(v))) \quad (14)$$

and make the new ant hatch on it.

- else, we proceed as specified in the original algorithm choosing a new edge using probabilities (equation (6) and following).

This procedure eliminates grabs and starvation. Grabbed ants die, and hatch in starvation areas. However it does not eliminate loops, and sometimes loops tend to reappear. In order to break them, we introduce more memory in ants: instead of being able to memorize only one vertex, ants can memorize three or more vertices.

## IMPLEMENTATION AND EXPERIMENTATION

In these experiments dynamic valuations of the edges are not shown, we concentrate on the pheromone levels, edge coloration and ant populations.

### Results of the Base CAS

We tried the CAS on the graph given in figure 1. This particular graph is problematic. Finding good parameters for it was difficult. The algorithm stagnating in a solution not very closed to the optimal solution (see figure 5 with parameters:  $\alpha = 1$ ,  $\beta = 4$ ,  $\eta = 0.0001$ ,  $\rho = 0.8$ ). This configuration appears after 30 steps and stays identical after 570 more steps.

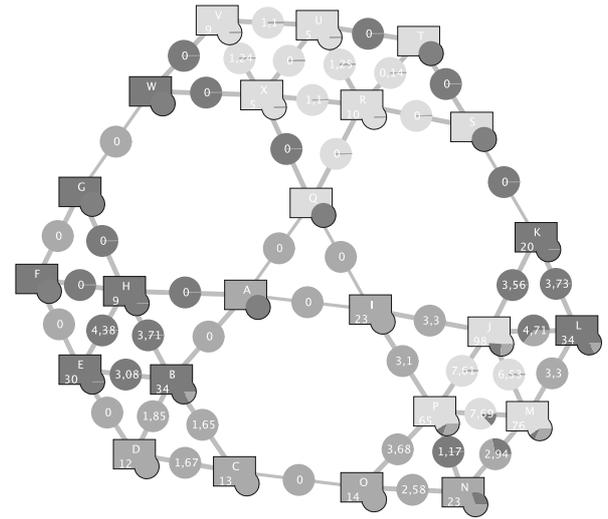


Figure 5: The original algorithm on a problematic graph after 600 steps

### Results With Ant Population Control

We then used the new algorithm with population control on the same problem with the same parameters plus the two new parameters:  $\phi = 0.3$  and the memory set to 4. A correct solution appeared after 23 steps as shown in figure 6. After 200 steps the configuration was stable (figure 7),

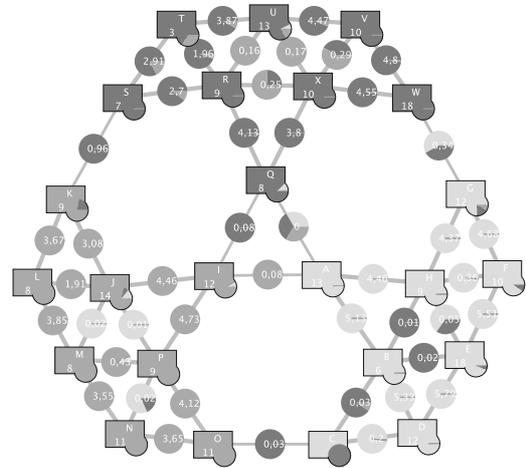


Figure 6: The algorithm with population control after 23 steps.

and remained the same at 600 steps (figure 8) where the experiment was stopped.

We also tested the algorithm with grid like graphs like shown in figure 9 and 10. Parts of the grid are subgrids of high communication. A good solution was found after

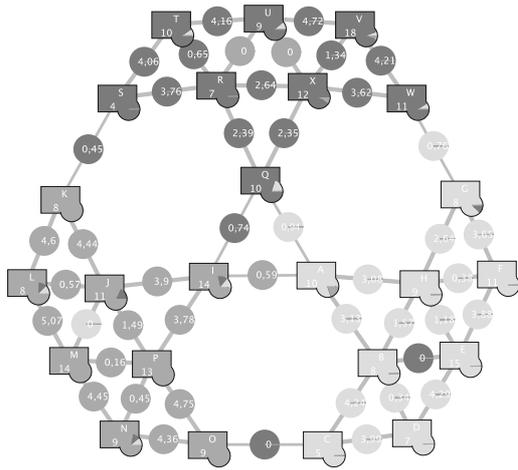


Figure 7: The algorithm with population control after 200 steps.

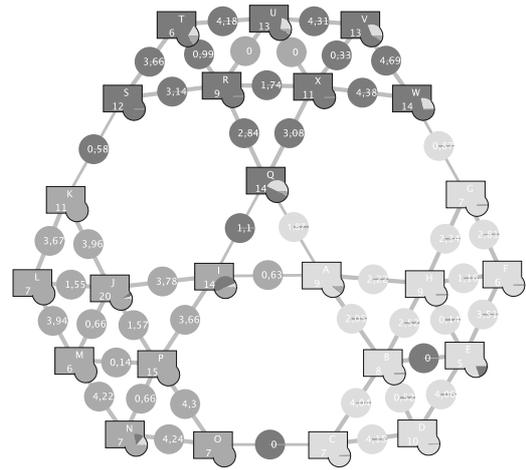


Figure 8: The algorithm with population control after 600 steps.

50 steps and stabilized at 100 steps.

We designed a software tool that allows us to edit graphs and modify them as the ant algorithm performs, adding or removing edges and nodes at runtime. The graph shown in figures 11, 12, and 13 is a modification of 1. we also obtain good results, though load balancing is here not optimal. Playing with parameters or adding a new processor (new color) breaks the biggest cluster in two or three (placing some nodes of it in two other colors, or allocating a new color to it following the solution chosen).

## CONCLUSION

In this paper we presented a variant of the Ant System called Colored Ant System that offers advices for entity migration in a distributed system taking care of the load and communication balancing. We described a base colored ant algorithm and then provided several improvements and shown their results.

This works aims at improving repartition of hydrosystem simulations that use a large number of distributed entities. Such simulations form a dynamic communication graph continuously evolving and requires an incremental distribution algorithm that can adapt to their dynamic nature and can provide placement hints at any time.

In the near future, we will consider handling dynamic ant populations where the ant count adapt to the available power and load of associated processors. Later we plan to add more control to our system: indeed, this system is reactive in nature, and it would be desirable to add some processing before giving migration hint to the application (simulation). Therefore, we plan to add an heuristic layer between the colored ant system layer and the application layer that will control and smooth the results given by

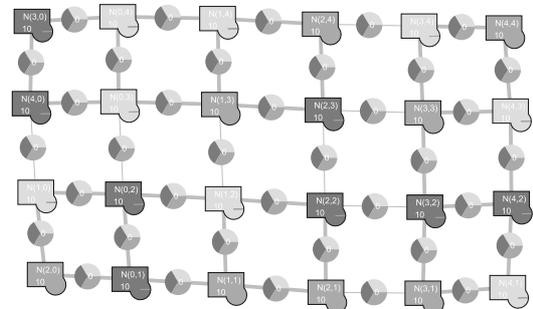


Figure 9: The initial grid.

the CAS. This layer would allow us to take into account constraints tied to the simulation that cannot be directly introduced in the ant system (e.g. non-migrable resources like databases).

## REFERENCES

- Bertelle, C., Dutot, A., Guinand, F., and Olivier, D. (2002a). Dimants: a distributed multi-castes ant system for dna sequencing by hybridization. In *NET-TABS 2002*, AAMAS 2002 Conf, Bologna (Italy).
- Bertelle, C., Dutot, A., Guinand, F., and Olivier, D. (2002b). Distribution of agent based simulation with colored ant algorithm. In *ESS2002*, pages 39–43, Dresden (Germany).
- Caro, G. D. and Dorigo, M. (1997). Antnet: A mobile agents approach to adaptive routing. Technical report, IRIDIA, Université libre de Bruxelles, Belgium.

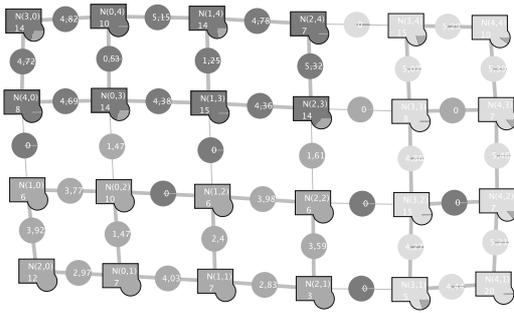


Figure 10: The algorithm on a grid after 100 steps.

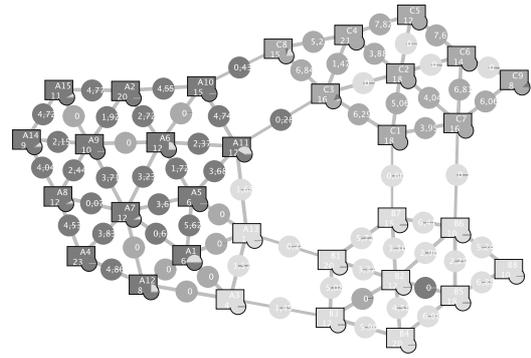


Figure 12: After 50 steps.

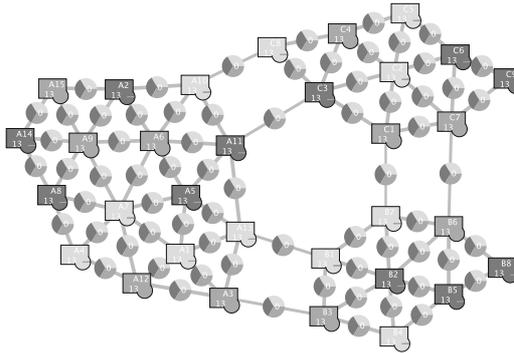


Figure 11: The original graph.

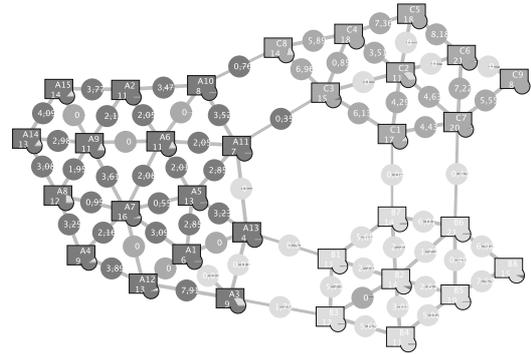


Figure 13: After 200 steps.

Dorigo, M. and Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.

Dorigo, M., Maniezzo, V., and Coloni, A. (1996). The ant system: optimization by a colony of cooperating agents. *IEEE Trans. Systems Man Cybernet.*, 26:29–41.

Faieta, B. and Lumer, E. (1994). Diversity and adaptation in populations of clustering ants. In *Conference on Simulation of Adaptive Behaviour*, Brighton.

Gordon, D. (1995). The expandable network of ant exploration. *Animal Behaviour*, 50:995–1007.

Kuntz, P., Layzell, P., and Snyers, D. (1997). A colony of ant-like agents for partitioning in vlsi technology. In *Fourth European Conference on Artificial Life*, pages 417–424, Cambridge, MA:MIT Press.

Langton, C., editor (1987). *Artificial Life*. Addison Wesley.

White, T. (1997). Routing with swarm intelligence. Technical Report SCE-97-15.

## AUTHOR BIOGRAPHIES

**Antoine Dutot** is PhD student in computer sciences research laboratory of Le Havre university (LIH). He works on swarm intelligence models for dynamic repartition of distributed simulations.

**Cyrille Bertelle** is associate professor in computer sciences research laboratory of Le Havre university (LIH). He works on natural complex systems modelling.

**Damien Olivier** is associate professor in computer sciences research laboratory of Le Havre university (LIH). He works on natural complex systems modelling.

**Frédéric Guinand** is associate professor in computer sciences research laboratory of Le Havre university (LIH). He works on scheduling for the parallel and distributed applications and on bioinformatic and genomic.

# A SIMULATOR MODULE FOR ADVANCED EQUATION ASSEMBLING

Stephan Wagner<sup>1</sup>, Tibor Grasser<sup>1</sup>, Claus Fischer\*, and Siegfried Selberherr<sup>2</sup>

<sup>1</sup>Christian-Doppler-Laboratory for TCAD in Microelectronics  
at the Institute for Microelectronics

<sup>2</sup>Institute for Microelectronics  
Technical University Vienna  
Gusshausstr. 27–29, A-1040 Vienna, Austria  
E-mail: Wagner@iue.tuwien.ac.at

\*Firma Dr. Claus Fischer  
Gustav Fuhrichweg 24/1, A-2201 Gerasdorf bei Wien, Austria

## KEYWORDS

Finite Boxes, Linear Equation Systems, Contact Current, Boundary Conditions, Interface Conditions

## ABSTRACT

We present a generally applicable simulator module which provides an advanced equation assembly system. The module has been originally developed for the simulation of semiconductor devices based on the Finite Boxes discretization scheme and is currently used in the general purpose device and circuit simulator MINIMOS-NT. In general, such simulations require the solution of a specific set of nonlinear partial differential equations which are discretized on a grid. Since the resulting nonlinear problem is solved by a damped Newton algorithm the solution of a linear equation system has to be obtained at each step. The module is responsible for assembling these systems and takes several requirements of the simulation process, namely the representation of boundary conditions, physically motivated variable transformation, preelimination and numerical conditioning, into account.

## INTRODUCTION

The Finite Boxes discretization method is employed in various kinds of numerical tools and simulators for fast and accurate solving of nonlinear partial differential equation (PDE) systems. The discretized problem is then usually solved by damped Newton iterations which require the solution of a linear equation system at each step. The extensibility and effectiveness of any simulator highly depends on the capabilities of its core modules responsible for handling the linear equation systems. We present an advanced equation assembly module successfully coupled to MINIMOS-NT.

MINIMOS-NT is a general purpose device and circuit simulator that has been developed at the Institute for Microelectronics for twelve years. Besides the basic semiconductor equations (Selberherr 1984), several different types of transport equations can be solved. Among these are the hydrodynamic equations which capture hot-carrier transport (Stratton 1962, Bløtekjær 1970), the lattice heat flow equation to cover thermal effects like self-heating (Wachutka 1990), and the circuit equations to connect single devices to a circuit (Grasser and Selberherr 2001), both electrically and thermally. Furthermore, various interface and boundary conditions are taken care of, which include Ohmic and Schottky contacts, thermionic field emission over and tunneling through various kinds of barriers. This demands a sophisticated system handling the equation assembly, in order to keep the simulator design flexible. To implement such a system, the requirements will be identified and generalized.

A crucial aspect is also the requirement of assembling and solving complex-valued linear equation systems. For that reason the module is able to handle both real-valued and complex-valued contributions and systems.

## THE ANALYTICAL PROBLEM

In order to analyze the electronic properties of an arbitrary semiconductor structure under all kinds of operating conditions, the effects related to the transport of charge carriers under the influence of external fields must be modeled. In MINIMOS-NT carrier transport can be treated by the drift-diffusion and the hydrodynamic transport models.

Both models are based on the semiclassical Boltzmann transport equation which is a time-dependent partial integro-differential equation in the six-dimensional phase space. By

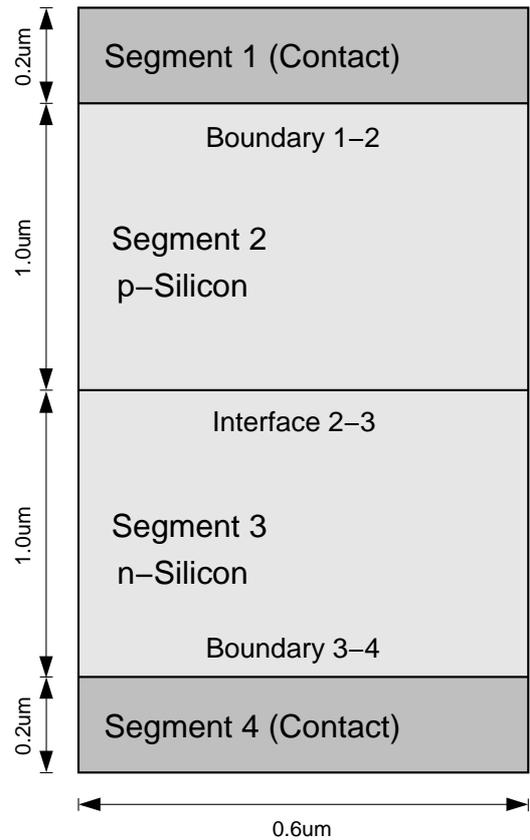
the so-called method of moments this equation can be transformed in an infinite series of equations. Keeping only the zero and first order moment equations (with proper closure assumptions) yields the basic semiconductor equations. Considering two additional moments gives the hydrodynamic model (Grasser et al. 2003).

The basic semiconductor equations, as given by VanRoosbroeck (VanRoosbroeck 1950), consist of the Poisson equation, the continuity equations for electron and holes as well as the current relations for both carrier types. The unknown quantities of this equation system are the electrostatic potential,  $\psi$ , and the electron and hole concentrations,  $n$  and  $p$ , respectively.  $C$  denotes the net concentration of the ionized dopants,  $\varepsilon$  is the dielectric permittivity of the semiconductor, and  $R$  is the net recombination rate. The heat-flow equation and thus the lattice temperature  $T_L$  is added to account for thermal effects. In the hydrodynamic case the carrier temperatures are allowed to be different from the lattice temperature, adding two more quantities, which are the electron and hole temperatures  $T_n$  and  $T_p$ .

Basically, a device structure can be divided into several segments to enable simulation of advanced heterostructures and to properly account for all conditions (which may cause very abrupt changes) at the contacts and interfaces between these segments, respectively. See Fig. 1 for an illustration of this concept. Every segment represents an independent domain  $D$  in one, two, or three dimensions where the PDEs are posed. The equations are implicitly formulated for a quantity  $x$  as  $f_{(x)} = 0$  and termed control functions. In order to fully define the mathematical problem, suitable boundary conditions for contacts, interfaces, and external surfaces have to be applied.

Generally, such a system cannot be solved analytically, and the solution must be calculated by means of numerical methods. This approach normally consists of three tasks:

1. The domain  $D$  is partitioned into a finite number of subdomains  $D_i$ , in which the solution can be approximated with a desired accuracy.
2. The PDE system is approximated in each of the subdomains by algebraic equations. The unknown functions are approximated by functions with a given structure. Hence, the unknowns of the algebraic equations are approximations of the continuous solutions at discrete grid points in the domain. Thus, generally a large system of nonlinear, algebraic equations is obtained with unknowns comprised of approximations of the unknown functions at discrete points.
3. The third task is to derive a solution of the unknowns of the nonlinear algebraic system. In the best case an exact



**Figure 1:** Illustration of the segment concept: a simple diode

solution of this system can be obtained, which represents a good approximation of the solution of the analytically formulated problem (which cannot be solved exactly). The quality of the approximation depends on the fineness of the partitioning into subdomains as well as on the suitability of the approximating functions.

For the derivation of the discrete problem several methods can be applied. We deal here with point residual methods: the finite difference method based on rectangular grids or the finite boxes (box integration) method allowing general unstructured grids. In the case of orthogonal rectangular grids both methods yield the same discretization.

## DISCRETIZATION

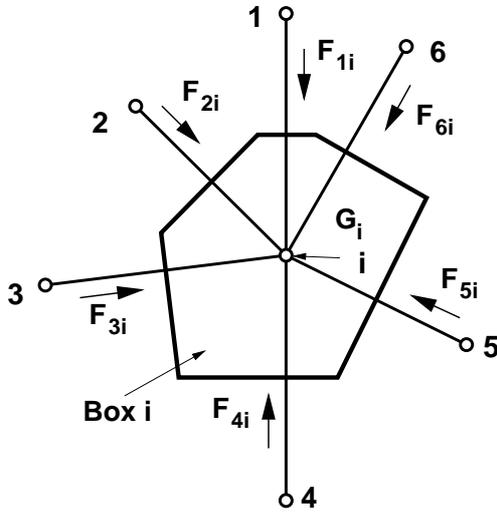
Nonlinear partial differential equations of second order can appear in three variants: elliptic, parabolic, and hyperbolic PDEs. The Poisson equation as well as the steady-state continuity equations form a system of elliptic PDEs, whereas the heat-flow equation is parabolic. To completely determine the solution of an elliptic PDE boundary conditions have to be specified. Since parabolic and hyperbolic PDEs describe

evolutionary processes, time normally is an independent variable and an initial condition is additionally required. Hence, also the transient continuity equations are parabolic.

Applying the finite boxes discretization scheme (Selberherr 1984) the equations are integrated over a control volume (subdomain, usually obtained by a Voronoi tessellation)  $D_i$  which is associated with the grid point  $P_i$ . For this grid point a general equation for the quantity  $x$  is implicitly given as

$$f_{x_i}^S = \sum_j F_{x_{i,j}} + G_i = 0 \quad (1)$$

where  $j$  runs over all neighboring grid points in the same segment,  $F_{x_{i,j}}$  is the flux between points  $i$  and  $j$ , and  $G_i$  is the source term (see Fig. 2).



**Figure 2:** Box  $i$  with 6 neighbors

Grid points on the boundary  $\partial D$  are defined as having neighbor grid points in other segments. Thus, (1) does not represent the complete control function  $f_x$ , since all contributions of fluxes into the contact or the other segment are missing. For that reason, the information for these boxes has to be completed by taking the boundary conditions into account. Common boundary conditions are the Dirichlet condition, which specifies the solution on the boundary  $\partial D$ , the Neumann condition, which specifies the normal derivative, and the linear combination of these conditions giving an intermediate type:

$$\mathbf{n} \cdot \text{grad}x + \sigma x = \delta \quad (2)$$

Generally, the form of these conditions depends on the respective boundary models, and the conditions of which depend on the interior information. For that reason, the equation assembly is often performed in a coupled way, causing complicated modules. For instance, it is absolutely necessary

to differ between interior and boundary points. Considering a general tetrahedron, there exist many kinds of boundary points (depending on the number of edges involved), which have to be treated separately. This leads to a complicated implementation of the models and can make simplifications necessary. Thus, due to organizational and implementational issues this form of coupling should be avoided.

More complex models with exponential interdependency between the solution variables such as thermionic field emission interface conditions (Schroeder 1994, Simlinger 1996) have also been implemented.

A method has been under development to implement segment models calculating the interior fluxes and their derivatives independently from the boundary models. The segment models do not have to differentiate the point type, they do not even have to care about the boundary model used. The assembly system is responsible for combining all relevant contributions by using the information given by the boundary models.

### Interface Conditions

To account for complex interface conditions, grid points located at the boundary of the segments (see Fig. 3a) have three values, one for each segment (see Fig. 3b) and a third point located directly at the interface which can be used to formulate more complicated interface conditions like e.g., interface charges. However, to simplify notation these interface values will be omitted in our discussion and only the two interface points,  $i$  and  $i'$ , are used.

Basically, the two (incomplete) equations  $f_{x_i}^S$  and  $f_{x_{i'}}^S$  are completed by adding the missing boundary fluxes  $F_{x_{i,i'}}$ :

$$f_{x_i} = f_{x_i}^S + F_{x_{i,i'}} = 0 \quad (3)$$

$$f_{x_{i'}} = f_{x_{i'}}^S - F_{x_{i,i'}} = 0 \quad (4)$$

The intermediate type of interfaces (2) and thus also the two other types of interfaces are generally given in linearized form by:

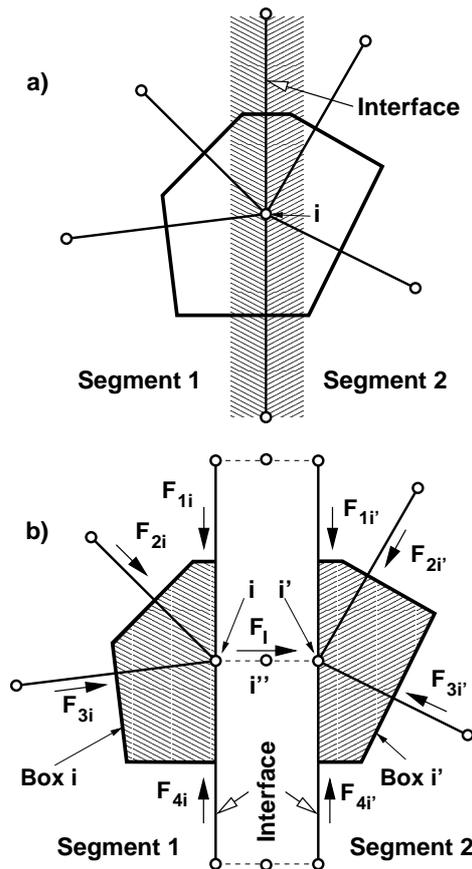
$$\alpha(x_i - \beta x_{i'} + \gamma) = F_{x_{i,i'}} \quad (5)$$

$\alpha$ ,  $\beta$ , and  $\gamma$  are linearized coefficients,  $F_{x_{i,i'}}$  represents the flux over the interface. The three types of interfaces differ in the magnitude of  $\alpha$ .

In the case of an arbitrary splitting of a homogeneous region into different segments, the boundary models have to ensure that the simulation results remain unchanged. By adding (4) to (3), the box of grid point  $P_i$  can be completed and the boundary flux is eliminated. The merged box is now valid for both grid points, for that reason the respective equation can not only be used for grid point  $P_i$ , but also for  $P_{i'}$ .

Whereas the segment models assemble the so-called segment matrix, the interface models are responsible for assembling and configuring the interface system consisting of a boundary and special-purpose transformation matrix. New equations based on (5) can be introduced into the boundary matrix without any limitations on  $\alpha$ , thus from 0 (Neumann) to  $\infty$  (Dirichlet). The interface models are also responsible for configuring the transformation matrix to combine the segment and boundary matrix correctly. Depending on the interface type there are two possibilities:

- Dirichlet boundaries are characterized by  $\alpha \rightarrow \infty$ . Thus, the implicit equation  $x_i = \beta x_{i'} - \gamma$  can be used as a substitute equation. As these equations are normally not diagonally dominant they have a negative impact on the condition number and are configured to be preeliminated (see Section ).
- For the other types (explicit boundary conditions) the boundary flux is simply added to the segment fluxes. In



**Figure 3:** Splitting of interface points: Interface points as given in a) are split into three different points having the same geometrical coordinates b)

the case of a large  $\alpha$ , the transformation matrix could be used to scale the entries by  $1/\alpha$  because of the preconditioner used in the solver module.

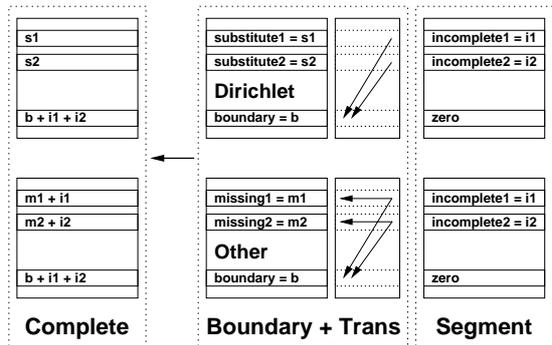
Note, that all interface-dependent information is administrated by the respective interface model only.

As an additional feature, the transformation matrix can be used to calculate several independent boundary quantities by combining the specific boundary value with the segment entries (also in the case of Dirichlet boundaries). For example, the dielectric flux over the interface is calculated as  $\sum_i f_{x_i}^S$  and introduced as a solution variable because some interface models require the cross-interface electric field strength to determine tunnel processes. Calculation of the normal electric field is thus trivial. Note that this is not the case when the normal component of the electric field  $\vec{E}_n$  has to be calculated using neighboring points when unstructured two- or three-dimensional grids are used.

See Fig. 4 for an illustration of these concepts. The transformations are set up to combine the various segment contributions with the boundary system.

### Boundary Conditions

Contacts are handled in a similar way to interfaces. However, in the contact segment there is only one variable available for each solution quantity ( $x_C$ ). Note that contacts are represented by spacial multi-dimensional segments. Furthermore, all fluxes over the boundary are handled as additional solution variables  $F_C$  (e.g., contact charge  $Q_C$  for Poisson equation, contact electron current  $I_{nC}$  for the electron continuity equations, or  $H_C$  as the contact heat flow).



**Figure 4:** The complete equations are a combination of the boundary and the segment system. This combination is controlled by the transformation matrix and depends on the interface type.

For explicit boundary conditions one gets

$$f_{x_i} = f_{x_i}^S + F_{x_i,C} = 0 \quad (6)$$

$$f_{F_C} = F_C + \sum_i f_{x_i}^S = 0 \quad (7)$$

with  $i$  running over all segment grid points. At Schottky contacts explicit boundary conditions apply. The semiconductor contact potential  $\psi_s$  is fixed and given as the difference of the metal quasi-Fermi level (which is specified by the contact voltage  $\psi_C$ ) and the metal workfunction difference potential  $\psi_{wf}$ .

$$\psi_s = \psi_C - \psi_{wf}, \quad \text{where} \quad \psi_{wf} = -\frac{E_w}{q} \quad (8)$$

The difference between the conduction band energy  $E_C$  and the metal workfunction energy gives the workfunction difference energy  $E_w$  which is the barrier height of the Schottky contact.

For Dirichlet boundary conditions one gets

$$f_{x_i} = x_C - h(x_i) = 0 \quad (9)$$

$$f_{F_C} = F_C + \sum_i f_{x_i}^S = 0 \quad (10)$$

Here,  $x_C$  is the boundary value of the quantity, which is a solution variable, whereas (10) is used as constitutive relation for the actual flow over the boundary  $F_C$ . For example, at Ohmic contacts simple Dirichlet boundary conditions apply. The contact potential  $\psi_s$ , the carrier contact concentrations  $n_s$  and  $p_s$ , and in the hydrodynamic simulation case, the contact carrier temperatures  $T_n$  and  $T_p$  are fixed. The metal quasi-Fermi level (which is specified by the contact voltage  $\psi_C$ ) is equal to the semiconductor quasi-Fermi level. With the constant built-in potential  $\psi_{bi}$  (calculated after (Fischer 1994)), the contact potential at the semiconductor boundary reads

$$\psi_s = \psi_C + \psi_{bi}. \quad (11)$$

For Neumann boundaries the flux over the boundary is zero hence the equation assembled by the segment model is already complete.

### Separate Contact Variables

Having a separate solution variable for the contact voltage avoids numerical problems with large arguments of the Bernoulli function  $B$ . Using a Scharfetter-Gummel discretization scheme (Scharfetter and Gummel 1969) the expression for the current between two grid points  $i$  and  $j$  reads

$$I_{i,j} = C_1(B(\Delta)n_j - B(-\Delta)n_i) x \quad (12)$$

$$\text{with} \quad \Delta = C_2(\psi_j - \psi_i) + C_3 \quad (13)$$

with  $C_i$  being material parameters. Applying the contact voltage directly to the boundary grid point could cause large arguments of  $B$  and hence numerical problems. This is avoided by having a separate variable for the contact voltage. At the beginning of the iteration procedure the constitutive relation for  $\psi_C$  is violated and will only successively be adapted which guarantees numerical stability. The generalized boundary condition is the constitutive relation for the contact potential  $\psi_C$  and reads:

$$f_{\psi_C} = \alpha\psi_C + \beta I_C + \gamma Q_C - \delta = 0 \quad (14)$$

where  $Q_C$  is the contact charge and  $I_C = I_{nC} + I_{pC} + \frac{\partial Q_C}{\partial t}$  the contact current. It should be noted that all these quantities are solution variables which are directly available.

### SOLVING OF THE NONLINEAR SYSTEM

MINIMOS-NT organizes the solving of the nonlinear, but discretized control functions  $\mathbf{f} = \mathbf{0}$  using a damped Newton algorithm ( $k$  is the number of the iteration step) (Selberherr 1984):

$$\mathbf{J}^k \cdot \mathbf{x}^{k+1} = \mathbf{f}(\mathbf{v}^k) \quad (15)$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + F_d \mathbf{x}^{k+1} \quad (16)$$

$$\mathbf{J} = -\frac{\partial \mathbf{f}}{\partial \mathbf{v}} \quad (17)$$

where  $\mathbf{J}$  is the Jacobian matrix,  $\mathbf{f}(\mathbf{v})$  the residual and  $\mathbf{x}$  the update or correction vector (solution vector of the linear system) that is then used to calculate the next solution vector  $\mathbf{v}$  of the Newton approximation.

To avoid overshoot of the solution several damping schemes suggested by Deuffhard (Deuffhard 1974) or Bank and Rose (Bank and Rose 1981) are providing a damping factor  $F_d$ . For each Newton iteration step a linear equation system  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$  has to be assembled and solved.

### THE ASSEMBLY MODULE

MINIMOS-NT consists of two separate modules responsible for assembling and solving linear equation systems:

1. the assembly module which is directly accessed by the implemented physical models of the simulator, provides an effective application programming interface, various transformation algorithms and the preelimination system. In addition, sorting and scaling plug-ins can be called.
2. the solver module which is plugged into the assembly module, is responsible for solving the so-called inner linear equation system. The module currently used

provides a direct (Gaussian) method and two iterative solver schemes.

The key demands on the assembly module (class) can be summarized as follows:

1. Application Programming Interface providing methods for
  - adding contributions to the segment system
  - adding contributions to the boundary system
  - adding contributions to the transformation matrix
  - deleting equations
  - setting elimination flags
  - administration of priority information
2. Row transformation: linear combination of rows to extinguish large entries (see Section ).
3. Variable transformation: reduce the coupling of the semiconductor equations (see Section ).
4. Preelimination: eliminate problematic equations by Gaussian elimination to improve the condition of the inner system matrix (see Section ).
5. Call of specific plug-ins (see Section ) for
  - **Scaling:** Since a threshold value (tolerance) is used to decide whether to keep or skip an entry, the preconditioner used (Incomplete-LU factorization) requires a system matrix having entries of the same order of magnitude.
  - **Sorting:** Reduction of the bandwidth of a matrix to reduce the fill-in.
  - **Solving:** Calculate the solution vector of the linear equation system.

The input of the assembly module are the contributions of the various segment and boundary models implemented in the simulator. The assembly module compiles these values to a linear equation system, which is subsequently transformed in order to improve the condition of the system matrix.

For some simulations, for example derivation of the complex-valued admittance matrix, several linear equation systems differ only in the right-hand-side vector. Thus, the effort for assembling, compiling, preeliminating, sorting, scaling and factorizing of the system matrix actually has to be done only once - and this factored matrix could then be used for all RHS-vectors. For that reason the module is able to simultaneously assemble several RHS vectors.

A plug-in concept has been implemented for scaling, sorting and solving the inner linear equation system, making it possible to adapt or replace these modules easily. The sorting and scaling modules get the system matrix on input and return the sorting and scaling (diagonal) matrices which are then applied by the assembly module. The solver module gets the system matrix and all RHS vectors on input and returns the solution vectors of all inner linear equation systems on output.

After reverting all transformations and backsubstituting the preeliminated equations, the output of the assembly module are the complete solution vector (or vectors in case of multiple right-hand-side vectors). In addition, the right-hand-side vector(s) are returned which can be used for various norm calculations.

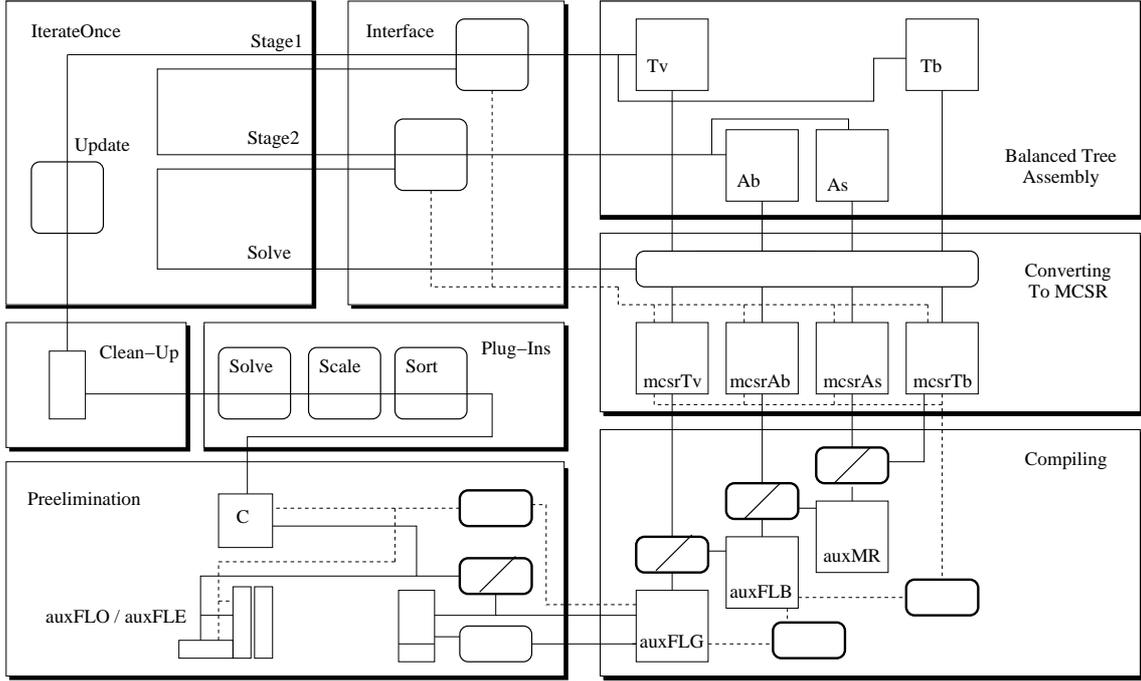
A schematic overview of the complete concept is given in Fig. 5. In the upper left corner the Newton iteration control function `IterateOnce` is represented, which is divided into two stages and uses an interface class to access the assembly module. Following the solid lines beginning at the interface, the four matrices  $\mathbf{T}_b$ ,  $\mathbf{A}_s$ ,  $\mathbf{A}_b$ , and  $\mathbf{T}_v$  (see Section ) are assembled by using a specific storage class. All diagonal elements are stored in one array, and the off-diagonals, the positions of which are not known in advance, in a balanced binary tree for each row, sorted by column. This allows flexible adding of all entries of the sparse matrices.

These structures are then converted to the *Modified Sparse Compressed Row* (MCSR) format (Saad 1990) and are compiled resulting in the complete linear system (`auxFLG`), which is preeliminated to get the inner and the outer linear equation system. The inner one (represented by  $\mathbf{C}$ ) is passed to the sorting and scaling plug-ins and finally solved by the solver module. After the solution has been calculated, scaling and sorting have to be reversed and the preeliminated equations are solved back.

The Newton adjustment levels (dashed lines) reuse already existing MCSR structures, which reduces the assembling effort: the balanced trees may be skipped completely, and during compiling and preelimination much simpler functions (bold boxes) can be used than in the conventional assembly mode (bold boxes with slash).

### Assembly of the Complete Linear Equation System

The semiconductor device is divided into several segments that are geometrical regions employing a distinct set of models. The implementation of each model is completely independent from other models and each model is basically allowed to enter its contributions to the linear equation system. All boundary and interface issues are completely separated



**Figure 5:** Schematic assembly overview

from the general segment models, which is represented by assembly structures for the boundary system which are independent from the segment ones.

Thus, the system matrix  $\mathbf{A}$  (the Jacobian matrix in Newton approximation) will be assembled from two parts, namely the direct part  $\mathbf{A}_b$  (boundary models) and the transformed part  $\mathbf{A}_s$  (segment models). The latter is multiplied by the row transformation matrix  $\mathbf{T}_b$  from the left before contributing to the system matrix  $\mathbf{A}$ . The right hand side vector  $\mathbf{b}$  is treated the same way:

$$\mathbf{A} = \mathbf{A}_b + \mathbf{T}_b \cdot \mathbf{A}_s \quad (18)$$

$$\mathbf{b} = \mathbf{b}_b + \mathbf{T}_b \cdot \mathbf{b}_s \quad (19)$$

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (20)$$

Although in principle every model is allowed to add entries to all components, the assembly module checks two prerequisites before actually entering the value: first, the quantity the value belongs to is marked to be solved (the user may request only a subset of all provided models) and secondly the priority of the model is high enough to modify the row transformation properties. As stated before, the row transformation is used to complete missing fluxes in boundary boxes. Since a grid point can be part of more than two segments, a ranking using a priority has been introduced. For example, contact models have usually the highest priority and thus their contributions are always used for completion. All three

matrices  $\mathbf{A}_b$ ,  $\mathbf{A}_s$ , and  $\mathbf{T}_b$  and the two vectors  $\mathbf{b}_b$  and  $\mathbf{b}_s$  may be assembled simultaneously, so no assembly sequence must be adhered to. In addition, a fourth matrix  $\mathbf{T}_v$  is assembled which contains information for an additional variable transformation (see Section ).

During the assembling process, all contributions are added to values stored in balanced binary trees. After the assembly is finished, these trees are converted to the sparse matrix format MCSR since all necessary mathematical operations are defined for these structures. The analogous, column oriented MCSC format is used to speed up column deleting required by the transformation matrix.

During the Newton iterations the structural configuration of these matrices is not modified very often (e.g., on enabling more derivatives), thus, the tree assembly may be skipped and the variables may be entered directly in the already existing MCSR structures. Hence, the effort for deleting, tree assembling, reallocating and converting can be saved which drastically speeds up the assembly process. However, if an entry in the structure is missing, the conventional assembly procedure can be easily restarted. The so-called Newton adjustment addresses not only the assembly matrices, but also the resulting structures of the compilation and preelimination process.

## Row Transformation

The complete linear equation system is built from an original system (segment system), which is the main matrix  $\mathbf{A}_s$  and the main right hand side vector  $\mathbf{b}_s$ , both of them representing cumulated fluxes and their derivatives to the system variables. Basically, the fluxes are calculated from segment models which are the models for the interior of discretized regions. The matrix is a linear superposition of very small matrices, one for each flux, with a few non-zero elements only. Consequently, the same superposition applies for the vector  $\mathbf{b}_s$ . All fluxes are assigned to boxes, a box is in turn assigned to each variable.

As the control function for a box is defined by the user, for example being the sum of all fluxes leaving the box, the fluxes leaving the boxes are entered into the vector  $\mathbf{b}_s$  in the places appropriate for the variables that are assigned to the boxes. In context of the Newton method, matrix  $\mathbf{A}_s$  contains the negative derivatives of the values in  $\mathbf{b}_s$  to the system variables, so that the change  $dx$  in the variables leads to a change  $-\mathbf{A}_s \cdot d\mathbf{v} = d\mathbf{b}_s$  in the right hand side. Considering  $\mathbf{b}_s$  a function of the variable vector  $\mathbf{v}$ , one can write:

$$\mathbf{b}_s = \mathbf{b}_s(\mathbf{v}) \quad (21)$$

$$\mathbf{A}_s = -\frac{d\mathbf{b}_s}{d\mathbf{v}} \quad (22)$$

The boundary conditions will enforce some special physical conditions at the boundaries. The control functions of boxes along the boundary will usually be completed by the boundary conditions. For example, a Dirichlet boundary condition will use the dielectric flux cumulated in the boundary box to calculate the surface charge on the surface of the adjacent material. The equation used to calculate the value of the boundary variable, however, will not always make use of the fluxes accumulated in the main system.

The boundary conditions are therefore implemented by two elements: a boundary system ( $\mathbf{A}_b$  and  $\mathbf{b}_b$ ) and a transformation matrix  $\mathbf{T}_b$ . The purpose of the matrix  $\mathbf{T}_b$  is the forwarding of the fluxes of the main system to their final destination or their resetting if they are not required. The system of  $\mathbf{A}_b$  and  $\mathbf{b}_b$  represents additional or substitutional parts of the final equation for the variables at the boundaries. Again, the entries in the matrix  $\mathbf{A}_s$  are the negative derivatives of the right hand side vector  $\mathbf{b}_b$  to the variable vector  $\mathbf{v}$ :

$$\mathbf{b}_b = \mathbf{b}_b(\mathbf{v}) \quad (23)$$

$$\mathbf{A}_b = -\frac{d\mathbf{b}_b}{d\mathbf{v}} \quad (24)$$

## The Complete Linear System

The full system is assembled from the segment system and the boundary system in the following way:

$$\mathbf{b} = \mathbf{b}_b + \mathbf{T}_b \cdot \mathbf{b}_s \quad (25)$$

$$\mathbf{A} = \mathbf{A}_b + \mathbf{T}_b \cdot \mathbf{A}_s \quad (26)$$

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (27)$$

$$(\mathbf{A}_b + \mathbf{T}_b \cdot \mathbf{A}_s) \cdot \mathbf{x} = \mathbf{b}_b + \mathbf{T}_b \cdot \mathbf{b}_s \quad (28)$$

As stated above, vector  $\mathbf{x}$  represents a linear change of the variables vector  $\mathbf{v}$ . With equation (29) and the new variables vector  $\mathbf{v}_n$  from equation (30), the new value  $\mathbf{b}_n$  of the vector function  $\mathbf{b}(\mathbf{v})$  will be described by the linear approximation in equation (31).

$$\mathbf{x} = d\mathbf{v} \quad (29)$$

$$\mathbf{v}_n = \mathbf{v} + \mathbf{x} \quad (30)$$

$$\mathbf{b}_n(\mathbf{v}_n) = \mathbf{b} + \frac{d\mathbf{b}}{d\mathbf{v}} \cdot d\mathbf{v} = \mathbf{b} - \mathbf{A} \cdot \mathbf{x} = 0 \quad (31)$$

## Variable Transformation

Especially in the case of mixed quantities in the solution vector, a variable transformation is sometimes helpful to improve the condition of the linear system. The representation chosen here allows to specify fairly arbitrary variable transformations to be applied to the system. Basically, a matrix  $\mathbf{T}_v$  is assembled and multiplied with the system matrix.

For example, to reduce the coupling of the semiconductor equations and thus improve the condition of the system matrix, a transformation of the stationary drift-diffusion model is suggested in (Ascher et al. 1986).

The transformation expressed by matrix  $\mathbf{T}_v$  is given by equation (32):

$$((\mathbf{A}_b + \mathbf{T}_b \cdot \mathbf{A}_s) \cdot \mathbf{T}_v) \cdot (\mathbf{T}_v^{-1} \cdot \mathbf{x}) = (\mathbf{b}_b + \mathbf{T}_b \cdot \mathbf{b}_s) \quad (32)$$

For compactness the following substitutions will be used hereinafter:

$$\tilde{\mathbf{A}} = ((\mathbf{A}_b + \mathbf{T}_b \cdot \mathbf{A}_s) \cdot \mathbf{T}_v) \quad (33)$$

$$\tilde{\mathbf{x}} = (\mathbf{T}_v^{-1} \cdot \mathbf{x}) \quad (34)$$

$$\tilde{\mathbf{b}} = (\mathbf{b}_b + \mathbf{T}_b \cdot \mathbf{b}_s) \quad (35)$$

## Prelimination

The main matrix  $\mathbf{A}_s$  consists of fluxes that will (if the control functions are correctly assigned to the variables) satisfy

the criterion of diagonal-dominance that is necessary to make the linear equation system solvable with an iterative solver. The transformations and additional terms imposed by the boundary conditions may heavily disrupt this feature both in structural and numerical aspects. Some of the boundary or interface conditions can make the full system matrix so ill-conditioned that this simply prevents iterative linear solvers from converging.

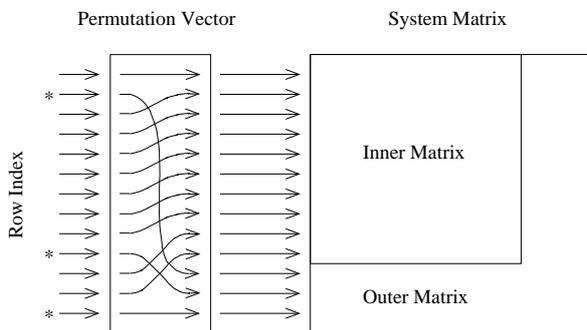
One solution to this problem which occurs only at the boundary variables that are affected in this way by the boundary conditions, is to apply Gaussian elimination to these variables/equations before the system is passed on to the linear solver. After the iterative solver has converged, the eliminated variables are calculated by backsubstitution into the eliminated equations.

Before they can be eliminated, the equations of this type are sorted to the back of the matrix, together with their assigned variables. This is done by applying a permutation matrix  $\mathbf{P}$  to the linear equation system. The permutation matrix is calculated automatically on solving the system. The equation causing a possibly ill condition have to be marked for pre-elimination. The outer system is removed from the linear equation system and later solved by Gaussian elimination, the inner system is passed on to an iterative solver. See Fig. 6 (Fischer 1994) for an illustration of this concept.

The resulting system is given by (36):

$$(\mathbf{E} \cdot \mathbf{P} \cdot \tilde{\mathbf{A}} \cdot \mathbf{P}^T) \cdot (\mathbf{P} \cdot \tilde{\mathbf{x}}) = (\mathbf{E} \cdot \mathbf{P} \cdot \tilde{\mathbf{b}}) \quad (36)$$

Here,  $\mathbf{P}$  is the permutation matrix with its inverse equal to its transposed matrix  $\mathbf{P}^T$ .  $\mathbf{E}$  is a matrix of elimination coefficients obtained as the lower matrix  $\mathbf{L}$  of a Gaussian elimination of the permuted system matrix.  $\mathbf{E}$  contains non-zero off-diagonals in the outer parts only, the inner matrix up to



**Figure 6:** All equations marked for preelimination (\*) are moved to the outer system matrix, the others remain in the inner one.

the row/column index that narrows the section passed to the linear solver is a strict unity matrix.

### Call of Specific Plug-Ins

Matrices arising from the discretization of differential operators are sparse, because only neighbor points are considered. To reduce memory consumption, only the non-zero elements are stored (see MCSR format). During a factorization of  $\mathbf{A}$  into an upper and lower triangular matrix  $\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$ , additional matrix elements termed fill-in (Selberherr 1984) become non-zero and require additional memory. In order to minimize the fill-in, the system matrix is usually sorted. The standard module provided by default obtains the sorting matrix  $\mathbf{R}_s$  (similar to  $\mathbf{P}$ ) by a Cuthill-McKee-based algorithm.

To provide the (ILU-) preconditioner with a normalized representation of the matrix, a scaling of all values has to be performed. The standard algorithm used by default works with a two-stage strategy (Fischer and Selberherr 1994): In the first stage, the matrix is scaled such that the diagonal elements are one. The second stage attempts to suppress the off-diagonals while keeping the diagonals at unity. The resulting scaling matrices  $\mathbf{S}_r$  and  $\mathbf{S}_c$  are diagonal matrices, and  $\mathbf{R}_s^T$  equals  $\mathbf{R}_s^{-1}$ . With  $\mathbf{A}_i \cdot \mathbf{x}_i = \mathbf{b}_i$  as the inner system, the effect of sorting and scaling is given in (37):

$$(\mathbf{S}_r \cdot (\mathbf{R}_s^T \cdot \mathbf{A}_i \cdot \mathbf{R}_s) \cdot \mathbf{S}_c) \cdot (\mathbf{S}_c^{-1} \cdot (\mathbf{R}_s^T \cdot \mathbf{x}_i)) = \mathbf{S}_r \cdot (\mathbf{R}_s^T \cdot \mathbf{b}_i) \quad (37)$$

The assembly module is finally responsible for passing the inner linear equation system to the solver module. There are several approaches to obtain the solution vector: a Gaussian solver factorizes the matrix with a complete LU factorization, followed by a forward- and a backward substitution. Iterative solvers use successive approximations of this vector to obtain more accurate solutions to a linear equation system at each step (Barrett et al. 1994). In addition, the solver module provides a general interface to alternative solvers. After returning from the solver module, scaling and sorting are reverted and the preeliminated equations are backsubstituted.

### CONCLUSION

We presented the concept and implementation of an advanced assembly module successfully applied in the device and circuit simulator MINIMOS-NT. The generally applicable module provides all conceptual and numerical features required for assembling and solving linear systems arising from discretized PDEs. The presented concepts result in superior stability of MINIMOS-NT without restricting model implementation and further development. The general approach for treating boundary conditions yields in combination with several preconditioning measures diagonal-dominant linear equation systems well prepared for advanced

solver algorithms. As a result, boundary conditions for specific operating points can be directly applied without stepping to the desired value as is very common even in commercial simulators.

## AUTHOR BIOGRAPHIES

**STEPHAN WAGNER** was born in Vienna, Austria, in 1976. He studied electrical engineering at the Technical University Vienna, where he received the master degree of "Diplomingenieur" in 2001. As a member of the MINIMOS-NT development group he joined the Institute for Microelectronics in November 2001, where he is currently working on his doctoral degree. His scientific interests include device and circuit simulation, numerical aspects and software technology.

**TIBOR GRASSER** was born in Vienna, Austria, in 1970. He studied communications engineering at the Technische Universität Wien where he received the 'Diplomingenieur' and the doctoral degree in technical sciences in 1995 and 1999, respectively. He joined the Institut für Mikroelektronik in April 1996 where he is currently employed as an Assistant. In 2002 he received the *venia docendi* on Microelectronics. Since 1997 he has been heading the MINIMOS-NT development group, working on the successor of the highly successful MINIMOS program.

**CLAUS FISCHER** Claus Fischer was born in Vienna, Austria, in 1967. He received the doctoral degree in technical sciences from the 'Technische Universität Wien' in 1994. He started the Minimos NT project in order to extend the proven concepts and numerical methods of Minimos to complex geometrical and topological structures and introduced a generalized physical interface treatment. After Ph.D., he worked in the semiconductor industry for three years. He now runs his own software engineering company in Austria. His interests cover a range of software engineering topics for industrial use, from technical to business applications.

**SIEGFRIED SELBERHERR** was born in Klosterneuburg, Austria, in 1955. He received the degree of 'Diplomingenieur' in electrical engineering and the doctoral degree in technical sciences from the 'Technische Universität Wien' in 1978 and 1981, respectively. Dr. Selberherr has been holding the '*venia docendi*' on 'Computer-Aided Design' since 1984. Since 1988 he has been the head of the 'Institut für Mikroelektronik' and since 1999 he has been dean of the 'Fakultät für Elektrotechnik'. His current topics are modeling and simulation of problems for microelectronics engineering.

## REFERENCES

Ascher, U.; P.A. Markowich; C. Schmeiser; H. Steinrück; and R. Weiss. 1986. Conditioning of the Steady State Semiconduc-

tor Device Problem. Technical Report 86-18, University of British Columbia.

Bank, R.E. and D.J. Rose. 1981. "Global Approximate Newton Methods". *Numer.Math.*, 37:279–295.

Barrett, R.; M. Berry T.F. Chan; J. Demmel; J. Donato; J. Dongarra; V. Eijkhout; R. Pozo; C. Romine; and H. Van der Vorst. 1994. *Templates for the Solution of Linear Systems: Building Blocks of Iterative Methods*. SIAM, Philadelphia, PA.

Bløtebjerg, K. 1970. "Transport Equations for Electrons in Two-Valley Semiconductors". *IEEE Trans.Electron Devices*, ED-17(1):38–47.

Deuffhard, P. 1974. "A Modified Newton Method for the Solution of Ill-Conditioned Systems of Nonlinear Equations with Application to Multiple Shooting". *Numer.Math.*, 22:289–315.

Fischer, C. 1994. *Bauelementsimulation in einer computergestützten Entwurfsumgebung*. Dissertation, Technische Universität Wien. <http://www.iue.tuwien.ac.at>.

Fischer, C. and S. Selberherr. 1994. "Optimum Scaling of Non-Symmetric Jacobian Matrices for Threshold Pivoting Preconditioners". In *Intl. Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits NUPAD V*, pages 123–126, Honolulu.

Grasser, T.; H. Kosina; M. Gritsch; and S. Selberherr, 2001. "Using Six Moments of Boltzmann's Transport Equation for Device Simulation". *J.Appl.Phys.*, 90(5):2389–2396.

Grasser, T.; T.-w. Tang; H. Kosina; and S. Selberherr. 2003. "A Review of Hydrodynamic and Energy-Transport Models for Semiconductor Device Simulation". *Proc.IEEE*, 91(2):251–274.

Grasser, T. and S. Selberherr. 2001. "Fully-Coupled Electro-Thermal Mixed-Mode Device Simulation of SiGe HBT Circuits". *IEEE Trans.Electron Devices*, 48(7):1421–1427.

Saad, Y. 1990. *A Basic Tool Kit for Sparse Matrix Computations*. Technical Report, RIACS, NASA Ames Research Center, Moffett Field, CA 94035.

Scharfetter, D.L. and H.K. Gummel. 1969. "Large-Signal Analysis of a Silicon Read Diode Oscillator". *IEEE Trans.Electron Devices*, ED-16(1):64–77.

Schroeder, D. 1994. *Modelling of Interface Carrier Transport for Device Simulation*. Springer.

Selberherr, S. 1984. *Analysis and Simulation of Semiconductor Devices*. Springer, Wien–New York.

Simlinger, T. 1996. *Simulation von Heterostruktur-Feldeffekttransistoren*. Dissertation, Technische Universität Wien. <http://www.iue.tuwien.ac.at>.

Stratton, R. 1962. "Diffusion of Hot and Cold Electrons in Semiconductor Barriers". *Physical Review*, 126(6):2002–2014.

VanRoosbroeck, W.V. 1950. "Theory of Flow of Electrons and Holes in Germanium and Other Semiconductors". *Bell Syst.Techn.J.*, 29:560–607.

Wachutka, G.K. 1990. "Rigorous Thermodynamic Treatment of Heat Generation and Conduction in Semiconductor Device Modeling". *IEEE Trans.Computer-Aided Design*, 9(11):1141–1149.

# VERIFICATION OF REAL TIME UML SPECIFICATIONS THROUGH A SPECIALIZED INFERENCE MECHANISM BASED ON A TOKEN PLAYER ALGORITHM AND THE SEQUENT CALCULUS OF LINEAR LOGIC

Stéphane Julia, Michel dos Santos Soares  
Faculdade de Computação  
Universidade Federal de Uberlândia,  
P.O. Box 593, 38400-902, Uberlândia-M.G-Brazil,  
email: stephjl@aol.com, michelssoares@yahoo.com.br

**KEYWORDS**—UML, Petri Net, Linear Logic, Scheduling, Real Time System, Batch System

## ABSTRACT

The objective of this article is to present an approach based on UML dynamic diagrams, on time Petri Nets and Linear Logic for scenario verification of Real Time Systems. The main idea consists of translating the sequence diagrams which express the initial specifications of the system to a unique p-time Petri Net model which represents the global behaviour of the entire system. For the Petri Net fragments involved in conflict situations, symbolic production and consumption dates assigned to tokens are calculated using a non-conventional (max;+) algebra based on the sequent calculus of Linear Logic. These dates are then used to solve conflict situations within a token player algorithm used for scenario verification of Real Time specifications and which can be seen as a simulation tool for UML interaction diagrams. The approach is illustrated through an example of Real Time System used at the global coordination level of a Batch System.

## I. INTRODUCTION

The Object Oriented methods seem to be very suitable for proposing approaches to represent Real Time Systems, as real objects of the physical system are naturally transformed into software objects that can be easily implemented and verified. Of all Object Oriented notations, UML [OMG 1999] is one of the best accepted in the software industry. In particular, with the dynamic diagrams proposed by UML, it is possible to represent the communication mechanisms among several objects for a specific scenario. Therefore, UML notations have their limitations when they are used for specifying Real Time Systems. For example, it is not possible with a unique UML diagram to represent the set of all dynamic interactions that exist at a global system level. As a consequence, it becomes very difficult to guarantee that the execution of several sequence diagrams in parallel will not lead to time constraint violations.

Petri Nets [Murata 1989] are very well adapted to model Real Time Systems, as they allow for a good representation of conflict situations, shared resources, synchronous and asynchronous communica-

tion, precedence constraints and explicit time constraints in the time Petri Net case.

As was presented in [Cardoso 2001], translating sequence diagrams of UML in Petri Net models allows one to define an operational semantic for the sequence diagrams in order to know how these diagrams are executed in real time.

The dynamic behaviour of a system imposes a scheduling of control flow. The scheduling problem consists of organizing in time, the sequence of the operations considering time constraints (time intervals) and constraints of shared resources utilization necessary for operation execution. From the traditional point of view of Software Engineering, the scheduling problem is similar to the activity of scenario execution. A scenario execution becomes a kind of simulation which shows the system's behaviour in real time. In the real time system case, several scenarios can be executed simultaneously and conflict situations which have to be solved in real time (without a backtrack mechanism) can occur if a same non-preemptive resource is called at the same time for the execution of operations which belong to different scenarios.

In [Julia 2002], a simulation technic based on a token player algorithm was presented whose purpose was to verify real time UML specifications. The basic principle was to generate a class graph [Khansa 1996] each time a conflict situation for a shared resource was met in order to guarantee that a time constraint violation would not be reached. The class graph allows one to represent all possible evolutions of p-time Petri net fragments involved in conflict situations but has the disadvantage of the combinatory explosion. Another problem is that for each new scenario execution, new class graphs which are based on real numerical dates and durations must be calculated.

In this paper, an approach, based on the sequent calculus of Linear Logic [Girard 1987], will be proposed to solve conflict situations in order to accelerate conflict resolutions during the scenario execution. This will be realized calculating symbolic dates for the tokens in conflict by using a non conventional (max;+) algebra [Rivière 2001].

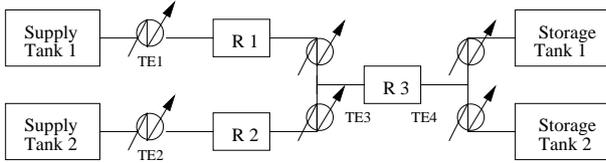


Fig. 1. Batch Production System

## II. REAL TIME SYSTEMS MODELLING BASED ON UML DIAGRAMS AND ON P-TIME PETRI NET

A Real Time System used at the global coordination level of the Batch System given in figure 1 will be considered to illustrate the specification activity of Real Time Systems using UML diagrams. A batch is a quantity of material which is transformed passing through different equipment and respecting a specific recipe which defines the sequence of operations. The production system of figure 1 executes two different recipes. Initially, two batches (1 and 2) are stored in their respective supply tanks (1 and 2). Recipe 1 consists on transferring batch 1 from supply tank 1 to reactor R1 to a processing stage utilizing thermal exchange TE1. After batch 1 is processed in R1, it is transferred to reactor R3 to another stage in processing, passing through TE3. When finalized, batch 1 is deposited in storage tank 1, passing through TE4 for the final product liberation. Recipe 2 behaves in a similar manner and consists of transferring batch 2 from supply tank 2 to storage tank 2, passing through reactors R2 and R3 and using thermal exchanges TE2, TE3 and TE4. The advantage of considering such a system is that some of the main features which generally appear in Real Time Systems will be considered. The execution of different recipes can be seen as a typical example of parallelism, and the utilization of common equipment (for example, reactor R3 will be used by both recipes) is an example of resource sharing.

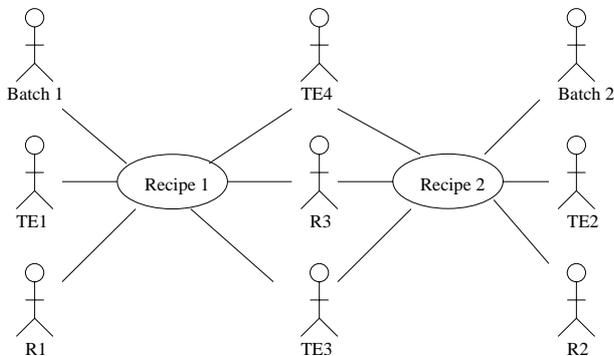


Fig. 2. Use Case Diagram

The proposed approach uses a Use Case diagram to show the main functions of the Real Time System from a user's point of view and the rela-

tionship between the system and the environment. Analysing Real Time Systems from an Object Oriented approach, the actors of the Use Case diagram are generally good candidates for objects. From the Use Case diagram of figure 2, it is possible to note that the execution of recipe 1 needs the physical equipment: TE1, TE3, TE4, R1 and R3. In the same manner, the execution of recipe 2 needs the physical equipment: TE2, TE3, TE4, R2 and R3. As actors are good candidates for objects, it is easy to conclude that there exists a recipe class which has a method that corresponds to the treatment of the corresponding batch, a reactor class which has a method that corresponds to the processing of a batch into a physical reactor, and a thermal exchange class which has a method called for transport operations. As a matter of fact, each actor of the Use Case diagram will have to communicate with its corresponding software object.

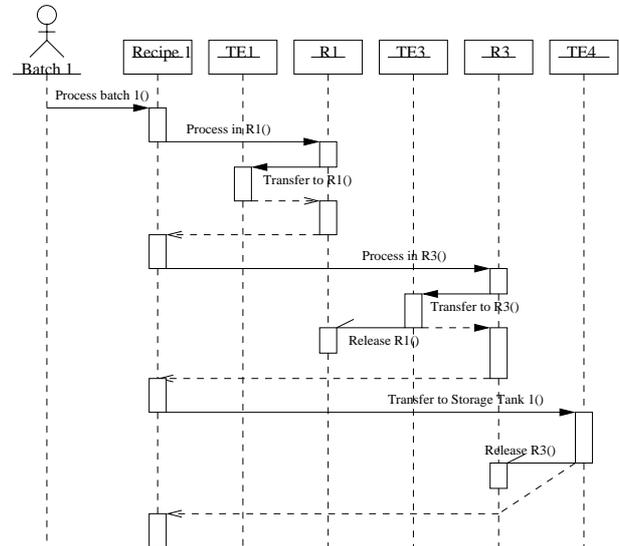


Fig. 3. Sequence Diagram for Recipe 1

After defining the main objects of the system, each function corresponding to a Use Case is given through a specific sequence of operations modelled by a sequence diagram that shows the communication mechanisms among the involved objects. This type of diagram explicitly shows the chronological order of operations. The scenario execution of recipe 1 and the interactions between the involved objects are shown in the sequence diagram of figure 3. Looking at this diagram, it is quite clear that initially, the actor Batch 1 calls the method of the software object which will execute recipe 1. After that, the object Recipe 1 calls the method of R1 for the processing operation. Then, object R1 calls the method of TE1 to transfer batch 1 from supply tank 1 to reactor R1 and, at the end of the first processing stage, a message is sent to the object Recipe 1 which can call the method of the object R3 to the second process-

ing stage. The object R3 calls the method of the object TE3 to transfer batch 1 from reactor R1 to reactor R3. Once batch 1 is transferred to R3, an asynchronous message is sent to R1 so that it becomes available and a synchronous response is sent to object R3 so that the processing in reactor R3 begins. At the end of the processing in reactor R3, the thermal exchange TE4 is requested to transfer batch 1 to storage tank 1 for product liberation. At the end of the transfer operation, an asynchronous message is sent to R3 so that it becomes available for other operations. The sequence diagram of recipe 2 is similar to the one of recipe 1. To build it, it is necessary to change the actor Batch 1 and the objects Recipe 1, TE1 and R1 from the sequence diagram shown in figure 3, by the actor Batch 2 and the objects Recipe 2, TE2 and R2, respectively.

One way of analysing specifications given through semi-formal UML notation is to formally define an operational semantic for the UML dynamic diagrams based on a formal notation which shows how these diagrams are executed in real time and which allows one to analyse qualitatively and quantitatively real time UML specifications. In particular, by analysing sequence diagrams separately, it is not possible to verify if a conflict situation can occur. For example, during real time execution, both sequence diagrams may have to request some common objects at the same time interval. Another limitation of the sequence diagrams in the real time system case is that explicit time constraints, like the initial date of a scenario execution, do not appear formally on these diagrams.

Petri Nets allows one to describe internal behaviour of objects as well as synchronous and asynchronous communications between objects (synchronous and asynchronous communications are represented by communication places (semaphore type)). Based on the interactions between objects specified in a sequence diagram, it is possible to obtain a Petri Net model which shows the interactions between the Petri Net objects (for each object, there is a corresponding Petri Net template which shows the internal behaviour of the object) involved in the execution of the scenario. Applying some of the reduction rules [Murata 1989] of the Petri Net theory which allow one to eliminate some of the communication places and of the waiting places, a reduced Petri Net model can be obtained where each object of the sequence diagram is represented by a non-preemptive resource. Some of these resources may be used by different scenarios (objects which belong to several sequence diagrams). As a consequence, the merging of these shared places produces a unique global Petri Net model which represents the global behaviour of the entire system. As was shown in [Julia 2000], explicit time constraints which exist in a Real Time System, can be formally defined using

a p-time Petri Net model. The static definition of a p-time Petri Net is based on static intervals which represent the permanency duration (sojourn time) of tokens in places and the dynamic evolution of a p-time Petri Net model depends on the time situation of the tokens (date interval associated with the tokens). Figure 4 shows the p-time Petri Net model

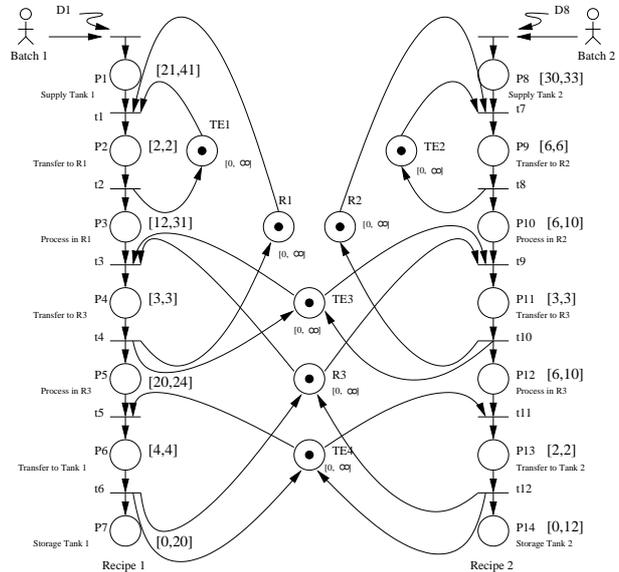


Fig. 4. p-time Petri net model for Recipe 1 and Recipe 2

which corresponds to the whole system (recipe 1 + recipe 2). Each recipe is represented by a production route (a sequence of transitions and places) and each object of the sequence diagrams is represented by a resource (a shared resource if the object appears in both recipes, like R3 for example). For example, as specified on the sequence diagram in figure 3, after the processing of batch 1 in R1, the resource R1 becomes available only when batch 1 is transferred from reactor R1 to reactor R3.

The static intervals associated to each place of the model in figure 4 and the execution beginning of each recipe, specified by the firing date of the first transition of each recipe (D1 for recipe 1 and D8 for recipe 2), will depend on specific production plans.

### III. CONFLICT SITUATION ANALYSIS IN A P-TIME PETRI NET

In p-time Petri Nets, as shown in [Julia 2000], conflict situations for shared resources are visible during a time interval (there exists the notion of Conflict Time Interval), and not at a single time point. For example, considering the p-time Petri Net of figure 5 (the Petri net fragment involved in the conflict for the shared resource R3), if a token appears in P3 at date  $D3=23$  and a token appears in P10 at date  $D10=36$ , then, the visibility intervals of these tokens are  $[(\delta_{p3})_{min}; (\delta_{p3})_{max}] = [12 + 23; 31 + 23] = [35; 54]$  for the token in P3 and  $[(\delta_{p10})_{min}; (\delta_{p10})_{max}] = [6 + 36; 10 + 36] = [42; 46]$  for the token in P10. The

conflict time interval associated to the pair  $(t_3; t_9)$  is given by the intersection of these visibility intervals:  $[35; 54] \cap [42; 46] = [42; 46]$ . So, an effective conflict between  $t_3$  and  $t_9$  is able to occur during the interval  $[42; 46]$ .

Using a class graph algorithm, it is possible to verify if a special class, called “death token class”, which represents a time constraint violation (the resource necessary in order to respect a time interval associated to a place of a p-time Petri Net is not available at the right time), can be reached when considering a Petri Net fragment involved in a specific conflict state. But the class graph has some limitations; the state space can be very large and the duration a token can remain in the same place has to be delimited by real numerical values and not symbolic ones. In particular, each time a new simulation is executed, new conflict time intervals are calculated and, for each conflict situation, a class graph has to be generated. As a consequence, the global simulation speed is reduced.

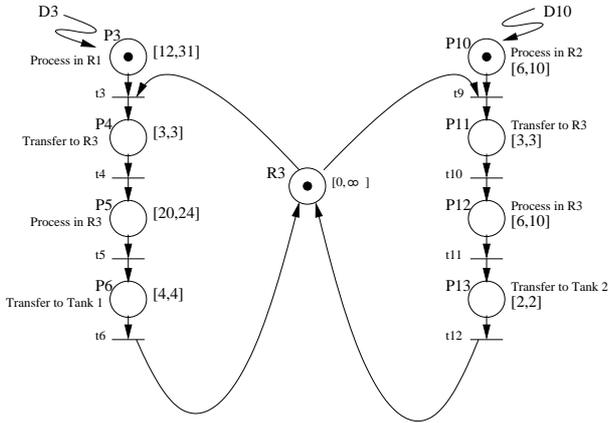


Fig. 5. Useful part of the conflict for R3

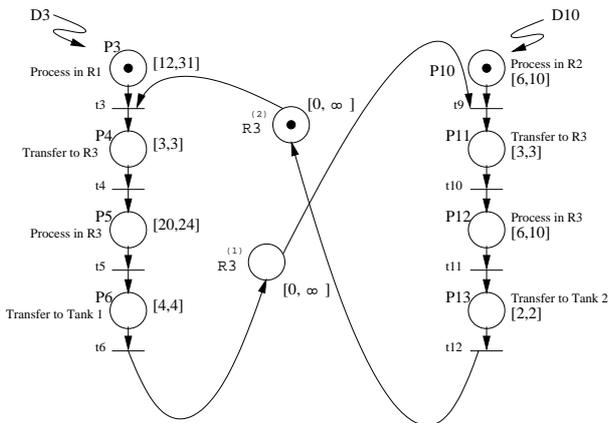


Fig. 6. Conflict Resolution for R3

A conflict situation for a shared resource in a p-time Petri Net is equivalent to a certain extent to a Watch Dog [Rivière 2001] (two transitions in structural conflict) represented by a t-time Petri Net (time

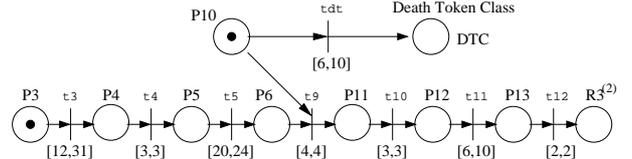


Fig. 7. Watch dog

intervals associated with transitions) where a specific place corresponds to the “death token class”. Watch Dogs are commonly used to analyse behaviours which deviate from their normal evolution. As there exists an equivalence between a p-time Petri Net and a t-time Petri Net in case of the earliest firing strategy [Khansa 1996], it will be possible to transform a conflict state given by a p-time Petri Net model into a t-time Petri Net model corresponding to a typical case of Watch Dog where one of the transitions in structural conflict represents the normal evolution of the system and the other one signals a time constraint violation. For example, considering the conflict situation for the shared resource R3 shown in figure 5, if the reactor R3 is used to treat batch 1 and, after that, batch 2, then, the Petri Net in figure 5 can be transformed into the Petri Net of figure 6. The problem is then to verify that making this decision, the reactor R3 will be available in place  $R3^{(1)}$  before the maximum bound of the visibility interval associated to the token in place P10, which represents the latest date authorized for the firing of transition  $t_9$ . On the contrary, a death token class will be reached. The Watch Dog modelled by the t-time Petri Net in figure 7 represents such a situation. In particular, this Watch Dog specifies that, at date  $D10 + 10$  ( $D10$  is the date when the token appears in P10 and 10 is the maximum bound of the time interval associated to transition  $t_{dt}$ ), if the transition  $t_9$  has not been fired yet (which means that the resource in  $R3^{(1)}$  is not available at this moment for the firing of  $t_9$  in figure 6), then the transition  $t_{dt}$  has to be fired and a death token class, which corresponds to a time constraint violation, will be reached. The place  $R3^{(1)}$  does not appear in the Watch Dog because its static interval is  $[0; \infty[$  and, as a consequence, the effect of this place on the temporal reasoning is useless.

In [Rivière 2001], it was shown that applying the sequent calculus of Linear Logic to a Watch Dog represented by a t-time Petri Net and using a non conventional  $(\max; +)$  algebra, production and consumption symbolic dates assigned to each atom (token) involved in the studied scenario can be calculated. In particular, comparing the symbolic dates of the tokens which belong to the places in structural conflict, it is possible to know if the Watch Dog effectively invalidates the normal operations, i.e. if a “death token class” can be reached.

With Linear Logic [Girard 1987], a marking  $M$  is a monomial in  $\otimes$ , that is a marking represented

by  $M = A_1 \otimes A_2 \otimes \dots \otimes A_k$  where  $A_i$  are place names. For instance, the initial marking on the Petri Net in figure 7 is  $P3 \otimes P10$ . A transition is an expression of the form  $M_1 \multimap M_2$  where  $M_1$  and  $M_2$  are markings. For example, transition  $t_3$  on the Petri Net in figure 7 is noted  $t_3 = P_3 \multimap P_4$ . A sequent  $M, t_i \vdash M'$  represents a scenario where  $M$  and  $M'$  are respectively the initial and final markings, and  $t_i$  is a list of non-ordered transitions. A sequent can be proved by applying the rules of the sequent calculus as there exists an equivalence between Petri Nets reachability and the proof of sequents in Linear Logic [Girault 1997]. A Linear Logic proof tree is read from the bottom, up and a proof stops when all the leaves of the tree are identity sequents ( $P_1 \vdash P_1$ , for example).

In a proof tree, each transition firing generates a symbolic date associated to each atom (token) as shown in [Rivière 2001]. In this article,  $D_i$  will denote a date and  $d_i$  a duration associated to a transition firing. A pair  $(D_p, D_c)$  will be associated to each atom of the proof tree; they respectively represent the production and the consumption date of atoms.

From the watch dog of figure 7, two scenarios can be derived :

$$\begin{aligned} Sc1 &= P3 \otimes P10, t_3, t_4, t_5, t_{dt} \vdash P6 \otimes DTC \\ Sc2 &= P3 \otimes P10, t_3, t_4, t_5, t_9, t_{10}, t_{11}, t_{12} \vdash R3^{(2)} \end{aligned}$$

These scenarios are in conflict i.e. one scenario invalidates the other. As a matter of fact, Sc2 represents the normal evolution of the system when Sc1 represents a time constraint violation (a “death token class”). Table 1 shows the production and consumption dates in the scenario Sc1 case. Table 2 and 3 show the production and consumption dates in the scenario Sc2 case.

In a t-time Petri Net model, any enabling duration  $d_i$  takes its values within a time interval  $[\delta_{i,min}; \delta_{i,max}]$ . For example, when considering the scenario Sc1, the domain for the consumption date of atom P4 (the token in P4) is given by the time interval  $[D_3 + d_{3,min} + d_{4,min}; D_3 + d_{3,max} + d_{4,max}]$ . Based on a property shown in [Rivière 2001], it is possible to say that Sc1 invalidates Sc2 (which means that the death token class will be reached) if the maximum value of the enabling duration of  $t_{dt}$  in scenario Sc1 is smaller than the minimal sojourn time of atom P6 in scenario Sc2. For example, when considering the Petri Net in figure 5, if a token appears in P3 at date  $D3=23$ , its visibility interval is  $[35 ; 54]$ , and, if a token appears in P10 at date  $D10=36$ , its visibility interval is  $[42 ; 46]$ . Then, it seems natural to fire transition  $t_3$  at date 35 when the token in P3 becomes available. From the symbolic production and consumption dates obtained when scenarios Sc1 and Sc2 are considered, the following results are obtained : the minimal production date of atom P6, which is equal to  $D_3 + d_{3,min} + d_{4,min} + d_{5,min} =$

Transition	Consumption Date	Production Date
$t_3 = P3 \multimap P4$	$P3(D3+d3)$	$P4(D3+d3)$
$t_4 = P4 \multimap P5$	$P4(D3+d3+d4)$	$P5(D3+d3+d4)$
$t_5 = P5 \multimap P6$	$P5(D3+d3+d4+d5)$	$P6(D3+d3+d4+d5)$
$t_{dt} = P10 \multimap DTC$	$P10(D10+ddt)$	$DTC(D10+ddt)$

Table 1. Consumption and Production dates for Scenario 1

Transition	Consumption Date
$t_3 = P3 \multimap P4$	$P3(D3+d3)$
$t_4 = P4 \multimap P5$	$P4(D3+d3+d4)$
$t_5 = P5 \multimap P6$	$P5(D3+d3+d4+d5)$
$t_9 = P10 \otimes P6 \multimap P11$	$P6(\max(D3+d3+d4+d5, D10)+d9)$ $P10(\max(D3+d3+d4+d5, D10)+d9)$
$t_{10} = P11 \multimap P12$	$P11(\max(D3+d3+d4+d5, D10)+d9+d10)$
$t_{11} = P12 \multimap P13$	$P12(\max(D3+d3+d4+d5, D10)+d9+d10+d11)$
$t_{12} = P13 \multimap R3^{(2)}$	$P13$ $(\max(D3+d3+d4+d5, D10)+d9+d10+d11+d12)$

Table 2. Consumption dates for Scenario 2

$23 + 12 + 3 + 20 = 58$ , is bigger than the maximal consumption date of atom P10, which is equal to  $D10 + d_{dt,max} = 36 + 10 = 46$ . As a direct consequence of this result, the firing of  $t_3$  as soon as the token in P3 becomes available will not be allowed by the conflict resolution mechanism which will be used at the global simulation level.

#### IV. Simulation principle

One of the approaches which allows one to execute dynamically a Petri Net is the one based on a token player algorithm. A token player algorithm is a special inference mechanism which allows the firing of the enabled transitions. When the model is based on a p-time Petri Net, the token player algorithm must take into account the conflict situations in real time in order to avoid the possibility of deadlock which can be caused by a “death token class”. The basic principle of such an algorithm was presented in [Julia 2000]. Figure 8 shows how this algorithm works. The basic difference between our proposed token player and other token player algorithms used in simulation tools based on Petri Nets

Transition	Production Date
$t_3 = P3 \multimap P4$	$P4(D3+d3)$
$t_4 = P4 \multimap P5$	$P5(D3+d3+d4)$
$t_5 = P5 \multimap P6$	$P6(D3+d3+d4+d5)$
$t_9 = P10 \otimes P6 \multimap P11$	$P11(\max(D3+d3+d4+d5, D10)+d9)$
$t_{10} = P11 \multimap P12$	$P12(\max(D3+d3+d4+d5, D10)+d9+d10)$
$t_{11} = P12 \multimap P13$	$P13(\max(D3+d3+d4+d5, D10)+d9+d10+d11)$
$t_{12} = P13 \multimap R3^{(2)}$	$R3^{(2)}$ $(\max(D3+d3+d4+d5, D10)+d9+d10+d11+d12)$

Table 3. Production dates for Scenario 2

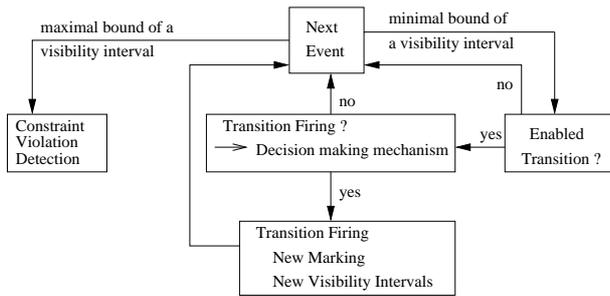


Fig. 8. Token Player Algorithm for a p-time Petri net

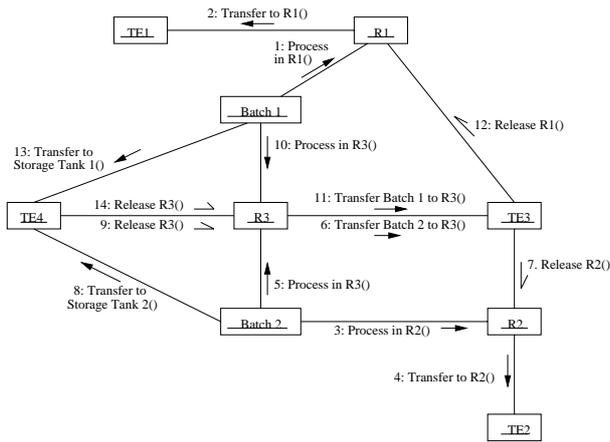


Fig. 9. Collaboration Diagram

models is that transitions are not fired necessarily as soon as they become enabled because of the conflict resolution mechanism presented in the previous section. For example, if the input transitions of P1 and P8 in figure 4 are fired at date  $D1=D8=0$  (beginning of recipes execution), the result of the global simulation based on the token player algorithm is the following one : date 21 = t1 fired; date 23 = t2 fired; date 30 = t7 fired; date 35 = t3 enabled, but not fired; date 36 = t8 fired; date 42 = t9 fired; date 45 = t10 fired; date 51 = t11 fired; date 53 = t12 fired; date 53 = t3 fired; date 56 = t4 fired; date 76 = t5 fired; date 80 = t6 fired. As a result of the token player execution, an acceptable scenario corresponding to a specific firing sequence is obtained and can be translated into the collaboration diagram of figure 9 which represents the global behaviour of the Real Time System from the UML point of view. In this diagram, it is clear that batch 2 has to be processed in reactor R3 before batch 1 in order to respect the time constraints.

## V. CONCLUSION

The principal advantage shown in this article is the possibility of accelerating the conflict resolution in a p-time Petri Net. This is realized by using the sequent calculus of Linear Logic and a non conventional (max;+) algebra which allows the calculation of symbolic dates, instead of real numerical

values. As a direct consequence, the generation of a class graph, each time a conflict situation is met, is not necessary anymore and the global duration of simulation when considering Real Time UML specifications is reduced.

## REFERENCES

- [Cardoso 2001] Cardoso, J., Sibertin-Blanc. (2001). Ordering actions in Sequence Diagrams of UML. *23 International Conference on Information Technology Interfaces*. Croatia.
- [Girard 1987] Girard, Jean-Yves. 1987. Linear Logic. *Theoretical Computer Science*. 50:1-102.
- [Girault 1997] Girault, F. 1997. *A logic for Petri nets, JESA Vol. 31, n.3, Edition Hermes*
- [Julia 2000] Julia, S., Valette, R. (2000). Real Time Scheduling of Batch Systems. *Simulation Practice and Theory, Elsevier Science*. pp. 307-319.
- [Julia 2002] Julia, S., Kanacilo, E., M. (2002). An approach based on dynamic UML diagrams and on a token player algorithm for the scenario verification of real time systems. *14TH European Simulation Symposium, Simulation in Industry*. Dresden, German. p. 377-381.
- [Khansa 1996] Khansa, W., Aygaline, P., Denat, J. P. (1996). Structural analysis of p-time Petri Nets. *Symposium on discrete events and manufacturing systems. CESA'96 IMACS Multiconference*. Lille, France.
- [Murata 1989] Murata, T. (1989). Petri Nets: Properties, analysis and applications. *Proceedings of the IEEE 77(4)*. pp. 541-580.
- [OMG 1999] OMG Unified Modeling Language Specification version 1.3. *Object Management Group*.
- [Rivière 2001] Rivière, N., Pradin-Chezalviel, B., Valette, R. (2001). Reachability and temporal conflicts in t-time Petri Nets. *9TH International Workshop on Petri Nets and Performance Models (PNPM'01)*. Aachen-Germany. pp. 229-238.

STÉPHANE JULIA received his Ph. D. degree from the University "Paul Sabatier" of Toulouse (France) in 1997. He is currently a Professor at the Federal University of Uberlândia (Brazil) in the Computer Science Department. His current research interests include the application of Petri net theory in Real Time Systems. He is also interested in the relationships between Petri nets and UML notation.

MICHEL dos S. SOARES received his degree in Computer Science from the Federal University of São Carlos (Brazil) in 2000. He is currently doing his Master's Degree at the Federal University of Uberlândia (Brazil), where he is researching on UML, Petri nets and Linear Logic. He is also a Computer Science professor, in the field of Software Engineering.

# UNPACED AND PACED SIMULATION FOR TESTING AGENTS

Adeline M. Uhrmacher

Mathias Röhl

Jan Himmelspach

Universität Rostock, Department of Computer Science

Albert-Einstein-Str. 21, D-18059 Rostock, Germany

## KEYWORDS

Parallel, distributed Simulation, Multi-Agent Systems, Paced and Unpaced Simulation

## ABSTRACT

Agents are autonomous software aimed at working in dynamic environments and thus form a specific type of embedded software systems. To test this type of software simulation systems can be successfully employed. Agents might be modeled, be partly embedded in, or coupled to the virtual environment they are tested in. Depending on the degree of being embedded in the virtual environment, the type of execution that supports an efficient and effective simulation varies. In JAMES (A Java-Based Agent Modeling Environment for Simulation) different simulators have been implemented. Unpaced and paced simulators support interaction in simulation- and real-time differently. Moving from unpaced to paced execution, the simulator exercises less control over the experiment and the coupling between simulation and agents to be tested is loosened.

## 1 INTRODUCTION

Agents can be interpreted as software systems that are aimed at working autonomously in dynamic and uncertain environments (Jennings et al. 1998). The interrelations between agents and simulation are manifold (Uhrmacher et al. 2001). Software agents are used to develop “state of the art” simulation systems and agents are used as a metaphor for modeling dynamic systems as collections of autonomously interacting entities. Software agents are often mission critical (or even safety critical) and, like other software systems, must be tested and evaluated before being deployed. Their autonomy

and the open heterogeneous nature of the environment in which they operate make testing and evaluation more difficult than in the case of more conventional software systems. As agents are aimed at working in dynamic environments, simulation seems a natural approach towards testing the behavior of an agent system in interaction with its environment.

The implementation and application of dynamic test scenarios for multi-agent systems require considerable modeling effort. Already early simulation systems for agents allowed to plug code fragments, or single modules into the skeleton of an agent model (Montgomery and Durfee 1990). Others treat agents as external source and drain of events (Pollack 1996). The continuity of models from specification via simulation to implementation shall help reducing flaws during the design of software systems (Hu and Zeigler 2002). However, it is less the continuity of models during designing agent systems we will discuss in this paper but the different types of simulators that accompany the different stages of developing agents.

Based on JAMES (Schattenberg and Uhrmacher 2001) (A Java based Agent Modeling Environment for Simulation) and based on the project AUTOMINDER (Pollack et al. 2003) that is aimed at developing a planning agent software supporting elderlies in their homes, we will illustrate our approach for testing agents.

## 2 JAMES

JAMES has been developed based on the formalism DYNDEVS. The model design of JAMES resembles that of DEVS (Discrete Event System Specification) (Zeigler et al. 2000) extended by means for reflection which allows agents to adapt their composition, interaction, and behavior patterns. Models can create new models and add them to the coupled model they belong to, they can remove themselves, and they can access their interaction structure. To initiate structural changes outside their

boundary, agents have to turn to communication and negotiation. Thus, a movement from one coupled model to another implies that another atomic model complies with the request to add the moving model into the new interaction context. To facilitate modeling, all atomic models are equipped with default methods that allow them to react to those requests. However, these default reactions can be suppressed to decide deliberately what requests shall be executed. The freedom to decide whether to follow a certain request, e.g. to commit suicide, and its knowledge, i.e. beliefs about itself and its environment, distinguish active agents from more “reactive” entities (Jennings et al. 1998). The ports of DEVS models, which are used for communication between models, are complemented by peripheral ports in JAMES. Models communicate via peripheral ports with processes that are external to the simulation. Time models allow to transform the resource consumption of the external processes into simulation time (Schattenberg and Uhrmacher 2001; Riley and Riley pear). The modular, hierarchical modeling concept facilitates the re-use of components and thus the construction of virtual test environments by composition.

JAMES has been used for testing different types of software agents. As other simulation systems (Montgomery and Durfee 1990) it allows to plug in code fragments, or single modules, whereas the agent itself is specified as part of the model. We followed this approach by testing planning agents in a TILEWORLD scenario in JAMES (Schattenberg and Uhrmacher 2001). In later phases, the ability to execute agents as they are, and to switch arbitrarily between an execution in the real environment and the virtual test environment gains importance. If simulator and agent software are only loosely coupled (Pollack 1996; Anderson 1997), agents are typically only perceivable by their effects in the virtual environment and no longer really controlled by the simulation. To bridge the gap between earlier and later phases in designing agents and to support the continuity of models, the idea of representatives has been introduced in JAMES. The idea took concrete form in testing and plugging agents of the mobile agent system MOLE into JAMES (Uhrmacher et al. 2002).

### 3 THE EXAMPLE - AUTOMINDER

AUTOMINDER is a software agent designed as a cognitive orthotic which shall assist elderlies with memory impairment in carrying out their daily life activities. Therefore, the activities of elderlies are monitored and elderlies, if they forget or confuse certain activities, shall be reminded in a timely and adequate manner (Pollack et al. 2003; McCarthy and Pollack 2002; et al 2002). The software is being developed in the context of the

Initiative on Personal Robotic Assistants for the Elderly (Montemerlo et al. 2002) and installed on the nursebot PEARL.

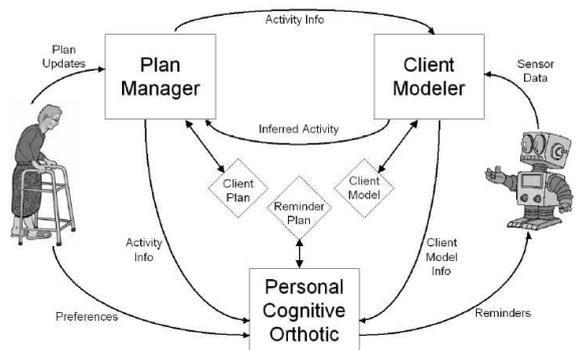


Figure 1: Structure of the AUTOMINDER Agent (Pollack et al. 2003)

The architecture of AUTOMINDER comprises the components: Client Modeler (CM), Plan Manger (PM) and Personal Cognitive Orthotic (PCO). The role of the CM is to interpret the sensor information and based on a given plan to identify activities that are just started or being ended and to notify the Plan Manager. The CM maintains and updates a client model which contains observations and is used to derive regular behavior patterns of the elderly. The Plan Manager maintains a plan in terms of activities the elderly is supposed to perform, i.e. the client plan. Subsequently the plan can be updated by the caregiver and to a certain degree also by the elderly. The plan manager checks the client plan for inconsistencies and is responsible for resolving potential conflicts. Based on the client plan and the client model, the PCO finally decides what kind of reminders to launch at which time.

The AUTOMINDER agent combines reactive and deliberative abilities, like planning and learning. Taking opportunities into account while not losing track of pursued goals is seen as one of the central challenges in developing successful agent software (Wooldridge and Ciancarini 2000). Therefore, the mediation between reactivity and deliberation and its effect on the performance of agents has traditionally been at the core of evaluating hybrid agents in small play world scenarios e.g. (Cohen et al. 1989; Kinny et al. 1996; Schattenberg and Uhrmacher 2001).

## 4 TOWARDS A MODEL-BASED TESTING OF AUTOMINDER

Some time has elapsed since Paul Cohen, Steve Hanks, and Martha Pollack wrote their paper on controlled experimentation, agent design, and associated problems (Hanks et al. 1993). Their controversy about testing in the small and testing in the large in designing agent systems has neither lost its topicality nor its virtue, though. Test beds, e.g. DVMT (Durfee 1988), PHOENIX (Greenberg and Westbrook 1990), TILEWORLD (Pollack and Ringuette 1990), soccer game (Kitano et al. 1997), and large scale disasters (Kitano et al. 1999) represent a complement to conventional benchmark tests, offering test scenarios which are aimed at revealing prototypical problems in dynamic environments. Within this testing in the small, it is not the purpose to confront the agent with a valid model of the concrete environment the agent shall dwell in. In contrast “testing in the large” is based on test cases that shall emulate requirements of the real environment. Often test cases are based on and sometimes even automatically generated from software requirements, source-code statements, and module interfaces (e.g. (Peraire et al. 1998)). However, whereas for many embedded systems a clear specification of the software and the required functionality exists, this is not necessarily true for agent systems. Instead Wooldridge and Jennings observe that “the development of any agent system - however trivial is essentially a process of experimentation” (Jennings and Wooldridge 1998). Therefore, experimentation has been part of developing agents from the very first.

AUTOMINDER’s activities are constrained by time: it has to react timely and appropriate to the elderlies actions and reactions. Most agent’s decisions and deliberations are limited by time. The timeliness and adequacy of their reaction determine their performance. AUTOMINDER’s activities are triggered by the flow of time: many of the activities of elderlies are scheduled for certain times of the day and the robot has to remind the elderly in time if these activities are crucial for the elderlie’s health. So AUTOMINDER displays situation-triggered and time-triggered activities.

If AUTOMINDER is tested in its real environment, interaction happens in physical time. A significant evaluation of the performance of AUTOMINDER would likely take at least a month — besides the problem to find suitable test persons for the experiments. The functionality of AUTOMINDER can not be tested based on one time point only. Its interaction with the environment has to be observed over a period of time. Thus, simulation seems a natural approach towards testing the behavior of this agent system in interaction with its environment.

Environment models are used to generate the differ-

ent test cases dynamically during simulation, including specific interaction patterns and time constraints (Schütz 1993, p.23). The focus of model-based testing shifts from the specification of the software to modeling the dynamic environment of the agent.

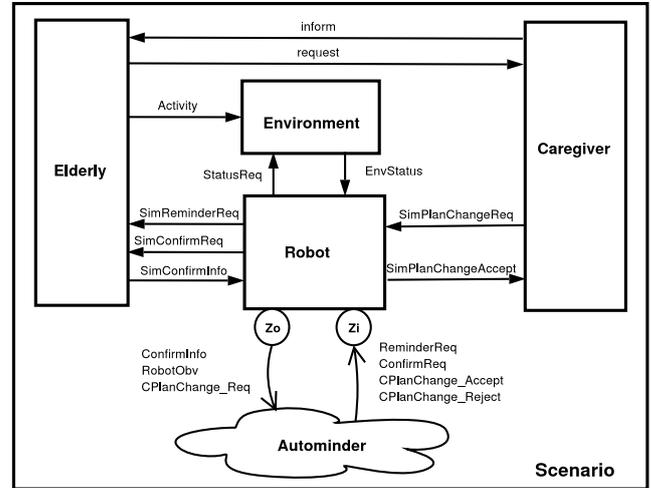


Figure 2: Structure of the JAMES Model for Testing the AUTOMINDER Software

Using simulation for testing software means to develop software, including models, routines for initialization of models and the simulation engine. Particularly, if software is used for testing other software, it is crucial that certain quality characteristics of the software product, including accuracy, can be assured. Component-based development of software is valued as an effective and affordable way to facilitate verification, validation and accreditation of modeling and simulation applications (Balci et al. 2002).

To ease the process of obtaining confidence that the model can be considered valid for its intended application (Sargent 1999), a component-based design of the model, where model components can be inspected separately, and the simulation system, which supports different approaches towards testing (see next section), has been chosen. The virtual environment in which AUTOMINDER shall be tested is built up of four different model components (Fig. 2) that can be evaluated separately. The model component **Elderly** represents the client to be supported by AUTOMINDER. The **Robot** represents a nursebot endowed with AUTOMINDER. Coupling AUTOMINDER, which runs concurrently and externally to the simulation, and the virtual environment is done by utilizing the **Robot** component as an interface between simulation and agent software. AUTOMINDER sends its time labeled events to the robot who charges its output ports with these events at the specified time. From the ports they are automatically transferred to the other models as defined by the couplings. The robot has to

explicitly request new status information about its environment from the model `Environment`. The model `Caregiver` forwards new plans to the `Robot` and `AUTOMINDER`.

The intention of our experiments with `AUTOMINDER` is behavioral testing or black box testing. The goal of `AUTOMINDER` is to provide elderlies with timely and appropriate reminders. This implies that `AUTOMINDER` has to find a balance between maximizing the elderlies compliance in performing his or her daily activities, maximizing the satisfaction of elderly and caregiver with the system, and avoiding making the elderly overly reliant on the system (Pollack et al. 2002). To achieve these goals the system has to be adaptable to different types of elderlies, actors, and circumstances which have to be represented with a sufficient accuracy.

So far `AUTOMINDER` has been tested manually: a user informs interactively `AUTOMINDER` about activities of the elderly and advances the virtual time. The goal of experiments based on `JAMES` is to test the behavior and the adaptation strategies of `AUTOMINDER` by using different model components representing explicitly different types of elderlies and domestic environments.

## 5 SIMULATOR

Executing the model according to the user’s specification and the given initial situation is the task of a discrete event simulator. Simulation models are interpreted and executed by a tree of processors, which reflect the hierarchical compositional structure of the model (Fig. 3). Each of the processors is associated with a component of the model and is responsible for invoking the component’s methods and controlling the synchronization by exchanging messages with the other processors of the processor hierarchy. The change of model structure is reflected in an according change of the processor tree. Different distributed, parallel execution strategies have already been implemented in `JAMES` based on the abstract simulator introduced in `DYNDEVS` (Uhrmacher and Gugler 2000; Uhrmacher and Kraemer 2001). Whereas one adopts a conservative strategy where only events which occur at exactly the same simulation time (including starting external processes) are processed concurrently, two other strategies split simulation and external processes into different threads and allow simulation and deliberation to proceed concurrently by utilizing simulation events as synchronization points. All simulators currently execute in an unpaced mode which means that simulation time does not elapse in relation to wall clock time but jumps as fast as possible from one event to the next, neglecting the simulation time (and thus the represented physical time) that lies

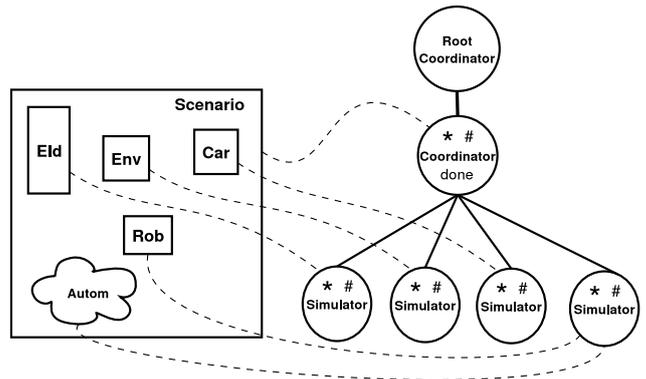


Figure 3: Models and Corresponding Simulator Tree in `JAMES`

inbetween (Fujimoto 2000). Within the limits that are determined by the wall clock time required to execute events, paced simulations can be scaled to allow a faster or a slower progression of time. The scaling factor can be changed during simulation, to skip in a fast mode through less interesting episodes and to zoom in to explore interesting episodes in detail. Paced simulations are typically used for training humans, whereas unpaced simulations are used for analytical purposes. Both allow the interaction with humans and with external soft- or hardware. However, their means and also their focus is different.

### 5.1 UNPACED SIMULATOR

The following conservative simulator supports an as-fast-as-possible discrete-event-simulation and exploits the parallelism inherent in concurrently deliberating multiple agents.

Figure 4 describes the `*`-handler of the simulator, which is at the core of the overall simulation algorithm. Besides a `*`-handler, simulators are equipped with a `#`-handler for dealing with inputs. Coordinators have additionally a `done`-handler which records that events have been processed or guarantees are given.

The simulator of a model is activated by the `*`-message, which indicates an internal, external, or confluent event. With the label *guarantee?* set to true the `*`-handler of the simulator is asked to guarantee that none of its pending external processes will finish before the time  $t$ .

The set *busy* is updated by the external agent processes which adds  $busy_i$  to it in the moment the process  $i$  is started. If the process has finished,  $busy_i$  is deleted from the set *busy*. The resource consumption of the process is recorded, as are the results with which the

---

```

when an input  $(*, guarantee?, xCount, t)$  has been received
am is the associated model
outCount = 1
if guarantee? then
  block until for all  $i \in busy_{fixed}$ 
     $t_{start_i} + timeModel(i) > t \vee i \notin busy$ 
  justFinished =  $\{i | i \in busy_{fixed} \setminus busy\}$ 
  for all  $i \in JustFinished$ 
     $t_{finished_i} = t_{start_i} + timeModel(i)$ 
     $t_{finished} = t_{finished} \cup \{t_{finished_i}\}$ 
     $t_{start} = t_{start} \setminus \{t_{start_i}\}$ 
     $busy_{fixed} = busy_{fixed} \setminus \{busy_i\}$ 
  send (done,  $\min(t_{next}, t_{finished}), \emptyset$ ,
    outCount, ( $busy_{fixed} \neq \emptyset$ )) to parent coordinator
else
  if  $\neg guarantee?$  then
    inpCount = xCount
     $t_{min} = \min(t_{finished}, t_{next})$ 
     $t_{finished} = t_{finished} \setminus \{t_{finished_i} \in t_{finished} | t_{finished_i} = t\}$ 
    if  $t = t_{min}$  then
      if  $t = \min(t_{finished})$  then flush  $z_i$ 
      send ( $\lambda(s, z_i)$ ) to parent
      if xCount = 0 then
         $(s, z_o) = \delta_{int}(s, z_i)$ 
      else
        block until inpCount = 0
         $(s, z_o) = \delta_{con}(s, xb, z_i)$ 
      endif
    else
      block until inpCount = 0
       $(s, z_o) = \delta_{ext}(s, t - t_{last}, xb, z_i)$ 
    endif
  for all  $i \in busy \setminus busy_{fixed}$ 
     $t_{start_i} = t$ 
     $t_{start} = t_{start} \cup \{t_{start_i}\}$ 
     $busy_{fixed} = busy_{fixed} \cup i$ 
  am =  $\rho(s)$ 
   $t_{last} = t$ 
   $t_{next} = t_{last} + ta(z, s)$ 
  send (done,  $\min(t_{next}, t_{finished})$ ,
    varStrucRequest(s), outCount, ( $busy_{fixed} \neq \emptyset$ ))
    to parent coordinator
  endif
endif
end

```

---

Figure 4: The  $*$ -handler of the unpaced simulator in JAMES

peripheral input ports  $z_i$  shall be charged. The procedure which starts the external process within a separate thread generates unique names for the processes with which its start time, the finish time, and the results are labeled.  $busy_{fixed}$  contains the processes that the simulator believes to be running.  $busy$  contains the processes which are actually running.  $t_{start}$  embraces the starting time of all processes the simulator believes to be running. It is incremented when a new process is started and decremented when the simulator discovers a completion.  $t_{finished}$  lists the completion time of all processes of whose completion the simulator is aware.

If at least one external process is running the simulator blocks until each of the processes running has

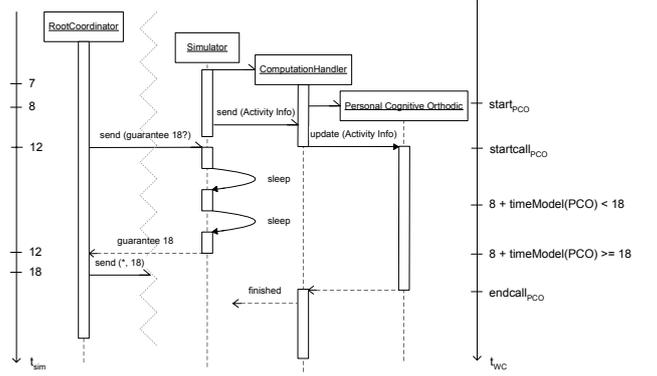


Figure 5: The Interaction between Simulation and AUTOMINDER in the Unpaced Version in JAMES

either reached the current simulation time or has been finished. The processes which just have been finished are determined, and for all of them the virtual time of finalization is calculated. The sets  $t_{finished}$ ,  $t_{start}$  and  $busy_{fixed}$  are updated.

If *guarantee?* is set to false, the  $*$ -message announces an event and a confluent  $\delta_{con}$ , internal  $\delta_{int}$ , or external  $\delta_{ext}$  transition is due. If at the current time the completion of a deliberation process is scheduled, the peripheral port  $z_i$  is charged with the results of this deliberation process. The transition functions update the state and generate outputs which are directed towards externally running software, e.g. invoking a deliberation process. After executing the transition function it is checked whether new processes have been started. Their starting time is determined and the sets  $t_{start}$  and  $busy_{fixed}$  are updated. Afterwards structural changes are executed at the level of the atomic model by invoking the model transition  $\rho$ . The time of last and time of regular next event are determined. The coordinator is informed of the time of next event and whether any structural changes at the level of the coupled model are due. If agents deliberate sufficiently long and a sufficient number of processors are available, multiple deliberating agents can be executed nearly at the cost of one deliberating agent (Uhrmacher and Kraemer 2001).

In the unpaced simulation (Fig. 5), the simulator controls the execution of agents, by invoking the methods, recording the simulation time when the external software has been started and by transforming the resources consumed during the execution of external software into simulation time to determine when the execution has been finished in simulation time. The simulation, its view of the world and particularly its view on time, controls the experiment: the software is executed embedded in the virtual environment. Performance criteria can be calculated, and bottlenecks can be identified. The user can interactively set parameters by exe-

cutting the simulation in stepping mode or by introducing break-points in the simulation. This version of the simulator is suitable if separate modules of AUTOMINDER shall be tested, e.g. the personal cognitive orthotic and specific methods of AUTOMINDER are invoked. The orthotic module of AUTOMINDER receives information about the elderly including his or her whereabouts and the originally planned schedule of the elderly. This information is provided by the client and plan manager module of AUTOMINDER. Part of its functionalities are now part of the modeled robot. In latter stages of designing AUTOMINDER the control of the simulator might be slightly loosened.

## 5.2 PACED SIMULATOR

The following simulator is a first attempt to support a paced simulation and an asynchronous exchange of messages between simulator and external software. In paced simulation: each advance in simulation time is paced to occur in synchrony with a scaling factor times an equivalent advance in wall clock time.

---

```

when an input (*, xCount, t) has been received
  am is the associated model
  outCount = 1
  inpCount = xCount
  block until W2S(WallclockTime)1 >= t
  flush zi
  if t = tnext then
    send (λ(s, zi)) to parent
    if xCount = 0 then
      (s, zo) = δint(s, zi)
    else
      block until inpCount = 0
      (s, zo) = δcon(s, xb, zi)
    endif
  else
    block until inpCount = 0
    (s, zo) = δext(s, t - tlast, xb, zi)
  endif
  am = ρ(s, zi)
  tlast = t
  tnext = tlast + ta(s, zi)
  send (done, tnext, varStrucRequest(s), outCount) to
    parent coordinator
end

```

1)  $W2S(t_{wc}) = t_{simStart} + scale * (t_w - t_{wcStart})$

---

Figure 6: The \*-Handler of a Simulator in JAMES - Paced

Each simulator blocks until its local virtual time has reached the wall clock time (Fig. 6). The simulator is a scaled paced simulator which allows to let the simulation run twice or half as fast as wall clock time. The speed up of simulation is constrained by the execution speed of simulation events and by the execution speed of the external software. Every time an event takes place

---

```

tnext = tnext(topmost coordinator)
repeat until tnext > tEndOfSimulation ∨ (tnext = ∞)
  while W2S(wallclockTime) < tnext - ε
    if externalMessageArrived
      externalMessageArrived = False
      tnext = W2S(wallclockTime)
    endif
  endwhile
  send (*, 0, tnext) to topmost coordinator
  wait for (done, t, varStruc, outCount)
  from topmost coordinator
  tnext := t
end

```

---

Figure 7: The Root Coordinator in JAMES

the peripheral ports are flushed, i.e. they are read and emptied afterwards. Simulator and external software exchange messages in an asynchronous manner.

The simulation is notified that messages from external software systems have been arrived and thus have to be processed. The notification is propagated up the tree towards the root coordinator. For propagating the notification, the done threads of the coordinators are used which mark the coordinators along the way and thus allow afterwards to trigger top down the correct components of the processor tree. During their propagation upward, notifications are caught by \*-messages traveling down the processor tree. In this case the current simulation pulse will be used for processing. Only if a message arrives in a sufficiently large gap inbetween events scheduled in the simulation, it will reach the root coordinator and trigger a simulation pulse.

The root coordinator is traditionally responsible for advancing simulation time in JAMES. Also in the paced variant it controls the advance in simulation time. Therefore, it blocks the simulation for some time before informing its children. If this were not the case and only the simulators were responsible for blocking the simulation, an external software could not trigger any event before the next scheduled event at  $t_{next}$  would have been executed. Holding the simulation at the root coordinator allows that the external software can trigger events between the current simulation time and the time of scheduled events. The  $\epsilon$  accounts for real time delays that are caused by propagating messages from the root to the simulators. The idea is to choose  $\epsilon$  sufficiently large, so that the simulators are triggered by the \*-message in advance to be able to block  $W2S(WallclockTime)$ <sup>1</sup>  $\geq t$  and do not fall behind the wall clock time. However, this blocking time of the simulators should be minimized. Since during the time the simulators (and coordinators) are processing events, the root coordinator is waiting for done messages and will not acknowledge incoming messages from external

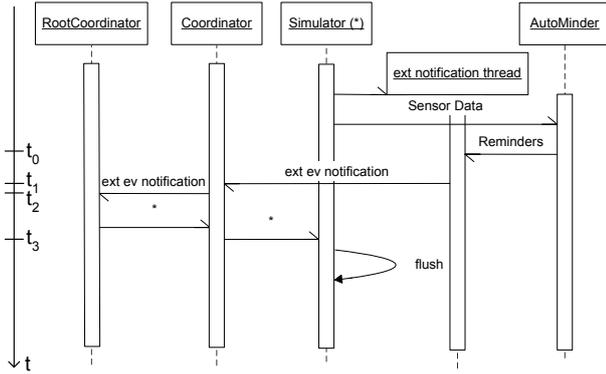


Figure 8: The Interaction between Simulation and AUTOMINDER in the Paced Version in JAMES

software systems. Messages will be handled promptly by the simulator however with a possible delay which is caused by propagating messages up and down the processor tree, and processing current events (Fig. 8).

Time stepped execution is the predominant model for paced simulation. Particularly in combination with visualization the procedure seems appropriate. The simulation is executed as a series of discrete intervals of time  $\Delta t$ . In each time interval messages are retrieved, the new internal state is computed, and messages are sent to other logical processes. After this the internal time is incremented by  $\Delta t$  and suspended until the time is reached. All messages are delivered in receive order and assumed to be relevant for now. Zeigler and his students adapted this approach and implemented a real-time event-based simulator (Cho et al. 2000). As the simulator exchange messages in receive order and not in time stamp order, no coordinator or root coordinator is necessary. Compared to this procedure of real-time execution our paced execution realizes a kind of hybrid. The proposed paced simulator asynchronously communicates with external software, both, simulator and agents, exchange information in receive order. However, within the simulation events are executed in strict time stamp order and resemble the typical analytic simulation. By supporting time stamped events in the distributed simulation of JAMES an overhead is induced. E.g. currently events that are initiated by messages from external software are processed at a simulation time that lags wall clock time. Even though first experiments have shown that the delay seems not critical, a more systematic analysis of the behavior of the simulator is necessary. For this purpose yet another real-time simulator is currently being implemented that will process events in receive order and not in time stamp order. In both simulators the problem of ensuring repeatability of simulation needs still to be addressed (McLean and Fujimoto 2000).

The role of the robot model is currently only to mediate between the dynamic virtual environment and AUTOMINDER: it frequently requests status information about the environment, information from the environment are forwarded to AUTOMINDER, and the output of AUTOMINDER is redirected into the simulation model and forwarded to the elderly. In other experiments, e.g. when the information about the elderly is transmitted directly by sensors in the flat, more detailed models about the soft- and hardware environment of the AUTOMINDER system will be probably required.

The paced version is used to loosely couple the entire AUTOMINDER software to JAMES. To tune the simulation to run faster than real time, the internal clock of AUTOMINDER has to pace time with the same scale as the simulator does.

## 6 DISCUSSION

Testing activities support quality assurance by gathering information about the nature of the software being studied. Little work has been done so far on developing methods for testing agents (Dam and Winikoff 2003). Whereas verification deals with transformational accuracy, validation deals with behavioral or representational accuracy. The analysis of temporal properties of software can be done statically or dynamically, the later is based on the execution of software.

Same as the functionality of real-time systems and embedded systems, the functionality of agent systems can not be evaluated based on an a priori fixed input specifications, but only as a course of reactions to the evolution of an environment. As agents are aimed at working in dynamic environments, simulation seems a natural approach towards testing the behavior of an agent system in interaction with its environment. Its interaction with the environment has to be observed over a period of time. The usage of a virtual environment in contrast to the real environment typically reduces costs and efforts and allows to test system behavior in “rare event situations”. Virtual environments are easier to observe and to control, and probe effects are easier to manage. Simulation is mainly seen as a tool to validate temporal properties of *models* (Edwards et al. 1997). We like to widen the usage of simulation to embrace validation of the final software product as well, i.e. to use simulation throughout the development process. Execution monitoring, profiling and tracing, as provided by the simulation system, can be employed to assess the performance of the agent software. The simulated environment facilitates the testing of agent’s behavior when confronted with extreme inputs and put under real-time pressure.

The simulation application itself is a software product and as such must be tested, i.e. verified and validated, in order to install confidence into the executed tests. Model verification implies that model transformation occurs with a sufficient accuracy. To verify the simulation it is required to analyze whether correct trajectories according to the model description are produced. The model verification does not really apply to our case. However, the simulation verification is important to assure that the simulation is not only efficient but also produces reliable results. Ongoing work is dedicated to analyze the repeatability of simulation runs and the role of the synchronous and asynchronous interaction of simulation and external software in more detail. The design of simulators as components and based on components requires a careful analyzing of the components, in isolation and once integrated, and to determine the context for which the developed components are truly exchangeable (Weyuker 1998).

To develop valid models, e.g. of the surroundings the agent AUTOMINDER is supposed to dwell in, poses serious difficulties. To validate the elderly model component data about the interaction of elderlies and nursebots are required. As the testing of basic functionalities of the nursebot robot in interaction with elderlies has just started (Montemerlo et al. 2002), a true validation of the elderly models in daily life seems out of reach. The best we can do is developing a set of prototypical and plausible model components that mimic the behavior of elderlies and a couple of bizarre ones to test the behavior of AUTOMINDER in borderline cases. Based on these models the simulation generates test cases dynamically, taking the activities of AUTOMINDER into account. Given the complexity of possible environment and AUTOMINDER interactions an exhaustive testing is not possible. To trace the execution of AUTOMINDER under a wider range of circumstances, AUTOMINDER can be tested in a virtual world whose model components include stochastic aspects. For this type of execution monitoring over many simulation runs the unpaced simulation will likely prove more suitable and the most interesting traces might be replayed by using the paced simulator. Thus, the temporal development of boundary cases can be analyzed in detail based on the paced simulator, whereas the unpaced simulation helps to identify these cases. Different simulators provide different approaches towards testing software, which eases the general experimentation, helps to reveal flaws within the tested system as well as in the used environmental models and simulators, thereby increasing the confidence into the used models and testing via simulation in particular.

As with all testing it is important that different groups are developing the software and the test scenarios. Independent verification and validation is a tech-

nique of long standing in the field of software engineering (IEEE 1998, p.58). The simulation software, particularly its usability, is evaluated by the AUTOMINDER research group. For this assessment, besides the offered functionalities, the user interface which is currently under development will play a crucial role. A close interaction with AUTOMINDER research group is currently directing the development of our models. However, as soon as the models have left the primordial ooze another group of the Nursebot project which is working closely with the elderlies shall help to evaluate the developed models. The more the project progresses the more the different research groups will be able to serve as a kind of software quality assurance group for each others software which answers part of the question “quis custodiet ipsos custodes” (who is guarding the guards).

## 7 CONCLUSION

The unpaced conservative simulator and the paced simulator are aimed at testing multi-agent systems containing a small number of deliberative, resource intensive agents. However, both simulators offer a different degree of control. The unpaced simulator invokes methods of the external software, responses of which are fed back into the simulation in simulation time. Our experiences with earlier agent projects indicated that as agents move from specification to implementation, different types of simulators are required. In the current project of coupling JAMES and AUTOMINDER we will explore this relation in detail. In this context a paced simulator is being developed, which advances simulation time in synchrony with wall clock time and supports a more realistic view on the virtual world. Agents are only loosely coupled to the simulator which facilitates a plug and play for the agent programmer. Whereas for earlier stages of developing agents an unpaced simulator seems appropriate for a first test of design decisions, later stages of implementation benefit from the easy plug and play and the more realistic view on the simulated environment the paced simulator provides in combination with an asynchronous exchange of messages. However, we expect that at some point in the designing process of agents performance issues of the complete software shall be analyzed again, and thus the unpaced simulator might come in handy.

## ACKNOWLEDGMENTS

This research is supported by the DFG (German Research Foundation). We would also like to thank the anonymous reviewer who motivated us to include the discussion section.

## REFERENCES

- Anderson, S. (1997). Simulation of Multiple Time-Pressured Agents. In *Proc. of the Wintersimulation Conference, WSC'97*, Atlanta.
- Balci, O., R. E. Nance, J. D. Arthur, and W. F. Ormsby (2002). Expanding our horizons in vv&a research and practice. In *Proceedings of the 2002 Winter Simulation Conference (San Diego, CA, Dec. 8-11)*, pp. 653–663. IEEE, Piscataway, NJ.
- Cho, Y., B. Zeigler, H. J. Cho, H. S. Sarjoughian, and S. Sen (2000). Design considerations for distributed real-time DEVS. In *Proceedings of Artificial Intelligence and Simulation*.
- Cohen, P. R., M. L. Greenberg, D. M. Hart, and A. E. Howe (1989). Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments. *AI Magazine* 10(3), 32–48.
- Dam, K. H. and M. Winikoff (2003). Comparing agent-oriented methodologies. In *Proceedings of the Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems*, Melbourne. to appear.
- Durfee, E. H. (1988). *Coordination of Distributed Problem Solvers*. Boston: Kluwer Academic Publishers.
- Edwards, S., L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli (1997). Design of embedded systems: Formal models, validation, and synthesis. *Proc. of the IEEE* 85(3).
- et al, M. P. (2002). Pearl: Mobile robotic assistant for the elderly. In *AAAI Workshop on Automation as Elder-care, August 2002*.
- Fujimoto, R. (2000). *Parallel and Distributed Simulation Systems*. John Wiley and Sons.
- Greenberg, M. and D. Westbrook (1990). The phoenix testbed. Technical Report UM-CS-1990-019, Computer and Information Science, University of Massachusetts at Amherst.
- Hanks, S., M. E. Pollack, and P. R. Cohen (1993). Benchmarks, Test Beds, Controlled Experimentation and the Design of Agent Architectures. *AAAI (Winter)*, 17–42.
- Hu, X. and B. Zeigler (2002). An integrated modeling and simulation methodology for intelligent systems design and testing. In E. Messina and A. Meystel (Eds.), *Proc. of PERMIS, Gaithersburg*. Proceedings of the 2002 PerMIS Workshop.
- IEEE (1998). IEEE standards for software verification and validation. IEEE Standard 1012. Washington, DC.
- Jennings, N. and M. Wooldridge (1998). Applications of Intelligent Agents. In N. Jennings and M. Wooldridge (Eds.), *Agent Technology: Foundations, Applications, and Markets*. Springer.
- Jennings, N. R., K. Sycara, and M. Wooldridge (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems* 1(1), 275–306.
- Kinny, D., M. Georgeff, and A. Rao (1996). A Methodology and Modelling Technique for Systems of BDI Agents. In W. Van de Velde and J. Perram (Eds.), *Agents Breaking Away*, Volume 1038 of *LNAI*, pp. 56–71. Springer.
- Kitano, H., S. Tadokor, H. Noda, I. Matsubara, T. Takhasi, A. Shinjou, and S. Shimada (1999). Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In *Proc. of the IEEE Conference on Systems, Men, and Cybernetics*.
- Kitano, H., M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada (1997). The RoboCup Synthetic Agent Challenge 1997. In *International Joint Conference on Artificial Intelligence IJCAI'97*.
- McCarthy, C. and M. Pollack (2002). A plan-based personalized cognitive orthotic. In *AIPS-2002*.
- McLean, T. and R. Fujimoto (2000). Repeatability in real-time distributed simulation executions. In *14th Workshop on Parallel and Distributed Simulation (PADS 2000)*, pp. 23–32.
- Montemerlo, M., J. Pineau, N. Roy, S. Thrun, and V. Verma (2002). Experiences with a mobile robotic guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada. AAAI.
- Montgomery, T. and E. Durfee (1990). Using MICE to Study Intelligent Dynamic Coordination. In *Second International Conference on Tools for Artificial Intelligence*, Washington, DC, pp. 438–444. Institute of Electrical and Electronics Engineers.
- Peraire, C., S. Barbey, and D. Buchs (1998). Test selection for object-oriented software based on formal specifications. In *PROCOMET*, pp. 385–403.
- Pollack, M. (1996). Planning in Dynamic Environments: The DIPART System. In A. Tate (Ed.), *Advanced Planning Technology*. AAAI.
- Pollack, M., L. Brown, D. Colbry, C. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos (2003). Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*.
- Pollack, M., C. McCarthy, S. Ramakrishnan, I. Tsamardinos, L. Brown, S. Carrion, D. Colbry, C. Orosz, and B. Peintner (2002). Autominder: A planning, monitoring, and reminding assistive agent. In *Seventh International Conference on Intelligent Autonomous Systems*.
- Pollack, M. E. and M. Ringuette (1990). Introducing the Tileworld: Experimentally Evaluating Agent Architectures. In *AAAI-90*, Boston, MA, pp. 183–189.
- Riley, P. and G. Riley (to appear). SPADES, a distributed agent simulation environment with software-in-the-loop execution. In *Winter Simulation Conference 2003*.
- Sargent, R. G. (1999). Validation and verification of simulation models. In *Winter Simulation Conference*, pp. 104–114.
- Schattenberg, B. and A. Uhrmacher (2001). Planning Agents in James. *Proceedings of the IEEE* 89(2), 158–173.
- Schütz, W. (1993). *The testability of distributed real-time*

*systems*. Kluwer Academic Publishers, Boston / Dordrecht / London.

Uhrmacher, A., P. Fishwick, and B. Zeigler (Eds.) (2001). *Special Issue: Agents and Simulation: Exploiting the Metaphor*, Volume 89 of *Proceedings of the IEEE*.

Uhrmacher, A. and K. Gugler (2000). Distributed, Parallel Simulation of Multiple, Deliberative Agents. In *Parallel and Distributed Simulation Conference PADS'2000*, Bologna. IEEE Computer Society Press.

Uhrmacher, A. and M. Kraemer (2001). A Conservative, Distributed Approach to Simulating Multi-Agent Systems. In E. Kerckhoffs and M. Snorek (Eds.), *Proc. European Multi-Simulation Conference*, San Diego, pp. 257–264. SCS.

Uhrmacher, A., M. Röhl, and B. Kullick (2002). The role of reflection in simulating and testing agents: An exploration based on the simulation system James. *Applied Artificial Intelligence* 16(9-10), 795–811.

Weyuker, E. J. (1998, September/October). Testing component-based software: A cautionary tale. *IEEE Software* 15(5), 54–59.

Wooldridge, M. and P. Ciancarini (2000). Agent-oriented software engineering: The state of the art. In P. Ciancarini and M. J. Wooldridge (Eds.), *First Int. Workshop on Agent-Oriented Software Engineering*, Volume 1957 of *Lecture Notes in Computer Science*, pp. 1–28. Springer-Verlag, Berlin.

Zeigler, B., H. Praehofer, and T. Kim (2000). *Theory of Modeling and Simulation*. London: Academic Press.

agent-oriented modeling and simulation and their applications. Web pages of authors can be found at: [www.informatik.uni-rostock.de/mosi](http://www.informatik.uni-rostock.de/mosi)

## AUTHOR BIOGRAPHIES

**JAN HIMMELSPACH** holds an MSc in Computer Science from the University of Koblenz. His research interests are on developing methods for agent-oriented modeling and simulation, with a focus on effective and efficient simulation mechanisms and interaction patterns between simulation and software agents. He is currently a research scientist at the Modeling and Simulation Group at the University of Rostock.

**MATHIAS RÖHL** holds an MSc in Computer Science from the University of Rostock. His research interests are on developing methods for agent-oriented modeling and simulation and their application to sociological, biological and software systems. He is currently a research scientist at the Modeling and Simulation Group at the University of Rostock.

**ADELINDE M. UHRMACHER** is an Associate Professor at the Department of Computer Science at the University of Rostock and head of the Modeling and Simulation Group. Her research interests are in modeling and simulation methodologies, particularly

# A PRACTICAL EFFICIENCY CRITERION FOR THE NULL MESSAGE ALGORITHM

András Varga<sup>†</sup> Y. Ahmet Şekercioğlu<sup>‡</sup> Gregory K. Egan<sup>‡</sup>

<sup>†</sup>Omnest Global Inc., Budapest, Hungary

<sup>‡</sup>Centre for Telecommunication and Information Engineering, Monash University, Melbourne, Australia

## KEYWORDS

Parallel simulation, discrete-event simulation, PDES, cluster computing

## ABSTRACT

This paper presents a quantitative criterion for efficient execution of the Null Message Protocol, the best-known conservative parallel discrete event simulation (PDES) protocol. By using the criterion, a model designer can use lookahead and communication latency as input and improve the efficiency of parallelization. Earlier works consider lookahead in relation to model properties like timestamp increment and in isolation from the capabilities of the underlying hardware/software simulation environment, and have not been able to provide quantitative criteria for performance prediction.

Our results suggest that the performance impact of lookahead can only be quantified when linked to other performance factors such as communication latency and throughput between partitions of a parallelized simulation model. The latency and throughput issues are becoming of increasing importance as clusters gain popularity as PDES platforms.

The criterion is based on a novel concept of the coupling factor, and allows one to use intuitive and easy-to-measure input parameters. The criterion can be used to assess simulation models' potential for parallel execution as well as the maximum partitioning that may still potentially yield good performance. This paper is also novel in that it uses the Ideal Simulation Protocol as a benchmark.

## INTRODUCTION

Telecommunication networks are increasingly becoming more complex as the trend towards integration of telephony and data networks into integrated services networks gains momentum. Discrete event simulation is an important tool for the research and design of these systems. However, simulation of telecommunications networks is generally a computationally intensive task. A single run of a wireless network model with thousands of mobile nodes may easily

take several days or even weeks to obtain statistically significant results even on today's computers. Additionally, many simulation studies require several simulation runs to improve statistical reliability of the outcomes (Bagrodia et al., 1998). These conditions have recently led to an increased interest in Parallel Discrete Event Simulation (PDES) techniques. Good overviews of PDES techniques can be found in (Nicol and Fujimoto, 1994) and (Soliman et al., 1995).

Although recent PDES literature focuses more on optimistic algorithms than conservative ones, optimistic algorithms are usually too complex or impractical to implement in practice. At the same time, we can observe a revival of conservative algorithms as they slowly seem to find their way into simulation tools. This is especially true for telecommunication network simulations where the demand for processing power is the strongest: SSFNet (ssfnet), ns2 (PADS Research Group), OMNeT++ (Şekercioğlu et al., 2003). Another trend is that clusters (as opposed to shared memory multiprocessor systems) are becoming an attractive PDES platform (Pham, 1999), mainly because of their excellent price/performance ratio. The best known conservative algorithm is still the classic Chandy-Misra-Bryant (CMB), also known as the null message algorithm (NMA) (Chandy and Misra, 1979) (Bagrodia and Takai, 2000); we will use the latter name throughout the paper.

Despite the intensive research efforts on lookahead and its effect on the performance of the parallel simulation, we still do not have quantitative criteria for predicting the performance of a parallel simulation. This is especially true in a cluster computing environment where finite bandwidth and relatively large communication latencies are present. Also, Ideal Simulation Protocol (ISP) (Bagrodia and Takai, 2000), a significant and powerful tool for the research of performance aspects of PDES algorithms was only discovered in 2000, and since then it has gone relatively unnoticed within the PDES research community.

The first section of this paper describes performance factors of a parallel simulation. The second part focuses on the classic NMA algorithm and derives the criterion that can predict the performance of NMA. The criterion allows the simulation designer to use input data that are easy to produce for any given simulation, and it can also be applied in a cluster environment because it takes into account the communication latency. The third part describes simulation

experiments done on a cluster to validate the performance criterion. Here, we use the ISP as a performance benchmark.

## PERFORMANCE FACTORS

### Lookahead

Lookahead has crucial importance in conservative algorithms. Lookahead is associated with the ability of a logical process (LP), in other words, a partition of a model, to predict its future behavior. It has many, slightly different definitions in the literature (Preiss and Loucks, 1990); for this paper we settle for the following: *At any simulation time  $T$ , if an LP can predict that the earliest event it will cause to occur in another LP is no sooner than  $T + L$ , it has a lookahead of  $L$  towards that LP.* Lookahead may be different for each (ordered) LP pair; moreover, lookahead is, in general, state dependent and may even change during the execution of the lookahead period.

For specific classes of models, different model components can lead to lookahead. In queueing simulations for example, the fixed processing time of a queue server may provide lookahead; so can a minimum inter-arrival time in a source. In telecommunication networks, propagation delay on links (or propagation delay plus the transmission time of a minimum-length frame) serves as a natural source of lookahead. Also, observing the full picture might provide better lookahead than individual components in the model – simulation techniques like *path lookahead* (Meyer and Bagrodia, 1999) exploit this observation.

It has been observed and is common knowledge that lookahead has a significant effect on the performance (Fujimoto, 1989). Lookahead and its impact on the performance of parallel simulations has been analyzed in (Lin and Lazowska, 1990), (Preiss and Loucks, 1990), (Wand and Abrams, 1995) and other papers, and yet the simple question of *when* lookahead is large enough to provide reasonable performance has remained unanswered.

Earlier works consider lookahead in relation to model properties like timestamp increment, and in isolation from the capabilities of the underlying hardware/software simulation environment. Our results suggest that the performance impact of lookahead can only be quantified when linked to other performance factors such as communication latency and throughput between LPs.

### Latency and throughput

Early PDES experiments on clusters have delivered disappointing results compared to shared memory multiprocessor systems (Pham, 1999). This fact is attributable to much higher communication costs on clusters, namely (a) high processing overhead associated with sending and receiving messages, (b) finite bandwidth of the communication channel, and (c) higher latencies.

Several studies have been conducted on the performance of parallel simulations on clusters, e.g. (Lemeire and Dirx,

2001) and (Xu and Chung, 2001). Most of them focus on aspects (a) and (b) and tend to ignore (c). In particular, no studies have linked the question of communication overheads with proper lookahead in an attempt to predict the performance of parallel simulation algorithms.

### The Ideal Simulation Protocol

One cannot expect linear speedup from PDES because, compared to sequential execution, some parts of the model are now separated by LP boundaries, and transmission of model messages across LPs presents an overhead that was not present in the sequential model. Since these messages are part of the model, we can do little to reduce this overhead.

In addition to this *messaging overhead*, we also have *synchronization overhead*, an overhead added by the parallel simulation (synchronization) algorithm. With NMA, this overhead is associated with the transmission of null messages and with time spent blocking on *earliest input times (EITs)* (an NMA overview, including the definitions of terms is provided in the next section. Synchronization overhead may be reduced by tuning the parameters of the PDES algorithm or choosing a different algorithm.

When evaluating the efficiency of a parallel simulation algorithm, it would be useful to know what is the *maximum achievable speedup*, that is, the speedup if synchronization overhead were zero. Until recently, researchers have not been able to directly measure the maximum achievable speedup, and hence, PDES studies have been published without this comparison. The Ideal Simulation Protocol (ISP) introduced by Bagrodia (Bagrodia and Takai, 2000) can provide this missing information.

ISP is not an abstraction as it may sound, but an actual parallel simulation algorithm that can be implemented and models can be run under it. Running a model under ISP does not incur any synchronization overhead (in fact, there *is* some overhead, but it is usually negligible), while all other overheads including messaging overhead remain unchanged. Therefore, ISP presents the upper limit on the performance that any parallel simulation algorithm can achieve under the same circumstances (executing the same model with the same partitioning, on the same hardware, using the same operating system and simulator, with same message transport between LPs, etc.).

When evaluating the efficiency of various PDES algorithms, their performance ratios in relation to ISP are far more useful and relevant numbers than the speedup figures. Comparison to ISP tells us how far we are from the best achievable parallel performance, while speedup also contains the messaging overhead, a factor that cannot be blamed on the synchronization algorithm itself.

## PERFORMANCE OF NULL MESSAGE ALGORITHM

### The Null Message Algorithm

For the discussion of the Null Message Algorithm (NMA), we use the terminology introduced by Bagrodia in (Bagrodia and Takai, 2000). According to NMA, LPs maintain variables called *earliest input times (EITs)* for each input neighbor, and *earliest output times (EOTs)* for each output neighbor LP. An EIT stores the earliest simulation time the LP may receive an event from another LP. Respectively, an EOT stores the earliest simulation time the LP might send a message to its neighbor. Practically, EOT represents the local simulation time plus the lookahead.

LPs are safe to process events until the minimum of their EITs (“safe” meaning without the danger of receiving events in their past). Once an EIT has been reached by a LP, it has to block until the EIT is updated. EITs are updated by *null messages* arriving from other LPs. A null message carries a timestamp, set by the sender LP to its current EOT. Once the null message arrives, the timestamp will be stored by the receiving LP as a new EIT. For obvious reasons, EITs grow monotonically.

LPs must send out null messages often enough to prevent deadlock, i.e. at least before the expiry of the EOT sent out in the in the last null message. In the hope of improving performance, LPs may actually choose to send out null messages more often than necessary, leading to the designation of *eager* and *lazy* algorithms. Laziness is a tunable parameter of NMA. For optimization, null messages may be piggybacked on normal outgoing messages. It has been proven that the nonexistence of zero lookahead cycles in the graph is enough to prevent deadlocks.

### Performance of NMA

When the NMA performs poorly compared to ISP (note that NMA can only approximate ISP performance – if parallel performance under ISP is already poor (e.g. because of too much cross-partition messaging), it will inevitably be poor under NMA, too), causes can be traced back to one of the following two reasons:

- a. *Too frequent null messages.* If lookahead is poor compared to the simulation time between events, that leads to excessive null message traffic. Resources are consumed by sending, waiting for and receiving null messages, instead of processing events.
- b. *Too much time spent on blocking on EITs.* In this case, processors idle too much, waiting for null messages to arrive. This can be caused by too tight coupling of LPs, as a consequence of too little workload on LPs, combined with poor lookahead and/or long communication latencies and/or poor load balancing.

Although not evidenced by measurements, it is strongly felt that additional factors (such as overhead of null message

sending in case of many LPs) may only be significant if (a) and (b) are solved. In the following sections we will examine the above two factors and provide quantitative criteria for them. We will use the following parameters as input:

- *P performance* represents the number of events processed per second (ev/sec) (we use the following notation: *ev*: events, *sec*: real seconds, *simsec*: simulated seconds). *P* depends on the performance of the hardware and the amount of computation required for processing an event. *P* is independent of the size of the model. On the authors’ computer, simulations using OMNeT++ (Varga, 2001) usually yield *P* values between 20,000 and 120,000 ev/sec, depending on the nature of the model.
- *E event density* is the number of events that occur per simulated second (ev/simsec). *E* depends on the model only, and not where the model is executed. *E* is determined by the size, the detail level and also the nature of the simulated system (e.g. cell-level ATM models produce higher *E* values than call center simulations.)
- *R relative speed* measures the simulation time advancement per second (simsec/sec). *R* strongly depends on both the model and on the software/hardware environment where the model executes. Note that  $R = P/E$ .
- *L lookahead* is measured in simulated seconds (simsec). When simulating telecommunication networks and using link delays as lookahead, *L* is typically in the msimsec- $\mu$ simsec range.
- $\tau$  *latency* (sec) characterizes the parallel simulation hardware.  $\tau$  is the latency of sending a message from one LP to another.  $\tau$  can be determined using simple benchmark programs. The authors’ measurements on a Linux cluster interconnected via a 100Mb Ethernet switch using MPI yielded  $\tau = 22\mu\text{s}$  which conforms to the measurements reported in (Ong and Farrell, 2000); specialized hardware such as Quadrics Interconnect (quadrics) can provide  $\tau = 5\mu\text{s}$  or better.

In large simulation models, *P*, *E* and *R* usually stay relatively constant (that is, display little fluctuations in time). They are also intuitive and easy to measure. For example, the OMNeT++ simulation tool displays these values on the GUI while the simulation is running, see Figure 1.

### Too Frequent Null Messages

If lookahead is too small compared to the mean simulation time between events, several rounds of null messages will be needed to advance simulation time over otherwise event-free time periods. As an illustration, consider a queueing network model executed in parallel. Let the processing time of the queue servers be 0.1 seconds. We use the processing time as lookahead, so EIT is increased in 0.1s steps. If jobs arrive at the queues only every 3 seconds, then 30 rounds

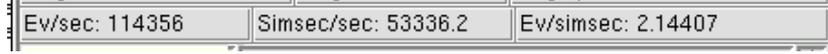


Figure 1: Performance bar in OMNeT++ showing  $P$ ,  $R$  and  $E$

of null messages are necessary to advance simulation time to the next event. This phenomenon is widely analyzed in the literature, see (Nicol and Fujimoto, 1994) for example. Each round of null messages takes at least  $\tau$  time (null messages have to arrive), which can be painful especially on cluster hardware.

In order to achieve good performance, the lookahead needs to be significantly larger than the time between events:  $L \gg 1/E$ . Rewriting this in a more convenient form:

$$LE \gg 1 \quad (1)$$

The actual threshold for  $LE$  depends on the relative cost of sending a null message and processing an event as well as  $\tau$  and  $P$ . It is also worth noting that the number of null messages sent out and thus the null message sending overhead grows as the number of output neighbor LPs grow.

Note that  $L$  and  $E$  are properties of the model, which means that some models will always perform poorly under NMA, regardless of the software/hardware environment. A remedy to this problem could be to use an alternative synchronization method, e.g. Conditional Event Algorithm (which relies on calculating global virtual time), as suggested by Bagrodia in (Bagrodia and Takai, 2000).

$LE$  is inversely proportional to the *lookahead ratio* (Preiss and Loucks, 1990).

### Too Much Blocking on EITs

Null messages must reach a target LP early enough so that the target LP does not need to block on the EIT before the null message arrives. In other words, an LP has to have enough work (events to process) until the next null message arrives.

We use a simple simulated scenario to examine and quantify this criterion. We model two networks that are connected by a 1000km fiber optic cable, with a propagation delay of about 5ms (Figure 2). The two networks are executed in separate LPs, using the Null Message Algorithm. We use the cable delay as lookahead, that is,  $L = 5\text{ms}$ . We run the simulation on a cluster computer that provides us  $\tau = 10\mu\text{s}$  latency. Also, assume the following ideal conditions: idle link (no modeled traffic), lazy null message sending, and virtual times proceed in sync in the two LPs. Then, LPs will periodically exchange null messages with each other (every  $L = 5\text{ms}$  simulation time; see Figure 3). Null messages sent out from one LP take  $\tau$  (real) time to arrive. They have to arrive at the other LP before that LP reaches the EIT received in previous null message, that is, in less time than it takes the target LP to advance  $L$  simulation time. This results in the following criterion:

$$\tau < \frac{L}{R} \quad (2)$$

Using  $R = P/E$  and rearranging the equation we get that

$$\tau P < LE \quad (3)$$

That is, if (3) holds and the described ideal conditions are present, LPs will never block on EITs. But what if conditions are different?

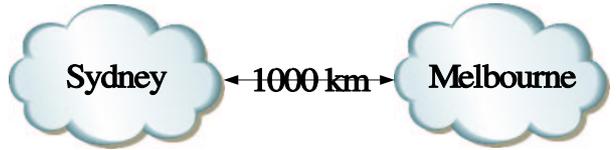


Figure 2: The simulated scenario

*If null message sending is not lazy*, there will be more frequent null messages, which further reduce the probability of blocking. Actually, more frequent null messages help “smooth out” fluctuations in latency and virtual time progression, at the cost of more messaging overhead.

*If there is traffic on the link*, it also means that more frequent null messages (piggybacked on simulated packets) are present, which reduces the possibility of blocking.

*What if simulation times are not in sync?* In real life, simulation time does not pass completely evenly ( $R$  relative speed fluctuates in both LPs), and this causes simulation times to be skewed. The NMA algorithm places an upper bound equal to the lookahead on the skew, because LPs have to block whenever an EIT has been reached. We will show that in order to reduce the amount of blocking, the inequality (3) has to hold “stronger” (that is,  $\tau P \ll LE$ ).

*If there are more partitions*, inequality (3) should hold for all ordered LP pairs. This is covered in greater detail in the section entitled “Applying To Several LPs”.

Inequality (3) displays the following property: the left hand side,  $\tau P$ , depends mainly on properties of the hardware (and much less on the model) and the right hand side,  $LE$  only depends on the model (and not at all on the hardware).  $\tau P$  expresses how many events are processed during  $\tau$  time. Its value characterizes the hardware: a small value indicates fast communication compared to processing speed, in other words, shared-memory-type hardware, while large values indicate strong processing power and slower communication, that is, cluster-type hardware. The actual boundary seems to be around  $\tau P = 1$ .  $LE$  shows the model’s potential for efficient PDES under NMA. It expresses “how many events the lookahead cover”. Small values mean poor lookahead, and large values mean good lookahead. The larger the value of  $LE$ , the more potential the model has to perform well on cluster-type hardware.

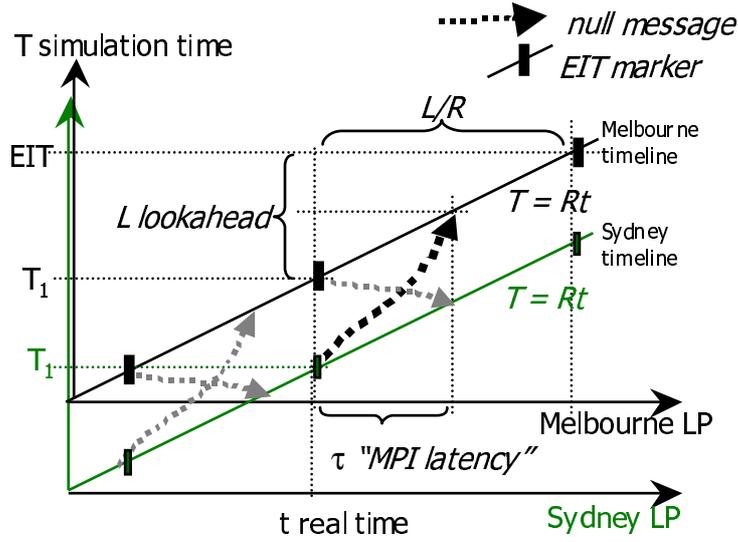


Figure 3: Periodic exchange of null messages in the simulated scenario

$\tau P < LE$  only guarantees that there's no blocking if simulation time passes completely evenly, that is,  $R = P/E$  relative speed is constant. In practice, there are fluctuations in  $P$  and  $E$ , so  $\tau P = LE$  will lead to frequent blocking because the LPs are too tightly coupled.  $\tau P < LE$  allows for fluctuations in  $P$  and  $E$  to occur, reducing the chance of blocking on EITs. Let us introduce  $\lambda$  *coupling factor*, the ratio of  $LE$  and  $\tau P$ :

$$\lambda = \frac{LE}{\tau P} \quad (4)$$

It follows from its definition that if  $\lambda < 1$ , frequent blocking on EITs is guaranteed and one cannot expect good performance from the simulation; if  $\lambda \geq 1$  but small, more or less blocking will occur depending on how much  $P$  and  $E$  fluctuate, and as experimental results show, this heavily affects performance. If  $\lambda > 1$  and large enough, then blocking on EITs is no longer a major performance factor. Our experimental results indicate that  $\lambda$  values below 10 should be regarded as “small” and values above 100 as “large”, the exact performance characteristics being dependent on the simulation model.

The question of “when is lookahead big enough” can then be answered: when  $\lambda$  is greater than a  $\lambda_0$  threshold chosen in the range 10 ... 100, that is:

$$L > \frac{\lambda_0 \tau P}{E} \quad (5)$$

One can think of several intuitive interpretations of  $\lambda$ : “how many times  $\tau$  it takes to progress  $L$  simulation time,” or “how big is lookahead  $L$ , in units of simulation time that can be covered during  $\tau$  time”.

Another interpretation can be derived from the skew of the simulation times in the LPs. As one can deduce from Figure 3, the simulation times can be skewed at most  $L - R\tau$

if we want to avoid blocking. (Intuitively: if the null message took zero time to arrive ( $\tau = 0$ ), simulation times could be skewed by  $L$  amount, but this is decreased by the simulation time that is covered during  $\tau$  time.) If expressed in terms of  $\lambda$ , the maximum skew is  $R\tau(\lambda - 1)$ . This explains why  $\lambda$  is called coupling factor:  $\lambda = 1$  means tight coupling (because it forces zero skew and simulation times to pass in sync in the two LPs, possibly at the cost of frequent blocking), and a large  $\lambda$  means loose coupling (allows for more skew and therefore more fluctuations in  $P$  and  $E$ , without heavily affecting performance).

### Applying To Several LPs

We need to apply the criteria for all (ordered) pairs of LPs, because the slowest LP pair determines simulation speed. Note that when virtual time progresses slowly in an LP, it also “pulls back” other LPs. Let  $E_i$  and  $P_i$  be the event density and performance in LP<sub>*i*</sub>, and let  $L_{i,j}$  and  $\tau_{i,j}$  be the lookahead and latency *toward* LP<sub>*i*</sub> from its LP<sub>*j*</sub> input neighbor (Figure 4). Let us define  $\lambda_{i,j}$  as follows:

$$\lambda_{i,j} = \frac{L_{i,j} E_i}{\tau_{i,j} P_i} \quad (6)$$

To achieve good performance, all  $L_{i,j} E_i$  (1) and  $\lambda_{i,j}$  (4) values have to be sufficiently high, thus we can define an *effective coupling factor* as

$$\lambda = \min_{i,j} \lambda_{i,j} \quad (7)$$

### NMA Scalability and Partitioning

How does NMA scale with the number of processors used? We can expect  $P_i$  and  $\tau_{i,j}$  values to stay relatively constant as

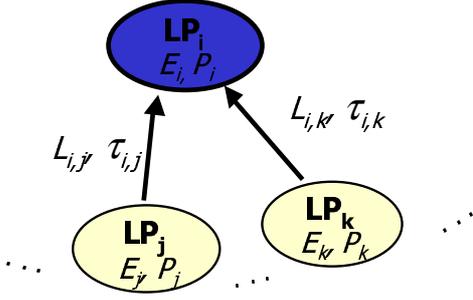


Figure 4: Multiple LPs

the number of LPs grows ( $n$ ).  $L_{i,j}$  lookahead might or might not decrease as  $n$  grows, depending on the model topology.  $E_i$ , however, will decrease. Events do not change as we partition the model, so the event densities,  $E_i$ , have to sum up to  $E_{seq}$ , the event density of the sequential execution. Thus, for the event density in  $LP_i$  with  $n$  LPs,  $E_{n,i}$ :

$$\min_i E_{n,i} \leq \frac{E_{seq}}{n} \quad (8)$$

Consequently, the effective coupling factor for  $n$  LPs,  $\lambda_n$ , will also diminish as  $n$  grows:

$$\lambda_n \leq \frac{\lambda_0}{n} \quad (9)$$

where  $\lambda_0$  can be derived from the two-processor simulation as  $\lambda_0 = 2\lambda_{n=2}$ . As  $\lambda_n$  falls below a critical value, performance degrades. The intuitive explanation is that with heavy partitioning, some processors might not get enough work to do between receiving null messages and thus they are often forced to block on EITs.

A practical use of (9) is to assess the available potential for parallelism in the model, that is, the maximum number of LPs where NMA can still be expected to produce good results.

Partitioning algorithms aware of coupling factor  $\lambda$  should consider only partitions where all  $\lambda_{i,j}$  values are above a threshold.

## EXPERIMENTAL VERIFICATION OF THE CRITERION

Experiments have been performed to verify the criterion. We have used the closed queueing network (CQN) model described in (Bagrodia and Takai, 2000) (Figure 5). The model consists of  $N$  tandem queues where each tandem consists of a switch and  $k$  single-server queues with exponential service times. The output of the last queue is looped back to the switch. Each switch randomly chooses the first queue of one of the tandems as its destination, using uniform distribution. The queues and switches are connected with links that have nonzero propagation delays. At the beginning of the simulation, a fixed number of jobs are injected in the system; no jobs are created or destroyed during simulation.

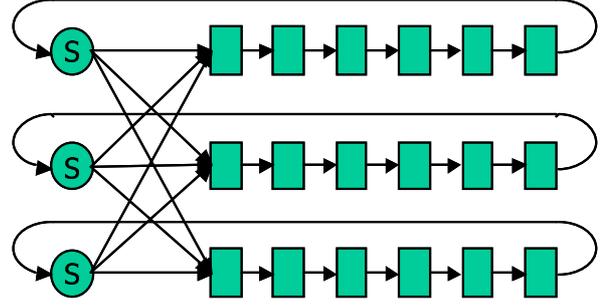


Figure 5: The Closed Queueing Network (CQN) model, which consists of  $N$  tandem queues where each tandem consists of a switch and  $k$  single-server queues

We used a model consisting of  $N = 16$  tandems. We performed experiments with 2-way and 4-way partitioning (that is, with 8 and 4 tandem queues per LP). The propagation delay between switches and first queues was used as the lookahead  $L$ , and we conducted a series of experiments with various lookahead values. The other variable parameter in the model is  $k$ , the number of queues per tandem, which we used to control the event density,  $E$ . Initially we inserted 2 jobs in each queue, propagation delays between queues were set to 1, and we used an exponential distribution with a mean of 10 as the queue service time.

The simulation environment was an in-house development version of OMNeT++ framework (Varga, 2001) (Şekercioğlu et al., 2003). The hardware environment was a Linux cluster (kernel 2.4.9) of dual 1 Ghz Pentium III PCs, interconnected using a 100Mb Ethernet switch. The communication library was LAM-MPI (lam-mpi). The MPI latency  $\tau$  was measured to be  $22\mu s$ . Sequential simulation of the CQN model achieved  $P_{seq} = 120,000$  ev/sec performance, a value that was independent of  $k$ .

We performed the simulation under NMA and (for comparison) under ISP. We recorded the average per-LP  $P_i$  performances, and also the  $E_i$  event densities so that we could calculate  $\lambda$ . We performed experiments with the following  $k = 1, 2, 5, 10, 20, 50, 100, 200$  and  $L = 1, 2, 5, 10, 20, 50, 100, 500$  values on 2 and 4 processors.

An excerpt of the results is shown in Table 1.  $S_{ISP}$ ,  $S_{NMA}$  are the speedups achieved under ISP and NMA, respectively (measured as  $P/P_{seq}$ );  $NMA/ISP$  is the ratio of the NMA and ISP speedups. Graph 6 shows  $NMA/ISP$  versus  $\lambda$ .  $\lambda > 50$  values all yielded  $NMA/ISP > 0.78$  and are not shown in the graph.  $\lambda > 200$  yielded  $NMA/ISP > 0.95$ . We should note that  $NMA/ISP$  values slightly bigger than 1 have been observed for very large  $\lambda$  values.

The performance shows strong correlation to  $\lambda$ . Small  $\lambda$  values (under 5 or so) result in poor performance, and large values (over 100) result in a performance near ISP (values larger than 100 do not have significant impact any more). One can observe that the threshold for  $\lambda$  is somewhere in the range  $10 \dots 50$ .

LPs	$k$	$L$	$E_i$	$\lambda$	$S_{ISP}$	$S_{NMA}$	$NMA/ISP$
2	5	1	5.81	2.20	0.94	0.41	0.43
2	5	2	5.80	4.40	0.99	0.62	0.62
2	5	5	5.76	10.91	1.01	0.92	0.92
2	20	1	21.66	8.20	1.34	0.84	0.63
2	20	2	21.62	16.38	1.40	1.10	0.79
2	20	5	21.59	40.90	1.41	1.33	0.94
4	5	1	2.90	1.10	2.03	0.21	0.10
4	5	5	2.88	5.46	2.01	0.77	0.38
4	5	20	2.78	21.08	2.12	1.67	0.78
4	5	50	2.57	48.72	2.16	2.08	0.97
4	20	1	10.83	4.10	2.94	0.68	0.23
4	20	2	10.81	8.19	2.90	1.10	0.38
4	20	5	10.80	20.46	2.92	1.84	0.63
4	20	10	10.77	40.79	2.99	2.35	0.79

Table 1: Experimental results on 2- and 4-processor configurations with various  $k$  and  $L$  values

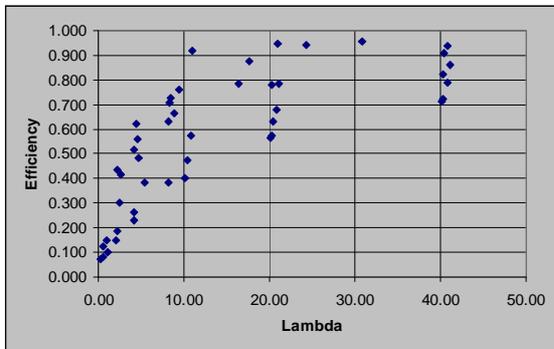


Figure 6:  $NMA/ISP$  performance ratio vs  $\lambda$

( $\tau = 22\mu s$ ). Experiments show that IP simulations produce around  $P = 80,000$  ev/sec on a single CPU. We use link delays as lookahead, typically around  $L = 1$ ms. What is the smallest network that has a chance to produce good speedup on  $n = 4$  CPUs (we would like at least  $\lambda = 20$ )?

Using Equation (4) and inequality (8)

$$E_{seq} \geq \frac{n\lambda\tau P}{L} = 140,800 \text{ ev/simsec}$$

Here,  $LE_{seq}/n = 35.20$ , thus inequality (1) also holds. Further simulation experiments can cast light on what event densities are expected to be generated for host, router models etc, under a given simulated network traffic. It is thus possible to deduce the size of the network.

## APPLYING THE CRITERION IN PRACTICE

The criterion can easily be applied in practice by the simulation designer to find out if a model has the potential for parallel execution with NMA. The  $P$  and  $E_{seq}$  values can be collected from a sequential run of the model, with the help of the simulation tool. For example, OMNeT++ displays  $P$  and  $E$  in both its command-line and GUI user interface.  $\tau$  can be measured by simple test programs, or its approximate value can be guessed from the hardware and software components (as mentioned earlier, a commonly available Linux cluster with MPI yielded  $\tau = 22\mu s$  in our measurements). Lookaheads,  $L$ , can be determined from the model itself.

With  $L$ ,  $E_{seq}$ ,  $\tau$  and  $P$  present, it is then possible to assess  $LE$  and  $\lambda$  for different numbers of LPs ( $n$ ), using the  $E = E_{seq}/n$  approximation. If satisfactory  $LE$  and  $\lambda$  values are present, the model could be deemed to have good potential for NMA.

An example: assume that we intend to run Internet simulations in parallel on the above mentioned Linux cluster

## CONCLUSION

We have derived a quantitative criterion to predict in which settings the null message algorithm (NMA) has a potential to perform well. The criterion accounts for the cases when the protocol would send too many null messages or it would block too frequently, and it is based on the newly introduced concept of the coupling factor. We have experimentally verified the criterion, using the performance ratio to the Ideal Simulation Protocol as a benchmark. Although the criterion was introduced in the context of network simulation and cluster computing, it is applicable to any simulation model and shared memory architectures as well.

The criterion provides a quick and practical way to predict whether a simulation model has potential to perform well under NMA in a given simulation environment, and may also help to determine the maximum degree of partitioning where the model can still be expected to produce good performance under NMA.

It is left to further studies to experimentally verify the criterion on different types of models and on a larger number of processors. Further studies are needed to analytically explain why the coupling factor has to be larger than

10...100. For this, probably stochastic tools and a quantification of fluctuations in performance  $P$  and event density  $E$  values will be needed.

## Acknowledgment

Authors would like to express their gratitude to Brett Pentland for his help on improving the text.

## References

- R. L. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H. Y. Song. Parsec: A parallel simulation environment for complex systems. *IEEE Computer*, pages 77–85, October 1998.
- R. L. Bagrodia and M. Takai. Performance evaluation of conservative algorithms in parallel simulation languages. *IEEE Transactions on Parallel and Distributed Systems*, 11(4):395–414, 2000. URL [citeseer.nj.nec.com/bagrodia98performance.html](http://citeseer.nj.nec.com/bagrodia98performance.html).
- M. Chandy and J. Misra. Distributed simulation: A case study in design and verification of distributed programs. *IEEE Transactions on Software Engineering SE-5*, (5): 440–452, 1979. URL <http://citeseer.nj.nec.com/context/58222/0>.
- Y. A. Şekercioğlu, A. Varga, and G. K. Egan. Parallel simulation made easy with OMNeT++. In *Proceedings of European Simulation Symposium (ESS2003)*, Delft, The Netherlands, October 2003. Society for Computer Simulation.
- R. M. Fujimoto. Performance measurements of distributed simulation strategies. *Trans. of the SCS*, 6(2):89–132, apr 1989.
- lam-mpi. LAM-MPI home page. URL <http://www.lam-mpi.org/>.
- J. Lemeire and E. Dirkx. Performance factors in parallel discrete event simulation. In *Proc. of the Int. Multiconference on Simulation and Modeling (ESM 2001)*, Prague, June, 2001. Society for Computer Simulation, 2001.
- Y.B. Lin and E.D. Lazowska. Exploiting lookahead in parallel simulation. *IEEE Transactions on Parallel and Distributed Systems*, (4):457–469, oct 1990. URL <http://citeseer.nj.nec.com/context/58222/0>.
- R. A. Meyer and R. Bagrodia. Path lookahead: a data flow view of PDES models. In *Proceedings of the 13th Workshop on Parallel and Distributed Simulation (PADS'99)*, pages 12-9, 1999, 1999.
- D. M. Nicol and R. M. Fujimoto. Parallel simulation today. *Annals of Operations Research*, (53):249–285, 1994. URL <http://citeseer.nj.nec.com/nicol94parallel.html>.
- H. Ong and P. A. Farrell. Performance comparison of LAM/MPI, MPICH and MVICH on a Linux cluster connected by a Gigabit Ethernet network. In *Proceedings of the 4th Annual Linux Showcase & Conference, Atlanta, October 10-14, 2000*. The USENIX Association, 2000.
- Atlanta PADS Research Group, Georgia Institute of Technology. PDNS - Parallel/Distributed NS home page. URL <http://www.cc.gatech.edu/computing/compass/pdns>.
- C. D. Pham. High performance clusters: A promising environment for parallel discrete event simulation. In *Proceedings of the PDPTA'99, June 28-July 1, 1999, Las Vegas, USA, 1999*.
- B. R. Preiss and W. M. Loucks. The impact of lookahead on the performance of conservative distributed simulation. In *Proc. 1990 European Multiconference-Simulation Methodologies, Languages and Architectures*, pages 204-209, Nuremberg, FRG, June 1990. Society for Computer Simulation, 1990.
- quadrics. Quadrics home page. URL <http://www.quadrics.com/>.
- H. M. Soliman, A. S. Elmaghraby, and M. A. El-Sharkawy. Parallel and distributed simulation: An overview. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'95)*, June 27-29, 1995, Alexandria, Egypt, 1995.
- ssfnet. SSFNet home page. URL <http://www.ssfnet.org>.
- A. Varga. The OMNeT++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM'2001)*. June 6-9, 2001. Prague, Czech Republic, 2001.
- J. J. Wand and M. Abrams. The impact of lookahead on conservative simulation. Technical Report [ncstrl.vatech\\_cs//TR-95-03](http://ncstrl.vatech_cs//TR-95-03), Computer Science, Virginia Polytechnic Institute and State University, 1995. URL <http://eprints.cs.vt.edu:8000/archive/00000418/>.
- J. Xu and M. J. Chung. Predicting the performance of synchronous discrete event simulation systems. In *Proceedings of ACM/IEEE International Conference on Computer Aided Design, San Jose 2001*, pp. 18-12., 2001.

## AUTHOR BIOGRAPHIES

**András Varga** received his M.Sc. in computer science with honors from the Technical University of Budapest, Hungary in 1994. He worked for several years as software architect for Encorus (formerly Brokat Technologies), which has provided distributed application server technologies for financial institutions in Europe and Asia, and now focusing on Internet and mobile payment solutions.

He is the author of the OMNeT++ open-source network simulation tool currently widely used in academic and industrial settings, and founder of Omnest Global, Inc. which provides commercial licenses and services for OMNeT++ worldwide. He is currently working towards PhD, his research topic being large-scale simulation of communication networks. Between February and September 2003 he visited CTIE at Monash University (Melbourne, Australia) to participate in the parallel simulation research project.

**Y. Ahmet Şekercioğlu** is a researcher at the Centre for Telecommunications and Information Engineering (CTIE) and a Senior Lecturer at Electrical and Computer Systems Engineering Department of Monash University, Melbourne, Australia. He also holds the position of Program Leader for the Applications Program of Australian Telecommunications Cooperative Research Centre (ATCrc, <http://www.atcrc.com>). He completed his PhD degree at Swinburne University of Technology, Melbourne, Australia (2000), MSc (1985) and BSc (1982) degrees at Middle East Technical University, Ankara, Turkey (all in Electrical Engineering). He has lectured at Swinburne University of Technology for 8 years, and has had numerous positions as a research engineer in private industry.

His recent work focuses on development of tools for simulation of large-scale telecommunication networks. He is also interested in application of intelligent control techniques for multiservice networks as complex, distributed systems.

His e-mail address is : [ASekerci@ieee.org](mailto:ASekerci@ieee.org) and his Web-page can be found at <http://titania.ctie.monash.edu.au>.

**Gregory K. Egan**'s principal research interests are the design, programming and the application of high-performance parallel distributed computer architectures.

He is currently Professor of Telecommunications and Information Engineering, Director of the Centre for Telecommunications and Information Engineering and Head of the Department of Electrical and Computer Systems Engineering at Monash University in Australia.



# **DISCRETE SIMULATION LANGUAGES AND TOOLS**



# RAMA : a lightweight rule-based tool for expressions analysis and code generation

Vincent Fischer, Loig Allain, Laurent Gerbaud

Laboratoire d'Electrotechnique de Grenoble, CNRS UMR 5529 INPG/UJF

ENSIEG BP 46, 38402 Saint Martin d'Hères, France

*vincent.fischer@leg.ensieg.inpg.fr, loig.allain@leg.ensieg.inpg.fr, laurent.gerbaud@leg.ensieg.inpg.fr*

**Abstract - This paper presents a lightweight tool for mathematical expressions analysis and code generation. This tool, called RAMA (Rule Applicator for Mathematical Analysis) is based on rules written in a XML format. In this way, it is generic and extensible and it can be used for various purposes. RAMA is based on a representation for mathematical expressions, on which rules are applied in order to perform some actions, e.g. : symbolic differentiation.**

**In a first part, the specifications of the software are presented. In the second part, its architecture and its operating are explained. Then, the representation model of mathematical expressions is presented. The software architecture is the detailed. Finally, some application cases are presented.**

## 1. INTRODUCTION

This paper presents RAMA, a lightweight tool written in JAVA, designed for the analysis and the treatment of mathematical expressions and code generation. This tool can be used for various purposes, from basic mathematical operations such as derivation, to high-level programming tasks like expression splitting.

## 2. SPECIFICATIONS

RAMA (Rule Applicator for Mathematical Analysis) is a Java tool dedicated to analysis and treatment of mathematical expressions, and code generation. RAMA is built in order to perform complex operations, decomposed into a set of elementary operations, on mathematical expressions. Such operations are named "rules" as RAMA acts like an inference motor.

The specifications of RAMA are then to be:

- lightweight : it has to perform quick operation without taking a large amount of time and of memory. Beside, it has to be easily distributable.
- generic : it has to be able to handle any type of mathematical expressions, and it has to apply every action defined by the user on these expressions. These actions are described by using a "rule" representation.
- easily extensible : new actions can be defined within a relatively short time and without any specific programming knowledge

- easy to use : actions are described in an intuitive way, and applying actions on an expression requires a small amount of instructions.

Symbolic softwares, like Maple [1], Mathematica [2] and Macsyma, etc., offer great possibilities for symbolic treatments. However, they are limited when it comes to generate computation code in several programming languages. In the paper, the aim of RAMA is mainly code generation with simple symbolic treatments, rather than complex ones.

## 3. SOFTWARE ARCHITECTURE

The RAMA architecture is based on an intern representation of mathematical expressions, and a set of operations to perform loaded from an XML file. In RAMA, these operations are known as "rules".

The intern representation of mathematical expressions is named MOM (which stands for Mathematical Object Model). Basically, a MOM is a tree, which is a classical representation for mathematical expressions [3][4].

RAMA is a "tree walker". By exploring the tree (MOM), rules are selected depending on the current explored structure.

An operation, for example derivation, is described by the user who provides a set of rules describing the operation to perform. These rules are transcriptions of the elementary mathematical actions that represent the result of symbolic treatment. For example, the result for the application of the differentiation operation on the node  $\sin(x)$  is  $dx \cdot \cos(x)$ . This defines an elementary rule for the differentiation operation. These rules are written in a XML format, like shown on Fig. 1 :

```
<RAMA:RULE>
  <RAMA:CONTEXT priority="2">
    <RAMA:MOM name="sin"/>
  </RAMA:CONTEXT>
  <RAMA:RESULT>
    <RAMA:MOM name="*" type="operator">
      <RAMA:apply-ruleset name="differentiate" select="self/child:*/>
      <RAMA:MOM name="cos" type="function">
        <RAMA:copy select="self/child:*/>
      </RAMA:MOM>
    </RAMA:MOM>
  </RAMA:RESULT>
</RAMA:RULE>
```

Figure 1: the rule for the differentiation of sinus

An operation is defined by a certain number of rules, which form this operation's rule set. This rule set is coded in an XML format and written in a file (one file per rule set).

The general operating of RAMA is presented in Fig. 2.

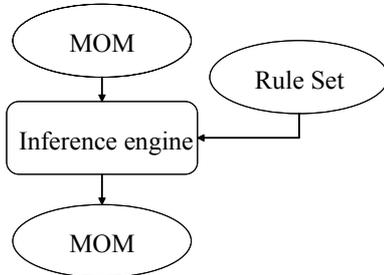


Figure 2: general operating of RAMA

The rule set is applied on a MOM object, and builds another MOM, which represents the result of the application of the operation defined by the rule set.

### 3.1. The Mathematical Object Model (MOM)

The Mathematical Object Model, or MOM, is a tree representation of mathematical expressions. The tree nodes are the operators, the functions, the variables and the constants of the expressions. A node is defined by :

- a type (operator, function, variable or constant)
- a name, represented by a string reproducing its identity
- some children.

An example of a MOM tree is given in Fig. 3.

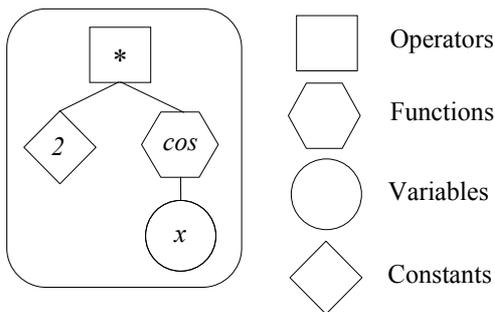


Figure 3: MOM representation of  $2 \cos(x)$

While nodes representing the operators (e.g. +, -, \*, /) are binary ones, nodes representing the sign operators + and - are unary nodes. Nodes representing variables and constants cannot have any child. Finally, nodes representing functions are planar nodes (they can have any number of children). The MOM representation of expressions is rather simple but very efficient for RAMA purposes.

### 3.2. The inference engine architecture

The principle of the inference engine is similar to the one found in expert systems (see fig 4). While an expert system values action to be performed depending on the current context, the inference engine selects a rule considering the current selected node in a tree (a MOM). A rule is defined by a context and a result. The context is at least a single node, but can be more detailed, with a whole branch of a tree. The result contains the description of the tree to build by the application of the rule on the node which is compliant to the context. This operating is illustrated in Fig. 4.

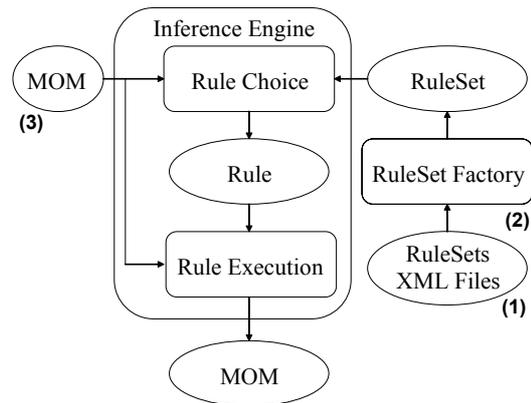


Figure 4: the operating of the rule applicator

The rule sets are described in an XML files (1), to ensure easy extensibility. Classical mathematical operations can be described as a list of rules formed as following :

*“if the object is compliant with <CONTEXT> then the result is <RESULT>”.*

It is very convenient to declare actions as a set of rules, in a file. This file is parsed and a “factory” (2) instantiates each elementary rule to be used by the inference engine.

Then a MOM is explored by the inference engine. This inference engine selects a rule from the set that complies with this MOM (3), and the associated actions are finally performed. The rule choice strategy is made in two steps :

- in the first step, all the rules applicable to the MOM object are selected
- in the second step, among these ones, the rule with the highest priority is chosen.

As the application of a rule results in an other MOM, this one can be reused by the inference engine to apply another rule set, or to translate the resulting MOM into other formats, such as ANSI-C mathematical expression (C code generation), or to a Java computation code, or to any other language for which a translator has been written.

### 3.3. The XML Rule Set Format

The rules are gathered together into a rule set. This rule set defines an operation, and its rules are written in a specific XML format. This structure is easy to understand for the user.

A rule set is a file that contains as much rules as it is needed to describe the operation. Each of the rules is composed of two parts:

- an application context : this is the context in which the rules applies. When RAMA is “walking” the tree, it comes up with a node on which it has to apply the operation wanted by the user. This node is compared with all the contexts contained in the rule set. Among the compliant contexts found, the one with the highest priority is chosen. The priority of each context is given by the user in the rule set file.
- the result of the application of the rule on this context, given by its tree representation.

Some basic actions are available to build the result of the application of the rule, like the copy of nodes, the call to the application of a rule set on the selected nodes, etc...

### 3.4. Rule example

For example, the result of the application of the action differentiation on the node  $\tan(x)$  is :

$$dx \cdot (1 + \tan^2(x)).$$

This is an elementary derivation rule. In this rule set for the differentiation operation, the rule defining the differentiation of tangent (“tan”) nodes will be written as shown in Fig. 5.

```

<RAMA:RULE>
  <RAMA:CONTEXT priority="2">
    <RAMA:MOM name="tan"/>
  </RAMA:CONTEXT>
  <RAMA:RESULT>
    <RAMA:MOM name="*" type="operator">
      <RAMA:apply-ruleset name="differentiate" select="self/child::*"/>
      <RAMA:MOM name="+" type="operator">
        <RAMA:MOM name="1" type="constant"/>
        <RAMA:MOM name="pow" type="function">
          <RAMA:copy select="self"/>
          <RAMA:MOM name="2" type="constant"/>
        </RAMA:MOM>
      </RAMA:MOM>
    </RAMA:MOM>
  </RAMA:RESULT>
</RAMA:RULE>

```

Figure 5: the rule for the differentiation of “tan”

When a “tan” node is encountered by the rule applicator, the result of the application of the derivation is a “\*” node with two children :

- the result of the application of the derivation action on the child of the “tan” function node, namely its parameter.
- a “+” node with a child being a “1” constant node, and its other child being a “pow” function node with two children :

the copy of the initial “tan” node and the constant node “2”.

This result can be illustrated by the following tree (see fig. 6):

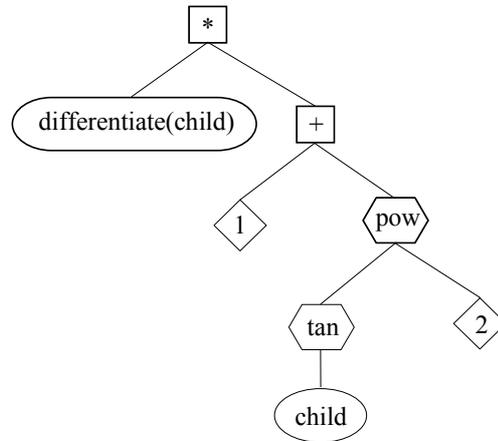


Figure 6 : the tree representation of the differentiation rule of “tan”

The path format used for node selection (like for example on the “apply-ruleset” node, to select the node on which the rule set will be applied) is derived from the W3C XPath specification [5], adapted to be specific to the MOM norm.

Finally, the application of an action on an expression consists in applying the dedicated rule set to the tree representation root. With the recursivity shown above (the call to the rule set application in the result of a rule), the symbolic treatment propagates itself along the branches of the MOM tree.

## 4. APPLICATIONS

At the present time, RAMA can perform various actions.

It is used for code generation purposes. The generated code may be written in C/C++, Java, or any other programming language. This code can be generated for simulation purposes [6], for sizing processes using optimization techniques [7], or any other purpose.

RAMA is also used to split up mathematical expressions into smaller ones, in order to reduce parsing times, and compilation times of the corresponding generated codes (for example in C programming). It is also used to split up complex expressions into their real and imaginary parts. Finally, RAMA also derives or differentiates mathematical expressions.

### 4.1. Differentiation

The differentiation of mathematical expressions is based on a rule set containing 25 rules for the operators, the usual mathematical functions, the

variables and the constants. The computing time is equivalent to Maple® one, but the memory occupation is far lesser with RAMA. However, the obtained expressions are not so simplified than with tools like Maple. Mainly, factorization methods are not applied.

#### 4.2. Java and C Code Generation

By defining a dedicated rule set, it is possible to transform the representation of an expression to be compliant with a language. For example, RAMA is used to perform such an operation in a tool called MAEL [6] that produces JAVA and C code for simulation processes and optimization processes.

#### 4.3. Real And Imaginary Parts Of Expressions

RAMA can be used to perform any action that can be described by a set of rules. Another action is the separation of real and imaginary part from a complex expression. At the present time, this functionality is used in MAEL, and is described by 15 rules. Such an operation performed on 224 elementary expressions, has a lower cost than using the functions of MAPLE software as an independent process. However, as for differentiation, the obtained expressions are not so simplified than with tools like Maple.

#### 5. CONCLUSION

At the present time, RAMA is used to performed treatment on mathematical expressions. Extensions to treat other structures which can be represented through a tree, like algorithm, may be also possible. The architecture of this tool is extensible to other activities thanks to the use of structured representations based on XML. Besides rules and actions, it is possible to build checking process, i.e. tests that are performed on a tree (e.g., to value the degree of a polynomial expression).

#### 6. REFERENCES

- [1] Maple : <http://www.maplesoft.com/>
- [2] Mathematica : <http://www.wolfram.com>
- [3] J. Davenport, Y. Siret, E. Tourmier, "Calcul formel. Systèmes et algorithmes de manipulations algébriques". Editions Masson 1993 275 pp., ISBN : 2 225 84200
- [4] M. Gondran and M. Minoux. "Graphes et algorithmes". Eyrolles, Paris, 3rd edition, 1995.
- [5] XPath, <http://www.w3.org/TR/xpath>
- [6] Loïg Allain, Laurent Gerbaud, Ch. Van Der Schaege, "Modeling electromechanical actuators for simulation : MAEL performs model capitalization and symbolic treatment", EPE'2003 (European conference on Power Electronics and

Applications), Toulouse, France, 2-4 September 2003

[7] Vincent Fischer, Laurent Gerbaud, Jean Bigeon, "Solving ODE for Optimisation : specific use of the Matrix Exponential Approach", OIPE'2002 (Optimization and Inverse Problem in Electromagnetism), Lotz, Poland, 12-14 September 2002

# CREATING DEVS COMPONENTS WITH THE METAMODELLING TOOL ATOM<sup>3</sup>

Andriy Levytskyy<sup>†</sup>

Eugène J.H. Kerckhoffs

Faculty of Information Technology and Systems  
Mediamatica Department  
Delft University of Technology,  
Mekelweg 4, 2628 CD Delft, The Netherlands  
a.levytskyy@cs.tudelft.nl

Ernesto Posse

Hans Vangheluwe

Modelling, Simulation and Design Lab (MSDL)  
School of Computer Science  
McGill University  
3480 University St., Montréal, Quebec, Canada H3A 2A7  
http://msdl.cs.mcgill.ca

## KEYWORDS

DEVS, Metamodelling, Graph Transformation, Domain-specific Modelling and Simulation Environments.

## ABSTRACT

DEVS is a well-known formalism that provides a rigorous basis for discrete event modelling and simulation. In this paper we present two possible DEVS metamodelling frameworks that are used to automatically generate 1) a tool that allows the graphical definition of DEVS models and 2) Zope products that allow storing DEVS models in a model library under the Web Application Server Zope. The tool is capable of generating a representation suitable for simulation by an external DEVS solver. The generation of executable model representations and Zope products is realized by graph transformation. The tool, the simulator and the model library form a dedicated DEVS modelling and simulation environment. The paper demonstrates how dedicated, domain/problem-specific modelling and simulation environments can be easily generated from metamodelling frameworks using graph transformation.

## 1. INTRODUCTION

The emergence of the world-wide web (WWW) and its popularity in the simulation community gave birth to the concept of *web-based simulation* (Fishwick 1996). This now includes (among others) activities that deal with the use of the WWW as an infrastructure to support distributed simulation execution. It also encompasses research into tools, environments and frameworks that support the distributed, collaborative design and development of simulation models (Page 1998).

Within this context we started a Collaborative Simulation project in which a generic web environment is developed to support simulation and modelling components in multidisciplinary collaborative projects (Levytskyy and Kerckhoffs 2000a). The practical application of our prototyped environment lies in the

NanoComp project ([nanocom.et.tudelft.nl](http://nanocom.et.tudelft.nl)), which investigates computing systems based on quantum devices. Hence is the name NanoComp Simulation Environment (NCSE). The environment's functionality is similar to that of the DLR-Virtual Laboratory (Ernst et al. 2003) and provides registered (web-)users with model registration and access to experiments. Registered models and tools are treated as limited Internet resources and are organized into a central Model Library. NCSE itself runs on top of the Web Application Server Zope ([www.zope.org](http://www.zope.org)) and its *resources* are created from Zope products.

Since 2002, all major parts of the environment are no longer coded, but rather (meta)modelled. Code is automatically generated by means of ATOM<sup>3</sup> (A Tool for Multi-formalism and Meta-Modelling) (de Lara and Vangheluwe 2002), which is also a part of the NCSE as a generic modelling client. ATOM<sup>3</sup> is a visual tool that uses metamodelling and graph grammars to specify and generate domain-specific environments. Meta-modelling refers to modelling formalism concepts at a meta-level, and model transformation refers to the automatic conversion, translation or modification of a model of a given formalism into another model in the same or different formalism (Vangheluwe et al. 2002).

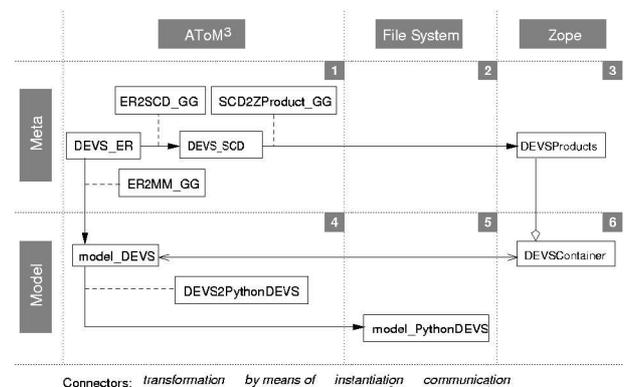


Figure 1: NCSE within a Metamodelling and Graph Transformation Framework

<sup>†</sup> Part of this work was carried out while the first author was a visiting researcher at the MSDL.

This paper demonstrates how meta-modelling and graph transformation can be used to construct a modelling and simulation environment, in this particular case dedicated to DEVS, a well-known formalism that provides a rigorous basis for discrete event modelling and simulation (Zeigler et al. 2000). Figure 1 depicts the process of constructing a DEVS environment within the NCSE. There are three domains: “AToM<sup>3</sup>”, “File System” and “Zope”. “Meta” and “Model” levels span across these domains. Their intersections create six zones. A number of models specified in various formalisms are present in the figure. Graph Grammar models specify transformations. When applied to a model, they convert it into another form and, only specific to our depiction, may transfer it into a different zone. We use `<model>_<formalism>` syntax to name models throughout the paper. For example, DEVS\_ER denotes a (meta)model of the DEVS formalism, described in the Entity Relationship (ER) formalism. Similarly, ER2SCD\_GG denotes a model of the transformation between a model in the ER formalism into its equivalent in the Simplified Class Diagrams (SCD) formalism, described in the Graph Grammar (GG) formalism.

AToM<sup>3</sup> allows one to create a DEVS metamodel in a formalism, such as ER (zone 1 in Figure 1). The ER2SCD\_GG transformation automatically converts DEVS\_ER into another form: DEVS\_SCD, which is the starting point for automated code generation for Zope. SCD2ZProduct\_GG produces DEVS Products for Zope (zone 3), from which clients can create containers for DEVS models at the lower level (zone 6). The DEVS tool itself (zone 4) is automatically generated from metamodel DEVS\_ER via ER2MM\_GG, an internal AToM<sup>3</sup> graph transformation discussed in (de Lara and Vangheluwe 2002). This tool provides DEVS2PythonDEVS\_GG, a transformation, which converts any valid DEVS models into a representation (zone 5) executable by an external solver. Communication between the DEVS tool (zone 4) and NCSE resources (zone 6) is being implemented.

The rest of the paper is organized as follows. Section 2 presents two possible DEVS metamodels. Section 3 presents the DEVS tool (generated from the first metamodel). Section 4 discusses the code generation schemes for an external DEVS solver and for Zope. Section 5 concludes the paper with final remarks.

## 2. METAMODELLING DEVS

Metamodelling refers to the definition or description of modelling languages or formalisms. A metamodel of a given formalism specifies the syntax of the formalism by defining the language constructs and how they are built-up in terms of other constructs.

To construct a DEVS metamodel we use Entity Relationship (ER) diagrams extended with constraints.

This is the default meta-formalism of AToM<sup>3</sup>. Constraints further restrict how a construct can be connected to another construct to be meaningful.

Each DEVS modelling construct is specified with attributes, constrained with constraints, visually presented with its appearance and participates in relationships according to its cardinality. We define each construct’s attributes with a minimum collection of features that form the basis for the DEVS semantics.

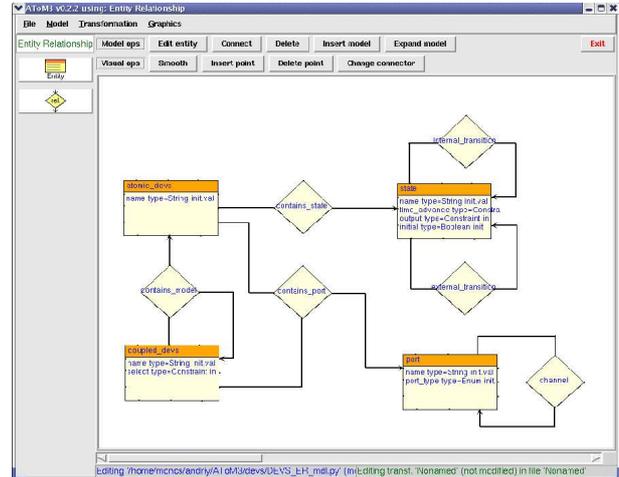


Figure 2: DEVS Metamodel in ER

The screenshot in Figure 2 illustrates how ER constructs build up a DEVS metamodel (Posse and Bolduc 2003). Note that constraints and appearance are element properties, which are not visible in the used notation. A brief formal specification of ER elements is given below (constraints and cardinalities are omitted to save space):

**atomic\_devs** implements a basic DEVS model with ports. It is a container for states and for the DEVS functions *time advance*, *internal transition*, *external transition* and *output*.

### attributes

*name* is a unique identifier of a component: *String*

*parent* is a quasi-feature that returns the reference to the parent of the **atomic\_devs** in a hierarchical model via the **contains\_model** relationship: *coupled\_devs*

*ports* is a quasi-feature that returns a collection of references to input and output ports via the **contains\_port** relationship: *Set {port}*

*states* is a quasi-feature that returns a collection of references to DEVS states via the **contains\_state** relationship: *Set {state}*

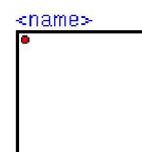


Figure 3: Atomic DEVS Appearance

### appearance

An atomic DEVS is represented as solid rectangle with its name above the top left corner (see Figure 3). It may contain states connected by transitions.

state allows a modeller to add a DEVS state to an atomic model and specify behaviour.

### attributes

*name* is a unique identifier of a component: *String*  
*initial* is a marker for the initial state: *Boolean*  
*internal\_transition* is a quasi-feature that specifies the internal transition via the *internal\_transition* relationship: *Constraint text*  
*external\_transition* is a quasi-feature that specifies the external transition via the *external\_transition* relationship: *Constraint text*  
*output* is the output function: *Constraint text*  
*time\_advance* is the time advance function: *Constraint text*

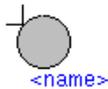


Figure 4: State Appearance

### appearance

A state is represented as solid gray circle with its name in the center (see Figure 4).

coupled\_devs is a model expressed in Coupled DEVS with ports. It is a container for atomic and coupled models. In case of Classic DEVS, the modeler has to implement the tie-breaking function *select*.

### attributes

*name* is a unique identifier of a component: *String*  
*parent* is a quasi-feature that returns the parent of the coupled\_devs in a hierarchical model via the contains\_model relationship: *coupled\_devs*  
*ports* is a quasi-feature that returns a collection of references to input and output ports via the contains\_port relationship: *Set {port}*  
*children* is a quasi-feature that returns a collection of references to children components via the contains\_model relationship: *Set {DEVS}*  
*EIC* is a quasi-feature that returns a collection of external input couplings via the channel relationship:  
*Set {((coupled\_devs, inport), (DEVS, inport))}*  
*EOC* is a quasi-feature returning a collection of external output couplings via the channel relationship:  
*Set {((DEVS, outport), (coupled\_devs, outport))}*  
*IC* is a quasi-feature that returns the collection of internal couplings via the channel relationship:  
*Set {((DEVS, outport), (DEVS, inport))}*  
*select* is the tie-breaking function: *Constraint text*

### appearance

A Coupled DEVS presentation is the same as that of the Atomic DEVS, but instead of states, it may contain instances of atomic\_devs and coupled\_devs.

port is a component's input or output interface.

### attributes

*name* is a unique identifier of a component: *String*  
*port\_type* specifies if the port is for input or output: *Enum { input, output }*



Figure 5: Port Appearance

### appearance

A port is represented as small (relative to the other appearances) square with its name labeled next to it (see ).

channel is responsible for specifying input/output connections between interfaces (ports) of DEVS components. Note on cardinality: one output port can be connected to many inports.

### appearance

A channel is represented as solid connector with an arrow end pointing to the input interface (port) of a component.

Along with the properties defined for each DEVS construct, a modeller can add global properties for the metamodel itself to, for example, document models belonging to this family of the DEVS formalism. All global properties and regular attributes are to be filled-in at the lower meta-level.

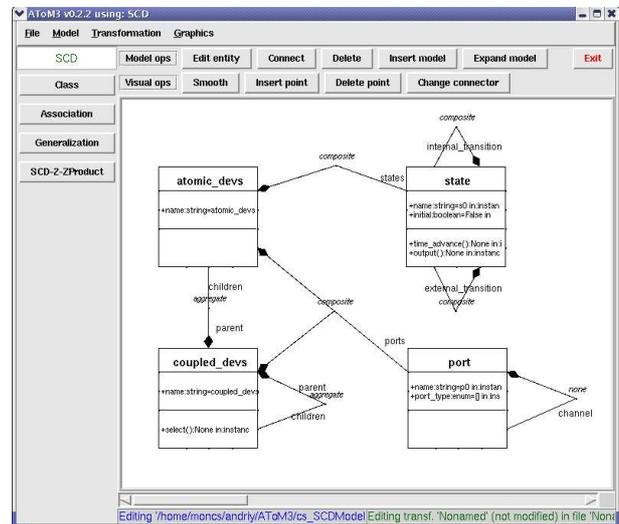


Figure 6: DEVS Metamodel in SCD

Figure 6 shows another DEVS metamodel (DEVS\_SCD) expressed in the Simplified Class Diagrams (SCD) formalism (Levytsky and Kerckhoffs 2003), a custom UML Class Diagrams-like formalism designed with ATOM<sup>3</sup> for the NanoComp project. This metamodel is equivalent to the previously described DEVS\_ER, but is less detailed: it is stripped of any syntax related to graphical nature, and any constraints related to well-formedness rules that are required for

modelling tools. The model shown is the result of automated transformation from the DEVS metamodel in ER into the SCD formalism by means of graph rewriting. This rather straightforward graph transformation is beyond the scope of this paper. DEVS\_SCD model is created solely for the purpose of code generation for Zope. We will return to it in subsection 4.2.

### 3. THE TOOL

Given the DEVS\_ER metamodel, AToM<sup>3</sup> can generate (by means of ER2MM\_GG as illustrated in Figure 1) a meta-specification, which, when loaded into the meta-level of AToM<sup>3</sup>, turns it into the modelled DEVS tool. A part of this meta-specification is a specification of the User Interface. This specification is a model in its own right and can be edited in AToM<sup>3</sup> at any time in the “Buttons” formalism. By default, this specification creates a button for every construct of the formalism. An instance of the generated DEVS modelling tool with a simple DEVS model on its canvas is shown in Figure 6. For a complete description of this tool, we refer to (Posse and Bolduc 2003).

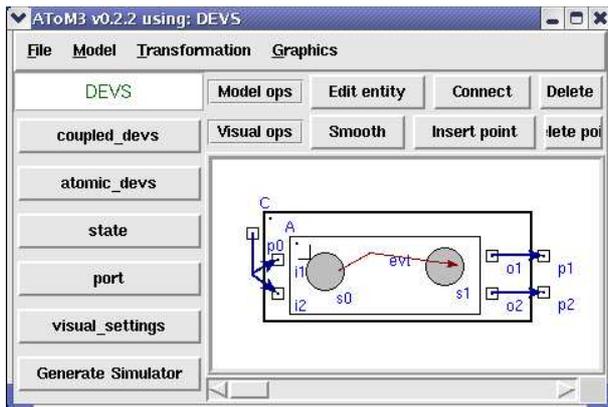


Figure 7: Generated DEVS Modeling Tool

In this tool, the user can create coupled or atomic DEVS models by clicking the corresponding button on the left and then clicking in the canvas. The same applies for each element that forms a DEVS model (states, ports, channels, and transitions.) To create a channel between ports, or a state transition, the user clicks on the **Connect** button, then clicks on the source and finally on the target. If the link is a state transition, the user is asked to select whether it is an internal or an external transition. To specify that a component is part of another (e.g. a submodel, or a port,) the user clicks on the **Connect** button, then clicks on the “parent”, and finally on the “child”. Each graphical element, which is not a link, has a label with its name. This can be edited using the **Edit** button.

Ports are labelled as either input or output ports. Each state has two attributes apart from its name. These are two fields which may contain an arbitrary Python script to specify the time-advance and output for the state.

External transitions between states also have an additional attribute which may contain some Python script to specify whether the transition is enabled or not. This script has as parameters the source state, the elapsed time, and the values at the input ports. It should return true or false. For example, if there is an external transition link between two states  $s_0$  and  $s_1$ , labelled with a condition such as  $e < 1.0$  and  $x_1 = 3$ , where  $e$  is a variable representing the elapsed-time since the last transition, and  $x_1$  is the name of some input port, then the external transition will take place if the condition becomes true. All these attributes for ports, states and external transitions can be specified by using the **Edit** button.

The **Generate simulator** button is used to produce the Python code for the DEVS model on the canvas. The generated code is a textual representation of DEVS models that can be used by the PythonDEVS simulator (Bolduc and Vangheluwe 2002), an implementation of the standard classic DEVS simulation algorithm. The underlying graph transformation (Posse and Bolduc 2003) is briefly described in subsection 4.1.

### 4. GRAPH TRANSFORMATION

One approach to manipulate graphical structures, such as our representation of DEVS models, is graph transformation. Graph transformation extends the idea of term rewriting to arbitrary graphs. The theory behind graph transformation has been thoroughly studied (see for example (Rozenberg 1999)), but there are still few software tools that support it. AToM<sup>3</sup> is one such tool.

The central notion in graph transformation is that of a *graph grammar*. A graph grammar is a collection of *productions* or *rules* specifying how a (sub)graph of a so-called *host graph* can be replaced by another (sub)graph.

Some graph grammars are enriched by associating with each rule, some additional conditions and actions. These can be used to model side-effects.

Informally, the operational semantics of graph grammars is as follows. We start from a *host graph* and a graph grammar. A *direct derivation* is the result of matching some subgraph of the host graph to the left-hand side of some rule in the grammar, checking if the additional condition is true, and if so, replacing that subgraph by the corresponding right-hand side of the rule, subsequently performing any additional actions associated with the rule. Some graph rewriting systems associate priorities to the rules, so that if more than one rule matches the host graph, the priorities act as tie-breakers. An *execution* or *trace* is a sequence of direct derivations<sup>1</sup>.

<sup>1</sup>This informal definition, as implemented in AToM3, is most closely related to the so-called SPO approach to graph transformation (Rozenberg 1999; Ehrig 1979).

Graph transformation has been used in a plethora of applications, such as specifying the operational semantics of graphical languages, and specifying formalism translations. This paper demonstrate two graph transformations that generate Python code (see below).

Code generation can be understood in terms of formalism transformation where the original representation is the source formalism and the language of the generated code is the target formalism. While it is theoretically possible to provide a purely graphical translation from a formalism such as DEVS into a real programming language such as Python, it is not a very practical approach, since it would require defining a meta-model for the target language. Real programming languages have too many constructs and special cases to make this approach feasible in practice. However, we can still have a graph transformation approach since rules in a graph grammar can have associated actions encoding side-effects. In our approach we use the graphical nature of the source formalism to traverse and annotate the model which is being translated, while the rule's actions generate the associated code.

#### 4.1 PythonDEVS Code Generation

In order to generate simulators from DEVS models represented graphically we use graph transformation. This, however, requires us to introduce some extensions to the meta-model. In particular we need some “pointers” or “markers” to traverse the DEVS model and mark which submodels have been already processed. There are two equivalent approaches to this: 1) use a graphical pointer, or 2) use an attribute in the nodes to represent the fact that a node has already been visited. Our graph grammar uses the second approach.

Another issue in the code generation scheme is that for a given model node we might require access to several of its neighbour nodes to generate its code. For instance, when generating code for any model we need to know which are the node's ports, or when generating code for a coupled model we need to know which are its submodels. None of these situations can be handled by a single rewriting rule, since the left-hand side of a rule always has a fixed number of nodes, but we need to apply the rule of interest for an arbitrary number of neighbours. One possible solution is to create a special “collecting” node, and have a rule that adds the neighbours to a list in this collecting node. This rule, when applied, marks each neighbour as visited so that it is not added twice. The rule also should have a priority higher than that of the actual code generation of the model of interest, since code generation should happen only after all the relevant neighbour's information has been collected.

The code generation rules themselves do not perform any important rewriting aside from getting rid of annotations such as the collecting nodes mentioned

above. The code generation is performed by the actions, which can access the annotations.

An example of the code generation rule of a coupled model, showing the collecting nodes is depicted in Figure 8. The collecting nodes (S and P) each contain a list of the names of the submodel nodes and port nodes respectively. The rule simply deletes the annotations (the collecting nodes,) and its action is to call an external function passing it the model and the relevant annotations. The action is executed before the graph rewriting takes place. The rule also marks the model's node as visited so that it will not be applied again to that node.

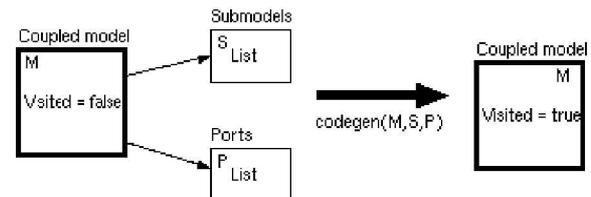


Figure 8: A Typical Code Generation Rule

As an example, consider the model in Figure 7. In the atomic model *A*, there is an external transition labelled *evt* from state *s0* to state *s1*. This transition has as condition the following script:

```
if e < 1.0:
    if i1 == 'a' and i2 == 0 or i1 == 'b'
        or i2 > 0: return 1
    else: return 0
elif e < 2.0: return i2 >= 1
else: return 0
```

where 0 stands for false and 1 for true, following Python's convention. Then the PythonDEVS code generated for *A*, is as follows:

```
class A(AtomicDEVS):
    def __init__(self):
        AtomicDEVS.__init__(self)
        self.state = 's0'
        self.elapsed = 0.0
        self.i1 = self.addInPort()
        self.i2 = self.addInPort()
        self.o1 = self.addOutPort()
        self.o2 = self.addOutPort()
    def extTransition(self):
        s = self.state
        e = self.elapsed
        i1 = self.peek(self.i1)
        i2 = self.peek(self.i2)
        if s == 's0':
            def guard1_condition(e, i1, i2):
                if e < 1.0:
                    if i1 == 'a' and i2 == 0
                        or i1 == 'b'
                        or i2 > 0: return 1
                    else: return 0
                elif e < 2.0: return i2 >= 1
                else: return 0
            if guard1_condition(e, i1, i2):
                return 's1'
```

#### 4.2 Zope Product Generation

The sequel describes the transformation *SCD2ZProduct\_GG*, which, given an SCD model (in this case, *DEVS\_SCD*), can generate Python code for a corresponding Zope product. This transformation also extends the source model with “markers” in a way

similar to that described above. A detailed description of this transformation can be found in (Levytsky and Kerckhoffs 2003).

Figure 9 shows a somewhat modified part of the metamodel related to the “Atomic DEVS” component. The core of this diagram is concrete class `atomic_devs`. In addition, we created two “dummy” classes `MRD` (Metadata for Resource Discovery) and `ARV` (Abstract Resource View) that are defined outside the namespace of this model. These “imported” classes provide features that enable on-line registration, discovery and processing of NCSE resources (Levytsky and Kerckhoffs 2001).

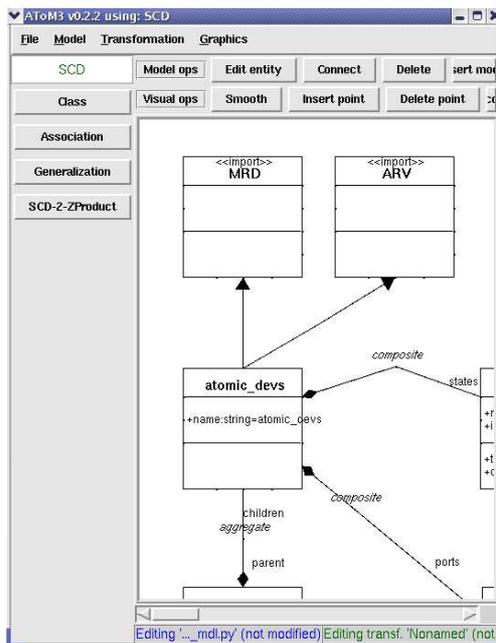


Figure 9: Atomic DEVS Component

The result of the `SCD2ZProduct_GG` transformation applied to the diagram in Figure 9, is a valid Python package implementing a Zope product. An excerpt of the code generated for Atomic DEVS product is

```
class atomic_devs(mxmSimpleItem, MRD, ARV):

    meta_type = 'atomic_devs'
    _allowed_meta_types = ('state', 'port')
    _properties = (
        {'type': 'string', 'id': 'name'},
    ) + MRD._properties + ARV._properties

    def parent (self):
        '''Return the parent coupled_devs.'''
        return self.getParentNode()

    def states (self):
        '''Return states of this atomic_devs.'''
        return self.objectValues('state')

    def ports (self):
        '''Return ports of this atomic_devs.'''
        return self.objectValues('port')

    index_html = HTMLFile('www/index_html',
        globals())
```

At this point, a Zope developer can finalize the synthesized product by for example specifying an implementation of the public interface `index_html` in the external HTML file `www/index_html`, and install the package in the `Products` directory of the Zope installation. After Zope has been restarted, a new type of objects, namely `atomic_devs`, can be created under Zope in the NCSE Model Library. These new objects, just like any other NCSE resources, are easily documented, searchable and executable on-line in a standard manner (Levytsky and Kerckhoffs 2001).

## 5. CONCLUSIONS

In this paper we have introduced a DEVS modelling environment, which allows the graphical definition of DEVS models, generates well-structured dedicated simulators for the models and allows storing the models in a central repository in a consistent manner. We also emphasize metamodeling and graph transformation as suitable frameworks for the construction of such dedicated modelling and simulation environments.

Future work will be done to implement the communication between the DEVS tool (zone 4) and the NCSE Model Library (zone 6) as shown in Figure 1. This connectivity will enable us to store semantic information of DEVS models (both atomic and coupled) in the Model Library under Zope and later, retrieve this information into the DEVS tool to reuse it in modelling components. This implements the *context-out* and *context-in* ideas introduced by (Bernardi and Santucci 2003)

## ACKNOWLEDGEMENT

The research reported in this paper is done in the framework of the NanoComp project, sponsored by the TU-Delft, and in close co-operation with the Modelling, Simulation and Design Lab (MSDL) of the School of Computer Science of McGill University, Montréal, Canada. The authors wish to thank Juan de Lara of Universidad Autónoma de Madrid, Spain, for his work on AToM<sup>3</sup>.

## REFERENCES

- Bernardi, F. and J.-F. Santucci. 2003. “Domain Management in a Hierarchical Generic Models Library”. In A. Bruzzone and Mhamed Itmi, editors, *Summer Computer Simulation Conference*, pp. 855 – 860. Society for Computer Simulation International (SCS), July 2003. Montréal, Canada.
- Bolduc, J.S. and H. Vangheluwe. 2002. “A modeling and simulation package for classic hierarchical DEVS”. Technical report, Modelling, Simulation and Design Lab (MSDL), School of Computer Science, McGill University. <http://moncs.cs.mcgill.ca/MSDL/research/projects/DEVS/>
- de Lara, J. and H. Vangheluwe. 2002. “AToM<sup>3</sup>: A Tool for Multi-Formalism Modelling and Meta-Modelling”. In: *European Conferences on Theory And Practice of*

- Software Engineering ETAPS02, Fundamental Approaches to Software Engineering (FASE)*. Lecture Notes in Comp. Sc. 2306, Springer-Verlag, pp.174 – 188.
- Ehrig, H. 1979. *Introduction to the algebraic theory of graph grammars* (a survey). 73:1-69, 1979.
- Ernst, T.; T. Rother; F. Schreier; J. Wauer; and W. Balzer. 2003. "DLR's VirtualLab: Scientific Software Just a Mouse Click Away", *Computing in Science & Engineering* magazine, vol.5, no.1, Jan./Feb.: pp. 70-79
- Fishwick, P.A. 1996. "Web-Based Simulation". In: *Proceedings of the 1996 Winter Simulation Conference*, pp. 772 – 779.
- Levytskyy, A. and E.J.H. Kerckhoffs. 2000. "Towards a Prototype Web-Based Collaborative Simulation Environment", SCS: paper of the 5th Euromedia Conference, May 2000, pp. 60 – 66.
- Levytskyy, A. and E.J.H. Kerckhoffs. 2001. "Integration of Simulation Tools and Models in a Collaborative Environment". In: *Proceedings of 2001 European Simulation Interoperability Workshop* (London, United Kingdom, June 25-27), Simulation Interoperability Standards Organisation, pp. 407 – 415.
- Levytskyy, A. and E.J.H. Kerckhoffs. 2003. "From Class Diagrams to Zope Products with the Meta-Modelling Tool AToM<sup>3</sup>". In A. Bruzzone and Mhamed Itmi, editors, *Summer Computer Simulation Conference*, pp. 295 – 300. Society for Computer Simulation International (SCS), July 2003. Montréal, Canada.
- Page, E. H. 1998. "The rise of Web-based simulation: implications for the high level architecture". In: *Proceedings of 1998 conference on Winter simulation* (Washington, D.C., United States), pp. 1663 – 1668.
- Posse, E. and J.-S. Bolduc. 2003. "Generation of DEVS Modelling & Simulation Environments". In A. Bruzzone and Mhamed Itmi, editors, *Summer Computer Simulation Conference. Student Workshop*, pp. S139 - S146. Society for Computer Simulation International (SCS), July 2003. Montréal, Canada.
- Rozenberg, G., editor. 1999. *Handbook of Graph Grammars and Computing by Graph Transformation: Foundations*, volume 1. World Scientific.
- Vangheluwe, H.; J. de Lara; and P.J. Mosterman. 2002. "An introduction to multi-paradigm modelling and simulation". In: *Fernando Barros and Norbert Giambiasi, editors, Proceedings of the AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems)*, Lisboa, Portugal, April, pp. 9 – 20.
- Zeigler, B. P.; H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation*. Second Edition. Academic Press, San Diego, USA.

## AUTHOR BIOGRAPHIES

**Andriy Levytskyy** graduated from Chernivtsi State University, Ukraine and holds an MSc-degree in Computer Science. Currently, he is a final year PhD student at Delft University of Technology, Faculty of Information Technology and Systems, Mediamatica Department. His main interest is in constructing domain-specific modelling and simulation environments.

**Eugene J.H. Kerckhoffs** holds an MSc-degree from Delft University of Technology (1970, Physical Engineering, thesis on analogue and hybrid computer simulation) and a PhD-degree from Ghent University (1986, Computer Science, thesis on parallel continuous simulation). Currently, he is an associate professor at Delft University of Technology (Faculty of Information Technology and Systems, Mediamatica Department, Knowledge-based Systems Group). He was also a holder of the SCS Chair in Simulation Sciences at the University of Ghent, Belgium. His main interests are in neural and numeric computing, and in knowledge engineering.

**Ernesto Posse** is a PhD student at the School of Computer Science of McGill University (Montréal, Quebec, Canada) working in the Modelling, Simulation and Design Lab. His main interest is in metamodelling of dynamic-structure systems.

**Hans Vangheluwe** is an Assistant Professor in the School of Computer Science at McGill University, Montréal, Canada. He holds a DSc degree, as well as an M.Sc. in Computer Science, and BSc degrees in Theoretical Physics and Education, all from Ghent University in Belgium. At McGill University, he teaches Modelling and Simulation, as well as Software Design. He also heads the Modelling and Simulation and Design (MSDL) research lab. He has been the Principal Investigator of a number of research projects focused on the development of a multi-formalism theory for Modelling and Simulation. He is an Associate Editor for the journal *Simulation: Transactions of the Society for Modelling and Computer Simulation*. His main interest is in (meta)modelling domain-specific modelling and simulation environments.

# DESIGN OF A MULTITHREADED PARALLEL MODEL FOR FIRE SPREAD

Eric Innocenti  
Alexandre Muzy  
Antoine Aiello  
Jean-François Santucci  
*University of Corsica  
SPE – UMR CNRS 6134  
B.P. 52, Campus Grossetti  
20250 Corti. FRANCE.  
e-mail : ino@univ-corse.fr*

David R.C. HILL  
ISIMA/LIMOS UMR CNRS 6158  
Blaise Pascal University  
Campus des Cézeaux BP 10125, 63177  
Aubière Cedex France  
e-mail : hill@isima.fr

## KEYWORDS

Parallel simulation, fire modeling, DEVS formalism, multicomponent formalism.

## ABSTRACT

*We present an approach allowing the simulation of fire spread on parallel computers. The speedup obtained shows that the technique used is efficient. Our algorithm is based on the DEVS formalism for Discrete Event Simulation. Two levels of abstraction are considered: a low and a high level. The low level takes into account particular conditions (vegetation, slope, wind, etc.) through cellular independent components which have their own states and behavior. The high level of abstraction considers an area of land with a fire front as a whole unit that evolves in time and space. Our design consists in proposing a multicomponent model. A set of active elements is defined and added to the multicomponent in order to improve the parallelism and to limit computations. We develop a two levels parallel approach. The first level, relying on fork() function calls, allows portable placement parallelism on real processors. The second level based on the parallelization of the active elements is adapted for hyperthreading processors, which authorize independent threads running at the same time. We use here POSIX thread library. The full advantage of all available CPUs and a significant speedup on shared memory multiprocessor machines are obtained. Experiments and results are commented on, in the last section.*

## 1 INTRODUCTION

Computer simulation of fire spread involves spatial effects that remain a challenging problem in terms of computation time and memory, to the extent that we want to work with large propagation areas. In addition, it is important to provide decision-aid tools for fireman advisors, hence the model is based on physics in order to obtain an acceptable precision level [1].

In this paper, we expose a method able to simulate the fire propagation on a two-dimensional area. This modeling approach falls into the class of cellular propagation models [2]. Our aim is to predict the position of the fire-front, but with the help of physical equations and experimental parameters [3]. The Multicomponent approach, the DEVS formalism (Discrete Event Simulation) and parallel computing provide us with a basis to tackle the simulation of large scale areas. This work is motivated by the

following observations: accurate fire simulation based on physics is costly in computation time; taking into account large areas requires a considerable amount of memory not available on traditional computers; thus many simulation models depend on a parallel architecture.

Our answer to these issues consists in an original parallel fire model which reduces considerably the computations of cellular elements. In order to reach that objective, a propagation plan is decomposed into independent propagation domains which are computed in processes. Each one is linked to a list of cells which stores active component references, and thus we are able to limit the computations to elements that change their states. The set of active cells is distributed across different threads of the processors. Computation is then easily portable on parallel computers based on hyperthreading processors and is limited to the fire front. Moreover our model supports multiple simultaneous fire fronts. Experiments show the effectiveness of our approach both in terms of execution time and domain size.

Another contribution of this work is the introduction of a new fire spread approach, based on DEVS formalism combined with the multicomponent model, allowing two levels of abstraction. The parallel implementation of the algorithm uses the fork() calls and the POSIX threads library [4].

The paper is organized as follows. The next section recalls the concepts and the formalisms used. Section 3 describes in detail the simulation model developed. Section 4 explains the DEVS simulator implemented, the overall algorithm of the simulation is described precisely. Section 5 describes the testbed we used, including the results and discussion. Finally, the achievements and limitations are summarized and the directions of further investigations are given.

## 2. CONCEPTS AND FORMALISM

### 2.1. DEVS Formalism

Since the beginning of the 1970's, developing the theoretical basis on discrete event dynamic system modeling and simulation is an active research field.

DEVS is an abstract universal formalism used for discrete event modelling introduced by Bernard Zeigler. A basic DEVS model is a structure:

$$DEVS=(X_M, Y_M, S, \delta_{ext}, \delta_{int}, \lambda, ta)$$

where

- $X_M$  is the set of ports and input values
- $Y_M$  is the set of ports and output values
- $S$  is the set of system's states
- $\delta_{ext}$  is the external transition function
- $\delta_{int}$  is the internal transition function
- $\lambda$  is the output function
- $ta$  is the advance time function

The components of the model are described via the descriptive variables,  $S$  representing the subset of state variables.  $X$  is the subset of input variables and  $Y$  the subset of output variables. The atomic models are influenced by internal and external events. The external events are generated on one of the input ports of an atomic model; the internal events are programmed as a result of the external events and imply the model response to the outside. The atomic model activity is described by the internal transition function  $\delta_{int}$ , by the output function  $\lambda$  and by the external transition function  $\delta_{ext}$ . The reader interested in more details will benefit from the new reference book for DEVS [5].

## 2.2 Multicomponent Formalism

The modeling we consider has to take into account the structural diversity of the medium as well as the behavioral diversity of the various elements. In addition, the modelling choices should also facilitate multiple fire front and its parallelization. In order to reach these objectives, we use the multicomponent specification system introduced by Zeigler in [5].

A multicomponent is a structure:

$$MC=\langle T, X, \Omega, Y, D, \{M_d\} \rangle,$$

where

- $T$  is a time base,
  - $X$  is the input value set,
  - $\Omega$  is the set of allowable input segments,
  - $Y$  is the output value set,
  - $D$  is the set of component references.
- For all  $d \in D$ ,

$$M_d=\langle Q_d, E_d, I_d, \Delta_d, \Lambda_d \rangle$$

is a component with

- $Q_d$ , is the set of states of the component  $d$ ,
- $E_d$ , is the set of its influencers,
- $I_d$ , is the set of its influencees,
- $\Delta_d$ , is the state transition function of  $d$ ,
- $\Lambda_d$ , is the output function of  $d$ .

Based on cellular automata, multicomponents are intrinsically parallel, thus they can be implemented efficiently onto parallel computers. In our case the communication flow between processors is low due to the regularity of the elements [6,7]. As stated previously the implementation of the parallel features of our algorithm will rely on fork and POSIX thread library functions [4].

## 3. FIRE PROPAGATION MODEL

### 3.1 Physical modeling

Before tackling the modeling, the phenomenon of the combustion of a solid is first specified. If a solid material is subjected to a quite important flow of heat, it will deteriorate: this is the chemical process of pyrolysis. Combustible gases are then delivered, the resulting flame comes from their reaction in stoichiometrical proportions, with the oxygen present in the atmosphere. The flame obtained is quasi isobaric and the reaction of combustion occurs in a narrow area which can be likened to a surface. The energy released by the pyrolysis is released in the atmosphere and also carried towards inert combustibles. Thus, these radiative and convective thermal transfers between the flame and the solid as well as the heat conduction will keep the fire spread going on.

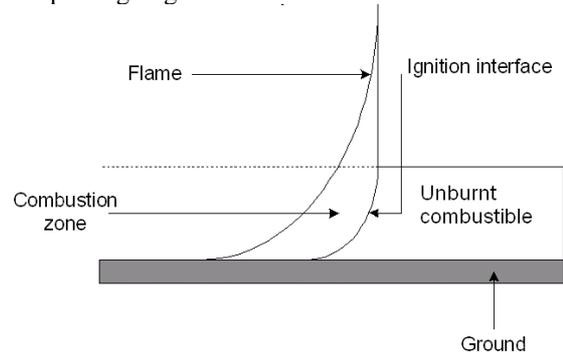


Figure 1: schematic representation of a fire front

We use as the basis for propagation behavior a one dimensional theoretical model, in which a second dimension can be obtained using propagation algorithms integrating empirically wind and slope [8]. Here, we consider fire spread within a 1 m<sup>2</sup> domain of pine needles, without slope nor wind. The spread plan is divided into elementary cells composing the ground and the plants. The previously physical study made by physicians of the University of Corsica allowed us to define a system of differential equations in order to describe the phenomenon [3].

In order to discretize the model, the method of finite elements is used so as to make its application easier. The domain considered is made up of 1 m<sup>2</sup> cells uniformly distributed and a 0.01 s time step is used. The resolution of the physical model, furnishes the following algebraic equation:

$$T_{i,j}^{k+1} = aT_{i-1,j}^k + aT_{i+1,j}^k + bT_{i,j-1}^k + bT_{i,j+1}^k + cQ\left(\frac{\partial\sigma_v}{\partial t}\right)_{i,j}^{k+1} + dT_{i,j}^k \quad (1)$$

where  $T_{i,j}$  is the temperature of one cell of the domain.

The coefficients a,b,c,d depend on the time step and the size of cells. These coefficients are identified on the basis of the experimental data of temperature according to time. This equation represents the temperature curve of a cell of the domain, as shown in figure 2. Once the temperature  $T_{ig}$  is reached, the combustion of cell starts and finishes at temperature  $T_f$ .

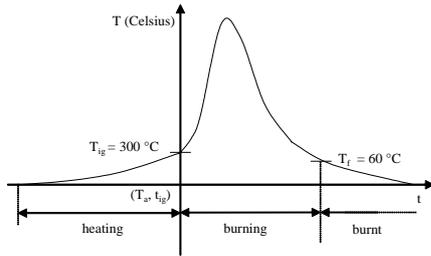


Figure 2: Simplified curve for the temperature of a cell

Since the modeling is based on the physical equation (1), it naturally makes us consider the spread domain as a cellular domain. The space is divided into a set of cells, each one with a temperature. Cell temperatures evolve through the simulation and each cell receives heat from its neighbors. A cell ignites when its temperature exceeds the  $T_{ig}$  threshold temperature. This approach will enable us to deal more easily with non homogeneous domains simply by changing the equation of an element. Consequently we are able to define different behaviors depending on of the elements of the domain.

In order to describe accurately the studied phenomenon, the multicomponent system specification described above is used because it is most suitable for the considered system.

### 3.2 Multicomponent Model

The atomic DEVS model that is envisaged is a multicomponent with an input port and an output port associated to a set of active cells. The input port starts the fire spread and the output port conveys the distribution of the temperatures of cells when the multicomponent has finished its internal transition (settlement of the temperatures). Two abstraction levels are considered: a high and a low level [9], the relation linking the two levels being a composition. At high level, the evolution of the front fire is governed by two transition functions; the external transition function  $\delta_{ext}$  which updates the global state variables; the internal transition function  $\delta_{int}$  which computes the active elements and updates the front fire. The temporary set needed for intermediate calculations is then reduced to a handful of elements.

The principal assets of our model are:

- the multicomponent approach which allows us to develop complex and accurate propagation models,
- the use of multicomponents coupled with a set of active cells which allows us to write easily parallel algorithms and take full advantage of shared memory multi-processor machines,
- the gain in time and memory requirements is substantial,
- the portability on parallel environment is facilitated.

At low level, a component is defined as being a specific address. A cell can access directly the state information of its neighbor cardinal cells. This approach allows each component to have its own set of states and a transition function. The internal transition relies on two key functions: **updateActiveSetFunction()** and **updatePropagationDomainFunction()**. The first one is responsible for updating the active cells and the second transfers the modifications on the multicomponent model using the addresses of the components. These two important stages of the internal transition involve many component function calls ( $\Delta_d, \Lambda_d$ ) that can be executed in parallel.

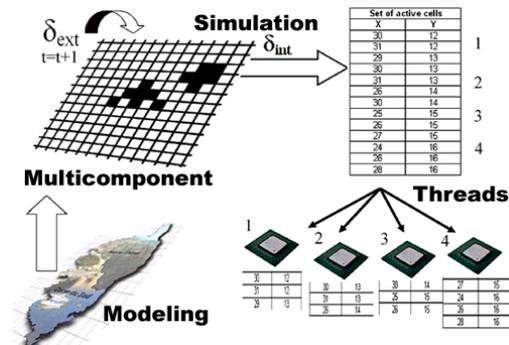


Figure 3: Schematic representation of the internal transition. The computation is limited to the fire- front, the set of active cells is partitioned in equal subsets and each one is computed into a thread

In a basic approach, the internal transition function  $\delta_{int}$  had to pass through all the components [5]. This algorithm includes the execution of their individual output functions  $\Lambda_d$  in order to define the output  $Y$  of the multicomponent, and their local state transition functions  $\Delta_d$  in order to update their states.

At component level, the first optimization consists in using a set of active cells (cells that must change their states at the next simulation time). The computation is then limited to the fire front components. Hence, this approach imposes the redefinition of the list of the active cells and for small domain sizes (less than 10 000 components in our case), component management generates overheads much greater than the original

approach. However, the overheads in the case of large scale domains are negligible.

The starting point of our second optimization is the set of active cells described above. This set is divided into equal subsets and each of them is placed into a thread. One thread has to perform the output and the transition functions of each cell of the subset and update the component linked to the multicomponent. During simulation the active cells are the most equitably packed. All the active cells of the multicomponent are updated synchronously; also we use synchronization routines to build simulation results. The evolution of the states of the active cells determines the global behavior.

At domain level, the optimization consists in decomposing the propagation area in smallest domains of propagation. Each one then constitutes an independent simulation process which is attached to a physical processor.

The disadvantages of the approach followed here are mainly memory and computation times overhead due to the large number of active components. Of course, this disadvantage is only present if a great part of the components are active, which is very improbable.

A less than optimal decomposition at high level generates bad active cells distribution that induces an imbalance in the workload distribution to the processors and requires some action, in order to rebalance the load over processors. The solution lies in the development of an efficient decomposition algorithm, which constitutes the next stage of our work.

The major advantages of this approach are the following: ability to compute multiple fire fronts, dynamic assignment of pack of components to threads, possibility to skip inactive components and last the solution retained gives faster simulation time both on sequential and parallel architectures. A transparent re-assignment of components to processors is achieved and the full advantages of hyperthreading processors, which authorize independent threads running at the same time, are exploited.

During the simulation, components burnt do not evolve any more. In this case, we remove the component from the set of the active elements, thus reducing computing and active list iterations. On the other hand, the course of the simulation will also require creation of additional active components due to the propagation of the front fire. Most generally, active components management is a combination of component removals on the one side and component adjustments on the other, augmented with packing and sending the cells and their states to another processes.

#### 4. SIMULATION ALGORITHM

The kernel of our algorithm is a discrete event simulator based on the DEVS formalism. In order to apply this formalism to fire spread, two states are defined: active and passive. The overall propagation plan is divided into propagation domains. This one is active when fire spreads, passive if not. The propagation domain is considered as an atomic DEVS model for the simulator. We create as many processes as domains we considered. An event on their

input port turns them active and they turn passive when they execute their internal transition. Each domain is attached to a process.

The algorithm can be divided into four steps:

1. Set the initial conditions for all model elements, and the initial bag of active cells.
2. Execute external transition.
3. Apply the internal transition function. We compute the next state  $S$  of the model and new cells (influencers) are added to it as it expands. The propagation domain is updated.
4. While the final simulation time is not reached, simulation back to step 3.

Events are represented through five messages which will enable us to put the simulation forward within an algorithm based on the DEVS formalism. The messages management will be made thanks to a schedule of dates, a clock guaranteeing the global time of the simulation as far as the root coordinator is concerned. The different types of exchanged messages will permit the pursuance of the simulation till final time  $t_f$ :

- (i,t)-message: is used to initialise the model with the group of rounded down values chosen by the user.
- (x,t)-message: this message is used when an external event occurs on one of the input ports of the model.
- (done,t)-message: is used to indicate that the model has completed its task; it's a message of settlement.
- (\*,t)-message: indicates a state change of the model, due to an internal event.
- (y,t)-message: indicates the emission of an output event.

The atomic model is a multicomponent with input and output ports and a set of actives cells. The input port starts the fire spread and the output port conveys the distribution of the different temperatures of the cells when the multicomponent has finished its internal transition. As depicted in algorithm 1, the simulator uses two temporal variables:  $t_{last}$  and  $t_{next}$ . The first one serves to store the simulation time once the last event has occurred and the second one serves to store the scheduling time of the next event.

```

DEVS simulator
Variables
  tlast , tnext

DEVS //→associated model

Switch (typeMessage)

  Case 'x' :
    If (tlast ≤ t ≤ tnext) Then
      e = t - tlast
      δext(x,t-message)
      tlast = t
      tnext = tlast+ta(s)
      send (done,t)-message
    Else
      Error« Bad synchronisation »
    End If
  Fin Case

```

```

Case '*' :
  If t <> tnext Then
    Error« Bad synchronisation »
  Else
    Y = λ(S)
    send (y,t)-message
    δint()
    tlast = t
    tnext = tlast+ta(s)
    send (done,t)-message
  End If
End Case
End Case
End DEVS Simulator

```

Algorithm 1: DEVS simulator

The next event time  $t_{next}$  is sent to the parent coordinator in order to permit a good synchronization of the events. The root coordinator implements the loop dedicated to the whole simulation. It distributes the tasks corresponding to the events scheduled to its direct subordinate processors. The simulation length is easily computed from the  $t_{final}$  time which is required from the user; the root coordinator simulates the propagation until the  $t_{final}$  time is reached.

```

DEVS root coordinator
Variables
  tsim //→ current time of simulation
  tfinal //→ final time of simulation
Simulator //→direct subordinate simulator

While ((Scheduler Not Empty) and (tfinal<>t))
  //→reading the first scheduler message
  Scheduler.ReadMessage()

  Switch (typeMessage)

    Case 'done' :
      tsim = t
      If (tsim < tfinal)Then
        send (*,t)-message
      End IF
    End Case

    Case 'y' :
      Save_State()
    End Case

    Case 'i' :
      // the model will evolve
      // until
      // time t of(i,t)- message
      tfinal = t
      //model initialisation
      init_Model()
      //Start simulation
      send(x,t0)-message
    End Case

  End Switch

End While
End DEVS root Coordinator

```

Algorithm 2: DEVS root coordinator

We run the simulation scheme, described above, on the different processors of the parallel computer.

## 5. RESULTS AND COMMENTS

We show here the experimental results by using the approach presented above. The experiment consists in a homogenous multiple point lighting. To simplify the analysis of the results, the burnt space is decomposed in equal propagation domains. Each domain consists in a multicomponent where each component in the matrix represents a square area of land.

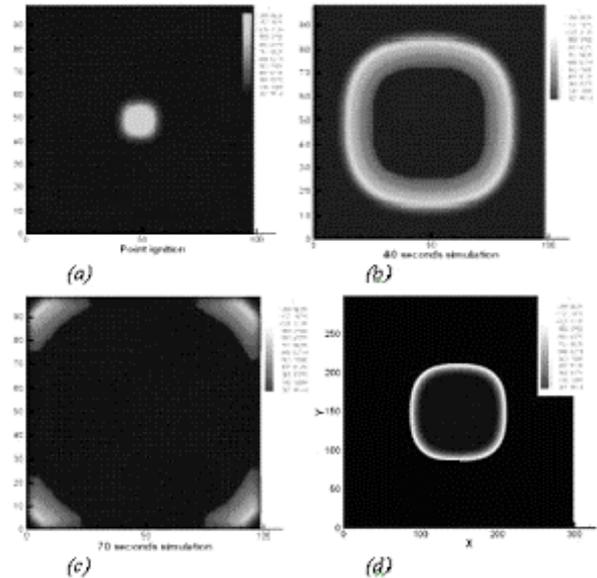


Figure 4: Four snapshots of the fire spread simulation in a multicomponent

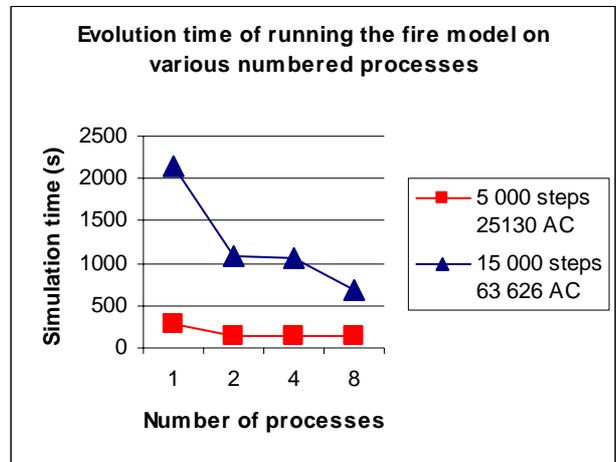


Figure 5: Simulation time for two experiments of simulation on various numbered processes. The maxima of active components reached are indicated

The measurements were obtained running on a bi-processor Intel Pentium 2.4 Ghz XEON distributed memory computer. We measured the time cost of running the fire model on various numbered processes and computed the speedups that we obtained.

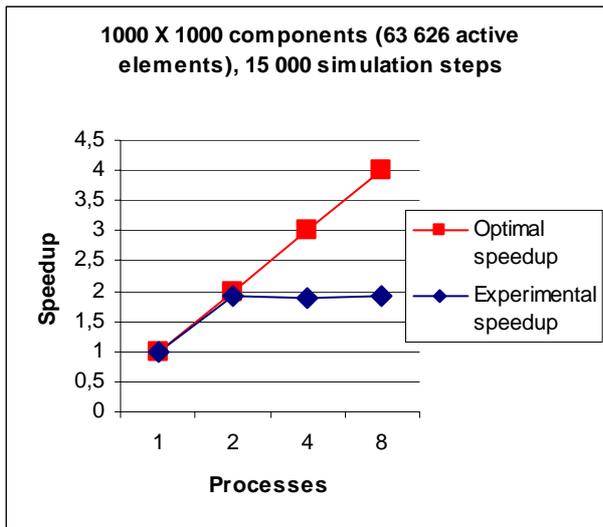


Figure 6: Speedup obtained for 150 s simulation time on a DELL XEON Pentium biprocessors

Simulation times are not directly dependent on the size of the domain, they depend on the number of active cells and processors, and the results showed in figure 6, compare them with the line  $y = x$  which we define to be the optimal speedup.

## 6. CONCLUSION

We have developed a new technique for simulating fire propagation on large domains with DEVS formalism, multicomponent modeling and parallel computation. An efficient model and a solution optimized for parallel simulation of fire spread are presented. The fire propagation is based on physical laws and a set of active cells is used to manage the fire-front. The DEVS simulator is based on a lattice of cells described as a multicomponent. This model can be applied to large areas and more complex configurations. The use of parallel techniques allowed us to obtain satisfactory computation time. In addition, the use of the multicomponent approach allowed precise description of the behavior of each cell. Our model also supports multiple simultaneous fire fronts. This approach does not require any special training in parallel computing from the end-user. The parallel simulation framework introduced, intended to simplify the parallelization of complex simulation models based on a

lattice of components. The practical aspect lies in the fact that parallelization difficulties are hidden by the model, enabling an efficient and accurate simulation exploiting computational concurrency at a minimum cost. In order to increase the number of processors, that are generally limited on SMP machines, we plan to work on an adaptation of this algorithm for the execution through a cluster of workstations.

## 7. REFERENCES

- [1] P. Eklund, S. Kirkby, J. Mann. 1999. "A Distributed Spatial Architecture for Bush Fire Simulation", Transactions on GIS, Blackwell Publishers Oxford, Vol. 3, No 3.
- [2] M. S. Veach, P. Coddington, G.C. Fox. 1994. "BURN: A Simulation of Forest Fire Propagation.". Available online from <http://citeseer.nj.nec.com/cs>.
- [3] J.H. Balbi and P.A. Santoni. 1998. "Dynamic modelling of fire spread across a fuel bed", Int. J. Wildland Fire, pp. 275-284.
- [4] F. Mueller. 1993. "A Library Implementation of POSIX Threads under UNIX" in Proceedings of the USENIX Conference, Jan, pp. 29-41.
- [5] B.P. Zeigler, H. Praehofer and T.G. Kim. 2000. "Theory of modelling and simulation", 2nd Edition, Academic Press.
- [6] D. Talia. 2000. "Solving Problems on Parallel Computers by Cellular Programming", IPDPS Workshops, Springer Verlag Heidelberg, pp. 595-603.
- [7] G. Spezzano, D. Talia, 1999. " Programming cellular automata algorithms on parallel computers" in Future Generation Computer Systems, No 16, pp. 203-216.
- [8] E. Pastor, L. Zàrate, E. Planas, J. Arnaldos. 2003. "Mathematical models and calculation systems for the study of wildland fire behaviour", Progress in Energy and Combustion Science, No 29, pp. 139-153.
- [9] J. Jorba, T. Margalef, E. Luque, J.C.S. Andre, D.X. Viegas. 1999. "Parallel Approach to the Simulation of Forest Fire Propagation", Umwelt-informatik-zwischen Theorie und Industrie-anwendung Umweltinformatik' 99. Metropolis-Verlag, Germany, pp. 69-81.

# STRIPS REPRESENTATION AND NON-COOPERATIVE STRATEGIES IN MULTI-ROBOT PLANNING

Adam Gałuszka and Andrzej Świerniak  
Institute of Automatic Control  
Silesian University of Technology  
Akademicka 16, 44-100 Gliwice, POLAND  
E-mail: agaluszka@ia.polsl.gliwice.pl

## KEYWORDS

Planning problems, multi-robot environment, STRIPS system, complexity of planning, non-cooperative strategies

## ABSTRACT

In multi-agent (multi-robot) environment each agent tries to reach its own goal and it implies that in most cases the agent goals conflict. Under some assumptions such problems can be modelled as a STRIPS system (for instance Block World environment) with one initial state and alternative of goal states. If STRIPS planning problem is invertible then it is possible to apply machinery for planning in the presence of incomplete information to solve the inverted problem and then to find a solution for the original problem. In the paper we propose the planning algorithm that solves problem described above and, based on known results, we analyse its computational complexity.

## INTRODUCTION

In multi-agent (multi-robot) environment each agent tries to achieve its own goal (Boutilier and Brafman 2001, Kraus et al. 1998). It leads to complications in problem modelling and searching for solution: in most cases agent goals are conflicting, agents have usually different capabilities and goal preferences, agents interact with problem environment simultaneously.

In this research problem environment was modelled as Block World with STRIPS representation. This domain is often used to model planning problems (Boutilier and Brafman 2001, Kraus et al. 1998, Smith and Weld 1998, Gałuszka and Swierniak 2001) because of complex actions definition and simple physical interpretation. Starting from 1970s STRIPS formalism (Nilson 1980) seems to be the most popular for planning problems (Weld 1999). Planning problems algorithms usually are at least NP-hard, even in Block World environment (here the problem of optimal planning is NP-complete).

Block World today is stated an experimentation benchmark for planning algorithms (Howe and Dahlman 2002). Also more realistic situations can be presented as Block World problems, where moving blocks correspond to moving different objects like packages, trucks and planes (Slaney and Thiebaux 2001). The case of Block World problem where the table has a limited capacity corresponds to a container loading problem (Slavin 1996).

## PROBLEM DEFINITION

We focus on the following situation:

- in the initial state there are a finite number of blocks and a table with unlimited place;
- two (or, in general case, more) robots want to rebuilt the initial state, each in its own way (each robot wants to achieve its own desired goal situation);
- goal of each robot consists of subgoals;
- each subgoal has its preference (subgoals are more or less important for robots);
- robots have different capabilities (i.e. each robot is not able to move all blocks);
- robots can not cooperate (this assumption is justified in the case where in the environment the communication is not allowed or communication equipment is broken down).

We are interested in the following two problems:

- to find a solution for above situation;
- to analyse computational complexity of searching for this solution.

## METHOD OF FINDING A SOLUTION

The problem where there are some possible initial states and one goal state is called problem of planning in the presence of incompleteness. The inverted problem is the situation with one initial state and more possible goal states. It corresponds to multi-robot Block World problem where each robot wants to achieve its own goal. If we are able to find a plan for problem of planning in the presence

of incompleteness, then it is possible to extract solution for multi-agent problem.

Below we define STRIPS System, invertible planning problem and inverse operators.

### Strips system

In general, STRIPS system is represented by four lists ( $C$ ;  $O$ ;  $I$ ;  $G$ ) (Bylander 1994, Nilson 1980):

- a finite set of ground atomic formulas ( $C$ ), called conditions;
- a finite set of operators ( $O$ );
- a finite set of predicates that denotes initial state ( $I$ );
- a finite set of predicates that denotes goal state ( $G$ ).

Initial state describes physical configuration of the blocks. Description should be complete i.e. should deal with every true predicate corresponding to the state. Goal state is a conjunction of predicates. In multi-agent environment each agent defines own goal. This description does not need to be complete. The algorithm results in an ordered set of operators which transforms an initial state into a goal state.

Operators  $O$  in STRIPS representation consist of three sublists: a precondition list ( $pre$ ), a delete list ( $add$ ) and an add list ( $del$ ). Formally an operator  $o \in O$  takes the form  $pre(o) \rightarrow add(o), del(o)$ . The precondition list is a set of predicates that must be satisfied in world-state to perform this operator. The delete list is a set of predicates that stay false after performing the operator and the add list is a set that stay true. Two last lists show effects of operator performing in problem state. Following (Koehler and Hoffmann 2000) the set of actions in a plan is denoted by  $P^o$ .

It is assumed that agents can have different capabilities (i.e. can deal with limited problem elements) and no negotiations are allowed. No negotiation assumption is satisfied in all situations where communication between agents is not allowed by problem environment or communication system fails. The case with negotiation is described for instance in (Kraus et al. 1998).

Goal preferences are also considered. We will understand the profit as a sum of preferences of goals being satisfied.

### Invertible Planning Problem

Definition of Invertible Planning Problem (Koehler and Hoffmann 2000) The problem ( $C, O, I, G$ ) is called *invertible* if and only if

$$\forall s : \forall P^o : \exists \bar{P}^o : Result( Result(s, P^o), \bar{P}^o ) = s,$$

where

$$\begin{aligned} Result(S, \langle \rangle) &= S, \\ Result(S, \langle o \rangle) &= (S \cup add(o)) \setminus del(o) \text{ if } pre(o) \subseteq S, \\ &S \text{ in the opposite case,} \\ Result(S, \langle o_1, o_2, \dots, o_n \rangle) &= Result(Result(S, \langle o_1 \rangle), \\ &\langle o_2, \dots, o_n \rangle), \end{aligned}$$

and  $\bar{P}^o$  is called *an inverted plan*.

### Inverse Operator

Definition of Inverse Operator (Koehler and Hoffmann 2000). An operator  $\bar{o} \in O$  is called inverse if and only if it has the form  $pre(\bar{o}) \rightarrow add(\bar{o}), del(\bar{o})$  and satisfies the conditions:

1.  $pre(\bar{o}) \subseteq pre(o) \cup add(o) \setminus del(o)$
2.  $add(\bar{o}) = del(o)$
3.  $del(\bar{o}) = add(o)$ .

Under closed world assumption condition applying an inverse operator leads back to previous state. It is proved that if there is an inverse operator for each operator, then the problem is invertible.

There are assumed four classical operators in Block World (Nilson 1980). The only difference is that operators *stack* and *unstack* precise only the block that is currently transformed (i.e. do not precise on which block is stacked a transformed blocks and from which block is unstacked a transformed block):

- pickup(x) - block x is picked up from the table;  
precondition list & delete list:  
*ontable(x), clear(x), handempty*  
add list: *holding(x)*
- putdown(x) - block x is put down on the table;  
precondition list & delete list: *holding(x)*  
add list: *ontable(x), clear(x), handempty*
- stack(x) - block x is stacked on any block y;  
precondition list & delete list:  
*holding(x), clear(y)*  
add list: *handempty, on(x,y), clear(x)*
- unstack(x) - block x is unstacked from any block y;  
precondition list & delete list:  
*handempty, clear(x), on(x,y)*  
add list: *holding(x), clear(y)*.

It is easy to see that *unstack* is an inverse operator for *stack* and *pickup* is an inverse operator for *putdown*. We have defined Block World as an invertible planning problem because it allows to apply planning in the presence of incompleteness methodology to search for solution of inverted multi-agent problem and then to extract solution for the right multi-agent problem.

**Plan in the presence of incompleteness as an inverted plan in multi-robot environment**

Algorithm of planning in the presence of incompleteness handle planning problems with uncertainty in the initial state (e.g. Weld et al. 1998). In this case the algorithm seeks to generate a robust plan by thinking over all possibilities. This approach is called *Conformant planning* (Smith and Weld 1998). Conformant planning algorithms develop non-conditional plans that do not rely on sensory information, but still succeed no matter which of the allowed states the world is actually in.

**Simulation results**

Block world environment was implemented using PDDL language (Planning Domain Definition Language) extended for handling uncertainty in the initial state (Yale Center... 1998). Sensory Graphplan algorithm was used to solve block world problems with uncertainty in the initial state

(www.cs.washington.edu/research/projects/www/sgp.html).

Two different problems are presented below. In both cases 2 robots are operating in the environment. In problem 1 Robot 1 is capable of moving blocks A,B and C whereas robot 2 can move blocks D, E and F. In Problem 2 Robot 1 is capable of moving blocks A, B, C and D whereas robot 2 can move blocks E, F, G and H. In both cases definitions of the operators are inverted (operator names are changed i.e. *unstack* for *stack* and *pickup* for *putdown*). It implies that the plan for the inverted problem is extract just by executing founded plan in the inverted order. In both cases agents goals are in conflict. The case when in multi-agent environment the goals do not conflict was explored in (Galuszka and Swierniak 2002).

Problem 1. The initial state is presented on figure 1. The goal state of robot 1 is on figure 2 and the goal state of robot 2 is on figure 3.

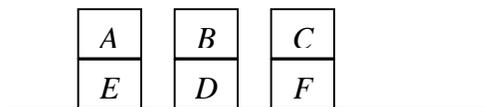


Figure 1: Initial state

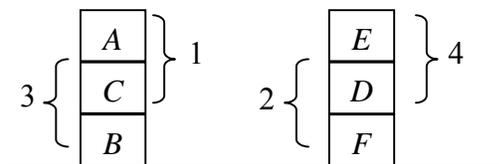


Figure 2: Desired goal state of robot 1 (the goal conflicts with the goal of robot 2) (each goal has its preference)

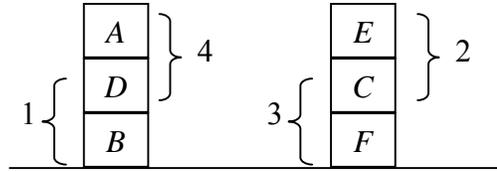


Figure 3: Desired goal state of robot 2 (the goal conflicts with the goal of robot 1) (each goal has its preference)

Solution to two-robot problem 1 (steps from 1 to 7):  
2 contexts

- step 7 - ((( STACK2 E)))
- step 6 - ((( PICK-UP2 E)) (( STACK1 A)))
- step 5 - ((( STACK2 D)) (( UNSTACK1 A)))
- step 4 - ((( PICK-UP2 D)) (( STACK1 C)))
- step 3 - ((( PUT-DOWN2 D)) (( UNSTACK1 C)))
- step 2 - ((( PICK-UP2 D)) (( PUT-DOWN1 B)))
- step 1 - ((( UNSTACK1 B)))

Problem 2. The initial state is presented on figure 4. The goal state of robot 1 is on figure 5 and the goal state of robot 2 is on figure 6.



Figure 4. Initial state for problem 2

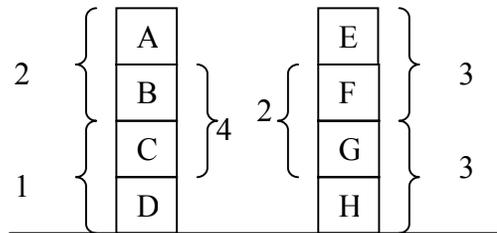


Figure 5: Desired goal state of robot 1 (the goal conflicts with the goal of robot 2) (each goal has its preference)

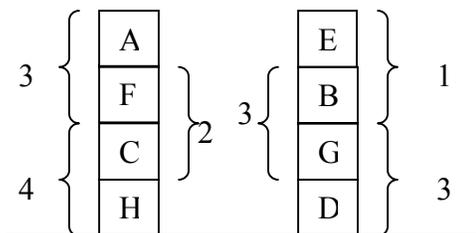


Figure 6: Desired goal state of robot 2 (the goal conflicts with the goal of robot 1) (each goal has its preference)

Solution for this two-robot problem 2 (steps from 1 to 6):

2 contexts

- step 6 - ((( STACK2 E)) (( STACK1 A)))
- step 5 - ((( PICK-UP2 E)) (( PICK-UP1 A)))
- step 4 - ((( STACK2 F)) (( STACK1 B)))
- step 3 - ((( PICK-UP2 F)) (( PICK-UP1 B)))
- step 2 - ((( STACK2 G)) (( STACK1 C)))
- step 1 - ((( PICK-UP2 G)) (( PICK-UP1 C)))

Both agents can apply the above-founded plan to satisfy their goals. However, when they are trying to achieve their goals simultaneously they are in conflict. Now we define the non-cooperative equilibrium (Nash equilibrium) [6] and indicate how the agents can maximise their profits (the sum of preferences of satisfied goals) by achieving non-cooperative (Nash) equilibrium.

### Non-cooperative equilibrium strategy

For presented problem a plan exists only if operators *stack* and *unstack* have only 1 parameter so they do not precise from which and on which block is stacked or stacked out. It implies that both agents to reach theirs goals can apply the founded plan but not simultaneously. When the goals preferences are also considered then it is possible to use Nash equilibrium strategy to precise how to apply the plan simultaneously and maximise the profit (the sum of satisfied goals preferences). The analysis of the problem leads to two remarks:

Remark 1. It is not always possible to find Nash strategy for defined problems and in general case it is depended on size of the problem.

Remark 2. More precisely the Nash strategy (if exists) defines the equilibrium for the whole plan when the number of *stack* operators in founded plan is even for each agent (2 operators for each agent in problem 1). When this condition is not satisfied (3 operators for each agent in problem 2) then the Nash strategy defines equilibrium only for a part of the problem.

The conflict between agents will be presented by a bimatrix game. Matrix A characterises the costs of the first agent (the profit with the negative sign), matrix B characterises the wastage of the second agent. We assume that agent 1 chooses rows and agent 2 chooses columns of the matrices. The agents are trying to minimise cost functions defined by matrices  $A = \{a_{ij}\}$  and  $B = \{b_{ij}\}$ .

Definition of Nash equilibrium. The strategy  $\{i_0, j_0\}$  determines non-cooperative (Nash) equilibrium in bimatrix game  $(A, B)$  if the following inequalities are satisfied:

$$a_{i_0 j_0} \leq a_{ij_0}$$

$$b_{i_0 j_0} \leq b_{i_0 j}$$

for all  $i = 1, 2 \dots n, j = 1, 2 \dots m$ .

Now we define the matrixes for problem 1. The strategies in matrices are corresponding to the plan that solves the problem 2. Agent 1 can stack block C either on B or F and block A on C or D whereas agent 2 can stack block D on B or F and block E on C or D. Values in matrices correspond to goal preferences (e.g. robot 1 stacks block A on C and robot 2 block D on F then profit of robot 1 is 5 – it satisfied 2 its subgoals - whereas profit of robot 2 is 0 – it satisfied none of its subgoals).

Table 1: Matrix A (profits of the first agent)

1 \ 2	stack D B	stack D F	stack E C	stack E D
stack C B	3	3+2	3	3+4
stack C F	0	2	0	4
stack A C	1	1+2	1	1+4
stack A D	0	2	0	4

Table 2: Matrix B (profits of the second agent)

1 \ 2	stack D B	stack D F	stack E C	stack E D
stack C B	1	0	2	0
stack C F	1+3	3	2+3	3
stack A C	1	0	2	0
stack A D	1+4	4	2+4	4

Table 3: Matrix A (costs of the first agent)

1 \ 2	stack D B	stack D F	stack E C	stack E D
stack C B	- 3	- 5	( - 3 )	- 7
stack C F	0	- 2	0	- 4
stack A C	- 1	- 3	- 1	- 5
stack A D	0	- 2	0	- 4

Table 4: Matrix B (costs of the second agent)

1 \ 2	stack D B	stack D F	stack E C	stack E D
stack C B	- 1	0	( - 2 )	0
stack C F	- 4	- 3	- 5	- 3
stack A C	- 1	0	- 2	0
stack A D	- 5	- 4	- 6	- 4

In this game we found one strategy that satisfies non-cooperative (Nash) equilibrium definition (in brackets). This strategy modifies the plan in such a way that agent 1 should place block C on B and agent 2 should place block

E on C. It leads to the situation when the final state for the problem 2 takes the form (figure 7).



Figure 7: Final state for problem 2 comes from Nash equilibrium

Finally, the profit of the first agent is now  $3 + 2 = 5$  and for the second agent  $2 + 4 = 6$ .

### COMPUTATIONAL COMPLEXITY OF SEARCHING FOR THE SOLUTION

In general planning with complete information is *NP*-complete. Planning in the presence of incompleteness belongs to the next level in the hierarchy of completeness (Baral et al. 2000). A condition for ‘incomplete’ block world problems that reduced complexity of finding a plan to the class *P* will be shown.

Non-optimal planning in Block World is easy (Gupta and Nau 1992, Bylander 1994). To analyse complexity in our case it is assumed that planning problems are limited to only completely decomposed initial state (i.e. all blocks are on the table) as it is shown in the example 2. Then the inverted problem is to decompose all possible initial states (i.e. goal definition consists only of ‘on-table’ predicates). The number of possible initials corresponds to number of robots. So the inverted problem is planning in the presence of incompleteness.

Now it will be shown a class of ‘incomplete’ block world problems for which finding a plan is also easy. In this class each block has the same position in stack in each possible initial state. This class belongs to the same class of complexity as classical block world planning.

To represent possible initial states of block world it will be used Hass Diagram (Gupta and Nau 1992). This diagram is a directed acyclic graph whose nodes are the blocks and arcs are from block *x* to *y* if and only if *on(x,y)* is in initial state. This diagram can be constructed in linear time (Gupta and Nau 1992). Since the number of possible initial states is bounded by number of robots then time necessary to built Hass diagram for all initial states is also linear.

Next for each block in each possible initial state the position in stack is calculated using Hass diagrams. It corresponds to the problem of length of path in a acyclic graph. In the problem 2 block positions are:

- for A and E – 3,
- for F and B – 2,
- for C and G – 1,
- for D and H – 0.

for two possible initial states.

If each block has the same position in stack in each possible initial state then there exists the same plan for each agent that solves the problem of decomposing all initials. So to find a plan only goal situation and block positions can be considered. Subgoals are serialised in decreasing order according to block positions. Then each subgoal can be solved using only 1 macro-operator. In Example 2 we have an order:

{(on-table A), (on-table E), (on-table F), (on-table B), (on-table C), (on-table G), (on-table D), (on-table H)}

Such order leads to a solution of Example 2.

Each step requires only polynomial-time, so presented planning problem is solved in polynomial-time (belongs to class *P* of complexity).

### CONCLUSION

Defining Block World environment as an invertible STRIPS planning problem allows to apply planning in the presence of incompleteness as a machinery of searching for a solution of inverted multi-agent problem and then extraction of a solution for the primary multi-agent problem. It is possible to use non-cooperative equilibrium strategy to improve the founded plan.

The result obtained for problems 1 and 2 using non-cooperative equilibrium definition should be understood as only example how to apply game-theoretic approach to solve rather narrow class of planning problems. The wide class of problems were not shown here e.g. how to modify the plan when there are more than one Nash equilibrium point, how to extend this methodology for more agents.

More solved problems can be found on [www.zts.ia.polsl.gliwice.pl/galuszka/index1.htm](http://www.zts.ia.polsl.gliwice.pl/galuszka/index1.htm).

### Acknowledgement

This work was supported by *State Committee for Scientific Research* grant No. 4 T11A 012 23 for the year 2003.

### REFERENCES

Baral Ch., V. Kreinovich, R.Trejo. 2000. Computational complexity of planning and approximate planning in

- the presence of incompleteness. *Artificial Intelligence*, 122: 241-267.
- Boutilier C., Brafman R.I. 2001. Partial-Order Planning with Concurrent Interacting Actions. *Journal of Artificial Intelligence Research*, 14:105-136.
- Bylander, T. 1994. The Computational Complexity of Propositional STRIPS Planning. *Artificial Intelligence*, 69:165-204.
- Gałaszka A, A. Świerniak. 2002. Planning in multi-agent environment as inverted STRIPS planning in the presence of uncertainty. *Recent Advances In Computers, Computing and Communications* (Ed. July 2002), WSEAS Press, pp.58-63.
- Gupta N., D.S. Nau. 1992. On the complexity of Blocks-World Planning. *Artificial Intelligence*, 56(2-3):223-254.
- Howe A.E., E.Dahlman. 2002. A Critical Assesment of Benchmark Comparison in Planning. *Journal of Artificial Intelligence Research* 17 (2002), pp. 1-33.
- Koehler, J.; J. Hoffmann. 2000. On Reasonable and Forced Goal Orderings and their Use in an Agenda-Driven Planning Algorithm. *Journal of Artificial Intelligence Research*, 12 (2000), pp. 339–386.
- Kraus, S.; K. Sycara; A. Evenchik. 1998. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104:1-69.
- Mesterton-Gibbons, M. 2001. *An Introduction to Game-Theoretic Modelling*. American Mathematical Society.
- Nilson, N.J. 1980. *Principles of Artificial Intelligence*. Toga Publishing Company, Palo Alto, 1980.
- CA.Slaney J., S. Thiebaux. 2001. Block World revisited. *Artificial Intelligence* 125 (2001) 119-153.
- Slavin T. 1996. Virtual port of call. *New Scientist*, June 1996, pp. 40-43.
- Smith, D.E.; D.S. Weld. 1998. Conformant Graphplan. *Proc. 15<sup>th</sup> National Conf. on AI*.
- Weld, D.S. 1999. Recent Advantages in AI Planning. Technical Report UW-CSE-98-10-01, *AI Magazine*, 1999.
- Weld, D.S., C.R. Anderson i D.E. Smith. 1998. „Extending Graphplan to Handle Uncertainty & Sensing Actions”. *Proc. 15<sup>th</sup> National Conf. on AI*, 897-904.
- Yale Center for Computational Vision and Control. 1998, *PDDL – The Planning Domain Definition Language*, Tech Report CVC TR-98-003/DCS TR-1165.

## AUTHOR BIOGRAPHIES

**ANDRZEJ ŚWIERNIAK** received M.Sc., Ph.D. and D.Sc. (habilitation) degrees in control engineering respectively in 1972, 1978 and 1988 all from the Department of Automatic Control, Silesian University of Technology in Gliwice, and M.A. in mathematics in 1975 from University of Silesia in Katowice, Poland. He is currently a professor at the Silesian University of Technology. His research interests are in modern control and optimisation theory, biomedical modelling and control, artificial intelligence and CADM.

**ADAM GAŁUSZKA** was born Ruda Slaska, Poland in June 6, 1972. He received M.Sc. degree in automation and robotics Silesian University of Technology, Poland, in 1996. Since 1996 he has been a doctorate student at the Silesian University of Technology and Teaching Assistant in Dept. of Automatic Control. In 2001 he received Ph.D. degree in automation and robotics from Silesian University of Technology, Poland. He is interested in artificial intelligence planning algorithms with STRIPS representation.

# MODELLING WITH THE INTEGRATED PERFORMANCE MODELLING ENVIRONMENT (IPME)

Anna M. Fowles-Winkler  
Micro Analysis and Design, Inc.  
4949 Pearl East Circle, Suite 300  
Boulder, CO, USA 80301  
E-mail: awinkler@maad.com

## KEYWORDS

Discrete-event simulation, High-Level Architecture, human performance, simulation tool, workload.

## ABSTRACT

The Integrated Performance Modelling Environment (IPME) is a commercially-available, Linux-based discrete-event simulation software application for building models that simulate real-life processes. With IPME models, users can gain useful information about processes that might be too expensive or time-consuming to test in the real world. Some example application areas for simulation modelling using IPME include the following:

- Evaluating procedures, workload, and staffing issues for aircraft and ships, including workstations.
- Modelling adaptation and alertness to changing time zones.
- Analysing information flow, staff sizes, and workload for a staff of 15-20 people in a military control centre.

IPME provides a plug-and-play model environment to allow for flexibility in evaluating different environments and crews. The built-in workload functionality accounts for operator resource demands in systems. Finally, IPME's communications interoperability provides an easy way to reuse existing models or simulations.

## INTRODUCTION

Micro Analysis and Design, Inc. began IPME development in 1995 with support from QinetiQ Centre for Human Sciences. In 1998, Defence Research and Development Canada—Toronto (DRDC-Toronto) joined the development program. Built from the software base of Micro Saint, a Microsoft Windows-based commercial discrete-event simulation product, IPME is an integrated environment of plug-and-play component models designed to analyze human system performance. From the beginning, development has focused on two goals: 1) providing a flexible modelling environment, allowing users to choose which components they needed for their analyses, and 2) allowing IPME to communicate with other simulation applications.

This paper will provide a basic understanding of IPME's plug-and-play models, human performance workload features, and interoperability capabilities.

## PLUG-AND-PLAY MODELS

An IPME model, or *system*, is a collection of models and data that represent what the user is currently analysing. A system is comprised of the following component models:

- An environment model
- A crew model
- A performance shaping model
- A task network model
- An experimental suite
- An external model

The only required component model is the task network model; all other models are completely optional. This section will describe all models except for the external model. The external model is discussed later in the section titled "Interoperability."

An environment model describes external factors such as physical, crew, mission, and threat factors. Physical factors include humidity and temperature. Crew factors can define how well the group performs as a team, including factors such as leadership. Mission factors include elements like intelligence and weapons reliability. Finally, threat factors describe the degree of threat from an enemy, and the position of the target. Each factor may have an associated expression that is evaluated at each simulation event.

By default an environment model includes a basic set of variables. Customizing an environment model by adding user-defined external factors allows the model to fit particular environments under consideration. Environment models of different areas can be developed and plugged into existing systems, providing a simple way to compare operator performance under different environmental factors. For example, a model of a military's ground forces could have one environment model to represent a jungle, and a second environment model to represent a desert.

A crew model defines individual operator roles, and includes operator characteristics such as non-physical traits (fitness, training), states that change during simulation (boredom, hunger), and physical properties including anthropometry (weight, height). Because

states can be updated during a simulation run, each operator defined in a crew model can have unique characteristics. Traits are generally held constant during a simulation run, but can be varied for a block of runs using the experimental suite, described later in this section. Currently, operator anthropometry cannot be varied as traits can, but a future version of IPME will include this functionality.

A performance shaping model is a collection of user-defined functions called performance shaping functions (PSFs) that modify the time it takes to complete a task, or the probability of task failure. The PSFs are linked to individual tasks through a task taxonomy, allowing one PSF function to be dynamically applied to any similar task in a model. Since PSFs can use operator states as expression variables, models can discriminate performance results as a function of operator characteristics. Therefore, simulations can have two operators performing the same task type with different, and therefore more realistic, task time and probability of failure outcomes.

The task network model is a graphical display of the system processes or tasks. Because IPME's task network model is based on the same technology as Micro Saint's task network model, the terminology used to describe the task network model components are clearly based in human performance modelling. A task network model consists of networks and tasks. Figure 1 shows a sample task network model.

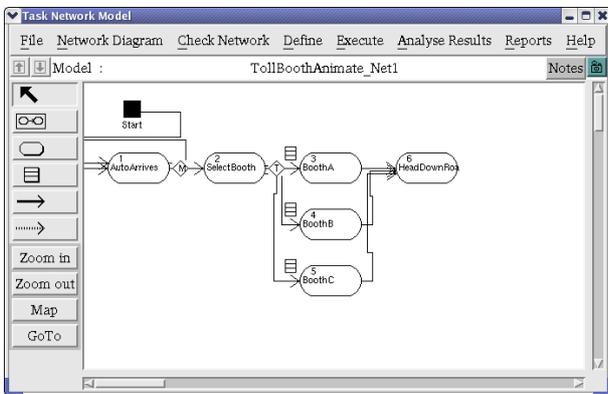


Figure 1: IPME Task Network Model Dialogue

Networks may be a sequence of tasks performed by an operator or a series of processes that define an organization. Tasks generally denote human activities, but they might represent other, non-operator processes, or logic to support the simulation. Tasks contain timing information, conditions for execution, and operator assignments. Operators from the crew model may be statically assigned to particular tasks, or they may be dynamically assigned depending on aspects such as what operators are available to perform the task.

A task has a set of expressions associated with it to control when the task executes, to control the state of

the system when the task begins or ends, and to specify what, if anything, should happen if a task fails to execute. These expressions may contain user-defined variables and functions. Variables and functions are defined globally, and may be used in any expression in any model.

While it is optional to use the environment, crew, and PSF models, using all three in combination with the task network model takes advantage of the plug-and-play nature of IPME, and the interconnectedness of the models. External factors from the environment model and operator states from the crew model may affect operator performance, with that performance being defined as a PSF. If an analysis requires different environments or crews to be analysed, those different models can be easily used with a single task network model. Figure 2 shows the relationship between these component models.

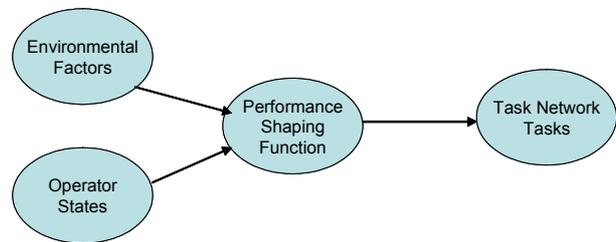


Figure 2: Component Model Relationships

After a model has been developed in IPME, the experimental suite can be used to vary system variables to show equipment, design, operator, or procedure variability. The user specifies independent and dependent variables and their values for each simulation run or for a block (a series) of runs. For example, the environmental physical factor *temperature* can be configured to have the value  $-10^{\circ}\text{C}$  for the first block of runs,  $0^{\circ}\text{C}$  for the second block of runs, and  $+10^{\circ}\text{C}$  for the final block of runs. Human trait values such as *fitness* can also vary across simulation runs and potentially represent differing human populations to be used in the model. The experimental suite provides a user-friendly method of running multiple experiment blocks with varying variable values.

Once a model has been developed, an analyst may decide to share portions of that model with other analysts. The built-in master library allows users to store and distribute operators, performance shaping functions, environment models, and networks. These models are under revision control in the master library, indicating new models and when models have changed. Users link to models in the master database in a read-only mode, preventing modification without permission. To modify a master library model, a user must either have permission to modify the library, or the user may unlink the model from the library. By unlinking a model, a user is then able to modify a local copy of the model without modifying the library version of the data.

## WORKLOAD

IPME includes the following built-in workload methodologies: Prediction of Operator Performance (POP), Visual, Auditory, Cognitive, and Psychomotor (VACP) and Workload Index (W/Index), and Information Processing/Perceptual Control Theory (IP/PCT).

POP, developed by the Defence Evaluation Research Agency (DERA) 1992-1995, predicts performance degradation from interference between concurrent tasks (MAAD 2003). Input (visual or auditory), central (mental operations), and output demands (manual or vocal) are considered for each task. Figure 3 shows the POP Workload Percentages portion of the task assignment and workload dialogue for a sample task.

Figure 3: POP Workload Percentages

The task assignment and workload dialogue includes a *Workload Percentages* section for POP workload data. The workload demand values are entered for the input, central, and/or output channels, with 100 representing maximum workload before overload. A task demand multiplier (TDM) is calculated based on the fraction of time a single resource for a single task is available to the time that it is actually used. The TDM then lengthens task time when it is multiplied with the task mean time. The implementation in IPME is symmetric; therefore, resources are divided symmetrically among tasks during simulation execution.

Tasks can be externally or internally paced in the POP algorithm. An externally paced task is scheduled by external source, for example, a supervisor requesting a report. An internally paced task is self-scheduled by the operator. Externally paced tasks are higher priority than internally paced tasks; therefore if an operator is overloaded, internally paced tasks will be rescheduled to accommodate the operator completing externally paced tasks in the time available.

VACP and W/Index measure the resource demand imposed upon an operator. The VACP algorithm measures the task loading for an operator within visual, auditory, cognitive, and psychomotor channels (McCracken 1984, Bierbaum 1987). The VACP portion of the task assignment and workload dialogue is shown in the upper portion of Figure 4.

Figure 4: VACP and W/Index Values

VACP values are selected for each of the channel categories, and then the corresponding weights are automatically generated. Each task is rated based on the weighted task demand within the channels. During simulation execution, attentional demand is calculated for each operator

The W/Index algorithm (Sarno 1992) measures the resource demand imposed upon an operator within six resource channels, and supports interference between channels. The VACP resource values can be automatically mapped to W/Index values, as seen in Figure 4 by the “Map to W/Index” button. Operator workload is then calculated during simulation execution. Figure 5 shows the instantaneous workload and demand per operator in a system. In this particular example, there is a single operator (Operator 1) in the system, with three tasks being executed concurrently by the operator. This dialogue also graphically shows the attentional demand ratings and workload values as they change.

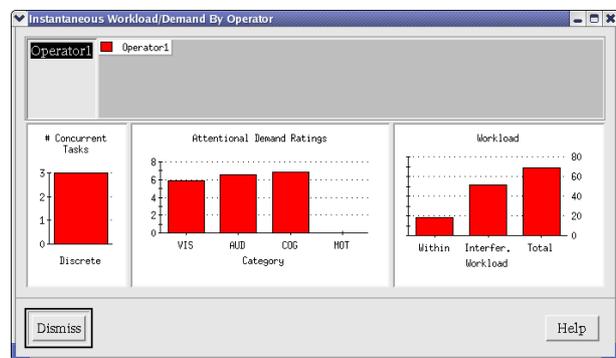


Figure 5: Instantaneous Workload and Demand

IP/PCT was developed by Keith Hendy of DRDC-Toronto. This workload methodology posits that all factors that impact human cognitive workload can be reduced to their effects on the amount of information to be processed (an operator’s cognitive limits), and the amount of time available before the task must be completed (an operator’s time pressure). According to IP/PCT, human operators change their processing strategy to reduce the amount of information to be processed, or increase the time available.

In IPME, the IP/PCT scheduler places tasks on the event queue based on how a human would order tasks. For example, the most important tasks are addressed first,

and tasks that are almost finished are completed before starting new tasks. Each task assigned to an operator is categorized according to time priority.

Additionally, cognitive and structural interference are considered. A memory limit is set for all operators, which affects the number of tasks an operator has in prospective memory, the memory before working memory. Tasks may be shed from prospective memory, meaning that the operator has forgotten about those tasks.

Structural interference between tasks appears when an operator is required to operate separate controls with a single limb, when visual focus is required for images too far apart, or when an operator is required to speak two (or more) distinct messages at the same time. The IP/PCT scheduler determines which task to execute based on time priority and available channels. Tasks that cannot execute due to interference may be delayed until they can execute, or they may be shed.

Visual, auditory, cognitive, or psychomotor interference in IP/PCT is determined by values set by the user for each task. The IP components tab, shown in Figure 6, allows the user to select the domain(s) used for the particular task, and the domain category. For example, a cognition task could be a passive monitoring task, or a skill-based task.

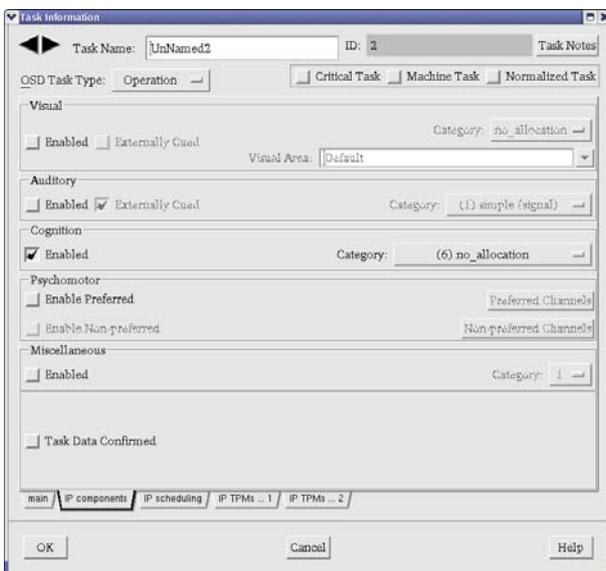


Figure 6: IP/PCT Component Domain Selection

Both IP/PCT and POP affect task execution, and may cause tasks to require more processing time, or to be delayed, rescheduled, or shed, depending on an operator's workload and other concurrent processes. These modifications to the simulation event queue occur automatically as a result of operator overload or interference, depending on the workload algorithm enabled.

## INTEROPERABILITY

The external model allows IPME to communicate with other existing models or applications. There are two supported communications protocols: a TCP/IP sockets interface and High Level Architecture (HLA).

The external TCP/IP sockets interface is a simple protocol that allows IPME to communicate with other simulations (IPME to IPME or IPME to custom simulations). This functionality extends the capabilities of the overall IPME system by allowing an IPME model to share data with custom simulations such as model optimisers, cognitive models, and applications that animate IPME models. IPME may be run as either the server or the client. When IPME is run as a client, the custom application controls IPME model execution by dictating when events can happen. More typically, IPME is run as a server to control event execution in the custom application.

There are three phases of communication: 1) Registration, 2) Event Processing, and 3) Termination. Because IPME is the interface controller, messages are received from a custom application during Registration and at scheduled event times during Event Processing. During the Registration phase, the custom application identifies itself to IPME and configuration information is exchanged. The custom application also identifies which IPME simulation variables it wants to be informed of each time events are executed. Optionally, during the Registration phase, the custom application may request that IPME's simulation time synchronise to the host clock. This functionality allows for real-time simulation execution.

Event Processing occurs after Registration. This phase involves IPME notifying the custom application when it may execute its next event (when IPME is running as a server), plus the exchange of variables. After Event Processing, the Termination phase terminates communication between IPME and the custom application.

HLA is the other communication method supported by IPME. The HLA implementation in IPME uses the Defense Modeling and Simulation Office (DMSO) Run-Time Infrastructure (RTI) version 1.3NG version 3.2. IPME does not yet define a Federation Object Model (FOM), as is typically required by federations. It is anticipated that with further interest and participation in using IPME as a federate in a federation, at that time a FOM will be developed.

IPME has a defined Simulation Object Model (SOM), required by federates. The SOM details how IPME interacts with other federates. The object class *IPMEVariable* is defined in the SOM to represent model execution variables. An *IPMEVariable* may be one of the following data types: integer, float, or string. When IPME is configured as a federate, the user selects which

system variables to exchange with other federates, as shown in Figure 7. Variables may be read or updated by IPME.

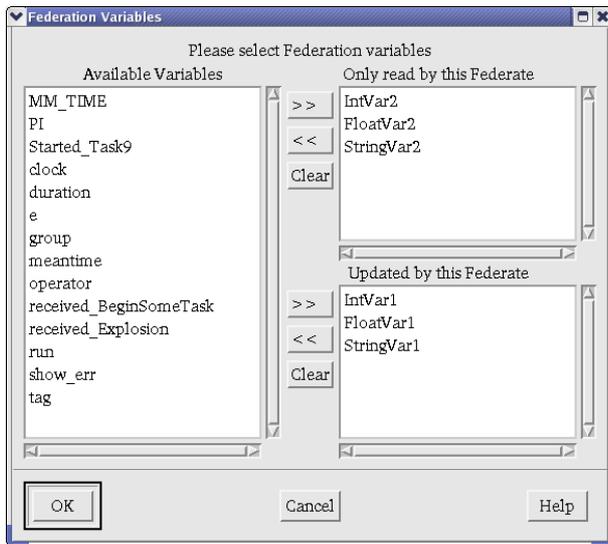


Figure 7: IPME Federation Variables Setup Dialogue

IPME sends variable values when the values are updated. When another federate sends a variable update to IPME, those values are then updated at their timestamp time during simulation execution.

IPME supports user-defined, model dependent interactions that subclass from the interaction *IPMEInteraction*, defined in the SOM. User-defined functions created in IPME may be used as interactions, with the function parameters acting as interaction parameters. Interactions provide a time-based method for inserting expressions into the IPME simulation event queue. Each interaction has a timestamp, therefore, when an interaction is received by IPME, the corresponding IPME function expressions are evaluated at that timestamp time.

Figure 8 shows the HLA Interaction Setup dialogue in IPME. All available user-defined functions in a task network model are listed in this dialogue. The user may then select which functions to send as interactions, and which to receive as interactions from other federates. Function parameters do not need to exactly match interaction parameters—extra parameters are either ignored (if the extra parameter is in a task network function) or set to a value of 0 (if the extra parameter is sent from another federate).

Although a sample Federation Execution Details (FED) file is provided, this file should be modified to specify user-defined objects and interactions. Both the *IPMEVariable* and *IPMEInteraction* classes are instantiated in the FED file to support data and interaction exchange between a particular federate and other federates.

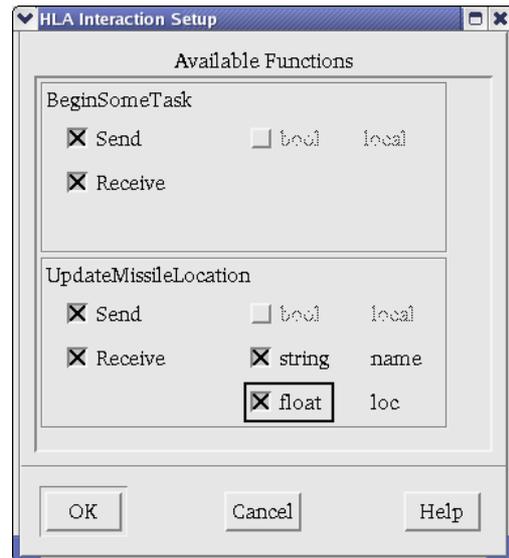


Figure 8: IPME HLA Interaction Setup Dialogue

IPME implements two time management strategies by being both a time-regulating and time-constrained federate. A time-regulating federate associates events to the federation time, determining when other federates may execute their next events (DOD 2000). IPME is also time-constrained, meaning that it expects to receive events with a timestamp (DOD 2000). The time lookahead value used by IPME can be specified by the user, and does not change during a simulation run.

## SUMMARY

This paper has focused on the three important aspects of IPME: plug-and-play models, built-in workload methodologies, and interoperability. The plug-and-play model framework unique to IPME allows the user to easily try different environments and crews with a single task network and collection of performance shaping factors. Built-in workload methodologies simplify task network model development by reducing workload calculations that might be otherwise required. IPME also supports two communications protocols, TCP/IP sockets and HLA, allowing an IPME system to take advantage of pre-existing models and simulations. IPME provides a powerful and flexible environment for model development in order to analyze human performance and stressors. Future development for IPME includes varying operator anthropometry and bundling useful, supportive client applications with IPME.

## REFERENCES

- Bierbaum, C.R.; Szabo S.M., and Aldrich T.B. 1987. "A comprehensive task analysis of the UH-60 mission with crew workload estimates and preliminary decision rules for developing a UH-60 workload prediction model." Technical Report ASI690-302-87[B], Anacapa Sciences Inc., Fort Rucker, Alabama.
- DOD (Department of Defense), Defense Modeling and Simulation Office. 2000. "High Level Architecture Run-

Time Infrastructure: RTI 1.3 – Next Generation Programmer’s Guide Version 3.2.”

Hendy, K.C.; and P. S. E. Farrell. 1997. “Implementing a Model of Human Information Processing in a Task Network Simulation Environment.” Defence and Civil Institute of Environmental Medicine, DCIEM No 97-R-71 (December).

McCracken, J.H.; and T.B. Aldrich. 1984. “Analysis of selected LHX mission functions: Implications for operator workload and system automation goals.” Technical Note ASI479-024-84. Fort Rucker, AL: Army Research Institute Aviation Research and Development Activity.

Micro Analysis and Design (MAAD). 2003. Integrated Performance Modelling Environment: User Guide, ver. 2.5.4, (May).

Micro Analysis and Design (MAAD). 2003. IPME Task Network: User Guide, ver. 2.5.4, (May).

Sarno, K.; and C.D. Wickens. 1992. “The Role of Multiple Resources in Predicting Time-Sharing Efficiency: An Evaluation of Three Workload Models in a Multiple Task Setting.” Technical Report ARL-91-3/NASA A31-91-1, NASA AMES Research Center, Moffett Field, CA.

## **AUTHOR BIOGRAPHY**

**ANNA FOWLES-WINKLER** is a Principal Software Developer at Micro Analysis and Design where she manages the IPME project. She has a Bachelor of Science in Computer Science from the University of Maryland. Ms. Fowles-Winkler is currently pursuing a Master of Arts in Linguistics from the University of Colorado. She started working on the IPME project in 1999 as a software developer, and moved to managing the project in 2001. Ms. Fowles-Winkler is interested in furthering IPME’s interoperability, including exploring cognitive modelling frameworks. Her e-mail address is : [awinkler@maad.com](mailto:awinkler@maad.com). For more information about IPME, see <http://www.maad.com/ipme>.

# SIMULATION TOOL FOR FUNCTIONAL VERIFICATION OF TTP/C-BASED SYSTEMS

Petr Grillinger and Pavel Herout  
Department of Computer Science  
University of West Bohemia  
Univerzitní 22, Plzen 30614, Czech Republic  
E-mail: pgrillin@kiv.zcu.cz

## KEYWORDS

SW tool for simulation, fault injection, brake by wire, TTP/C protocol, the C language.

## ABSTRACT

This article describes a software tool that implements C-language written simulation model of distributed embedded computer system that is interconnected by means of TTP/C protocol. The aim of simulation is to evaluate specified system's properties when used as a safety critical control system. The method that uses simulated faults to disturb system's activity was developed during the solution of the EU/IST project FIT — Fault Injection for Time Triggered Architecture (TTA). A utilization of the described simulation tool is demonstrated when evaluating the time that a TTP/C cluster that is executing a realistic brake-by-wire control application needs to stop the car while the braking process is disturbed by transient faults.

## INTRODUCTION

Verification of dependability is one of the most important steps in the design of a fault-tolerant embedded computer system. This includes testing the system's fault tolerance, i.e. its reaction to real-time disturbances from its environment and to faults inside and outside the system. Finding an appropriate verification method may save a considerable amount of time, expenses and manpower, therefore it is paid ever-growing attention. Different experimental verification methods were suggested so far, mostly using some kind of *fault injection* (FI). Fault injection can be performed on a simulation model of the system to be evaluated, on a prototype, or on the system itself. Each of these approaches has its advantages and disadvantages (flexibility and ease of implementation on the one hand, more convincing results on the other hand).

The method described here is based on digital simulation whose output is used both for qualitative and quantitative evaluation of the tested system properties. It uses a close-to-real code (C-language) describing the FT system function together with the computation dynamics and with a sub-model of the controlled environment. Thus the system behavior (even in the presence of faults) can be studied using a conventional PC workstation. The paper presents some of the results

of the presented method utilization obtained by two research groups within the EU project IST-1999-10748, Fault Injection for Time Triggered Architecture (FIT), see (<http://www.cti.ac.at/fit>).

The goal of the FIT project was an evaluation of time-triggered architecture (TTA). This is architecture of ultra-reliable embedded computer systems aimed to control cars, planes, trains, etc. (Kopetz 1997; Heiner and Thurner 1998). Special feature of TTA is fixed partitioning of node time slots on the bus, which guarantees predictable time behavior of nodes that are connected to the bus. The method of access to the bus is then TDMA (*Time Division Multiple Access*) instead of common CSMA (*Carrier Sense Multiple Access*, used e.g. by the CAN bus) and one of possible implementations of the method yields the so-called TTP/C protocol (Kopetz 1997).

TTP/C is a real-time communication protocol for interconnection of electronic modules of distributed fault-tolerant real-time systems. TTP/C is intended to meet the requirements for SAE class C automotive applications (TTP/C means Time Triggered Protocol — C class of SAE requirements). Every node connected to the bus consists of three main parts:

- communication controller, which executes the TTP/C protocol (existing C1 chip produced by TTTech company in Vienna was tested within the FIT project),
- host processor, which executes a part of an application program (and has its own I/O interface to the controlled process),
- dual-port memory, that serves as an interconnection device between the controller and the host (it is called CNI — *Computer Network Interface*).

Nodes connected to the bus form TTP/C cluster. The basic period of the bus communication activity is called a TDMA round. Within this round every node has its own slot assigned to transmit its messages.

The specified properties of TTA architecture have been evaluated experimentally using the method of fault injection (FI). The FIT project includes several kinds of FI, like hardware induced FI (UPV Valencia), software

implemented FI (SWIFI, used by TU Vienna), heavy-ion impacts on the chip (CTH Gothenburg), etc. The mentioned methods use the real HW (TTP/C evaluation cluster produced by TTTech) as an experimental environment. Another approach is to use simulation model of the communication controller and/or the whole TTP/C cluster. Within the project, two levels of simulation modeling (that differ in abstraction level) were used. The lower level is a VHDL model of communication controller itself (used by UPV Valencia, CTI Karnten). It uses abstractions like *gate*, *pin*, etc. and transient faults like *pin\_x grounded for one microsecond*, see (<http://www.cti.ac.at/fit>).

The task of our team was to *build a silicon implementation independent description (C-language based) of the TTP/C protocol* in order to have a precise and flexible description of specified TTP/C data structures and functions. In co-operation with TU Vienna and TTTech, we first built a C-language written TTP/C specification (so-called C-reference model of TTP/C protocol). This model was to simulate the behavior of TTP/C protocol at the level of message transmission and basic process interaction. Low level activity simulation (e.g. gates, pin signals, etc.) was not required — such model was already available in VHDL (due to its complexity, a VHDL description is not suitable as reference model).

However, being just a specification, this model is not executable itself. To verify its correctness and to use it for the FI purpose, we embedded the C-language written protocol specification into the C-Sim simulation environment that allows a Simula-like (coroutine) style of pseudo-parallel computing (<http://www.c-sim.zcu.cz>; Hlavička et al. 2000). The simulation model of TTP/C was verified by comparison of results with the other FI techniques — namely the SWIFI method — see (Ademaj et al. 2002).

This paper describes only in the most basic form the simulation tool and the TTP/C model itself. It concentrates more on the possible use of this kind of model in high-level verification of dependable systems. As a case study, a real-world application — Brake by wire 4 (BBW4) — is presented. It is important to notice that this application is very specific, but the simulation tool itself, the applied methods and the lower layers of the software (see table 1 in section 3) are general. This paper does not intend to propose a new original approach to simulation; rather it presents a complete utilization of one particular technique.

## C-LANGUAGE BASED SIMULATION MODELING

The discrete-time simulation modeling principle enables to run several instances (i.e. processes in simulation terminology) of TTP/C protocol with their activity "interleaved" in the global model-time with regard to local-time flow (microticks/macroticks counters) of

protocol instances. The protocol/controller instances execute the same program but they have their own data (i.e. CNI instances). Other processes (e.g. threads of a control program executed by a host processor, threads modeling controlled objects, threads of simulation experimental environment, etc.) can be added without any unintended mutual time-intrusion effects — only the speed of simulation depends on the number (and complexity) of executed processes. The computation is completely deterministic, so the obtained results are fully reproducible.

The method needs no special hardware. Typically a PC station with a C-language development environment (C++ can be used as well) can be used to develop and to perform (visualized) FI experiments. Moreover, massive non-visualized FI experiments can be automatically performed using "batch mode" execution on a powerful mainframe-like supercomputer.

The method is general enough (Hlavička et al. 2000), but we especially present its implementation that was used within EU/IST FIT project solution for an experimental validation of TTP/C based system properties (Herout et al. 2002). The lessons learned within the FIT project solution identify two main application areas of the SW implemented fault injection using C-language based simulation model:

- **TTP/C specification level** (an earliest and most abstract development phase of a class of distributed embedded computer systems architecture): The presented simulation methodology enables an evaluation of communication protocol specified properties, e.g. fail-silence property for a single TTP/C controller permanent fault, measurement of time of controller recovery that follows after transient fault, etc.
- **Application level** (the final TTP/C system development phase): Evaluation of a TTP/C based real-world application, using its C-language source codes. As the C-Sim based model enables C-language coded application SW modules to be incorporated, it is generally possible to use the C-Sim based and PC station executed development (evaluation) system instead (or as a counterpart) of a HW based TTP/C evaluation cluster. The FIT project results confirm the usefulness of utilization of the TTP/C cluster simulation model for a realistic FT application design including fault injection and a visualization of faults influence.

*This article especially concerns the second item stated above.*

## SOURCE CODE MODULAR STRUCTURE

In order to achieve a clear and consistent SW structure with a sufficient degree of portability and reusability,

we introduced a two-dimensional layering of SW modules.

The first dimension (i.e., the first way of SW layering) reflects the C-code stability, and consists of three layers (the letters assigned to individual layers are used as first and second letters in file names, so the purpose of a module can be easily identified by looking at its name):

- **L** — stable code (i.e. libraries, protocol independent code),
- **P** — protocol version dependent code,
- **A** — application-dependent code (e.g., empty application, sine wave application, brake-by-wire application etc.).

The second dimension reflects the degree of extension of C-language code properties:

- **F** — pure functionality,
- **S** — simulation, i.e., multithreading ability and time properties,
- **V** — visualization of a TTP/C cluster activity.

The second dimension determines the code portability — **F** and **S** layers are ANSI-C written (and use only ANSI-C standard libraries), so they are generally ANSI-C portable. Modules of the **V** layer are C++ Builder v.5.0 coded and assume the use of 32-bit Windows operating system.

The modular layering scheme is graphically displayed in Table 1.

Every module from the structure depicted below may export its services via an interface to the modules on the right or below. More details can be found in (Hlavička et al. 2001; <http://www.cti.ac.at/fit>).

The described software tool is not designated for wide

non-initiated public as a ready-to-use program. It should serve programmers in the C programming language to prepare their own (realistic) application by adding the C-code of an application. Some parts of the tool can serve as libraries and it is possible to use these parts as components in a visual programming environment. Moreover, there is a possibility to unify the development process for hardware and software-based evaluation clusters.

### CASE STUDY: BRAKE BY WIRE APPLICATION

The first brake-by-wire application (BBW1) was proposed in (Lönn 2001) to enable fault injection testing with transient faults, which can attack arbitrary volatile information at the TTP/C level, especially controller's local data and CNI data. An extended version of this application, that is closer to reality because it simulates four wheels instead of just one (BBW4) has been developed later and this later application is presented here.

Both BBW application cores were developed by Volvo Technological Development Corporation in Matlab/Simulink for the FIT project and then converted to ANSI C source files.

The BBW application consists of wheel simulation with ABS controller and vehicle simulation. The current four-wheel BBW simulation cannot run on the HW cluster, because it uses floating point variables and operations. It is possible replace the floating point arithmetic routines by fixed point routines — this means however an additional overhead and the development time of such application would be longer. This shows an advantage of simulation method that has no restrictions in this direction.

### The Structure and the Principles of Four-Wheel BBW Simulation

The four-wheel application is very close to reality (the model of the car provides nine degrees of freedom). All wheels are simulated, the external vehicle model

		<i>j</i> → code properties extension		
		<b>F</b>	<b>S</b>	<b>V</b>
↓ code instability	<b>L</b>	ANSI-C library	C-Sim kernel	C++ Builder library
	<b>P</b>	FP <sub>x</sub>	SP <sub>x</sub>	VP <sub>x</sub>
	<b>A</b>	FA <sub>xy</sub>	SA <sub>xy</sub> (main)	VA <sub>xy</sub> (main)
		<i>ANSI-C portable</i> ←		→ <i>OS dependent</i>
				↓ <i>executable</i>

x-axis — TTP/C version, here assumed 1.0, i.e., x = 1  
y-axis — cluster configuration + application

Table 1: Source code modular structure

simulates vehicle reactions and the distribution unit redistributes brake force when a brake failure is detected. Figure 1 schematically explains the interconnection of individual modules.

**Individual parts of the simulation:**

- a) **Vehicle simulation** — the vehicle simulation is based on the behavior of the Volvo V70 car and except when moving with low speeds its behavior is practically the same. The vehicle simulation runs with a period of 500 microseconds. The simulation uses one parameter (initial vehicle speed) at the beginning of the simulation and four input variables in each simulation step. These variables are: *Brake\_Force* vector, *Traction\_Torque* vector, *Steering\_Angle* vector and *Adhesion* vector. All vectors have four members, one for each wheel. The *Brake\_Force* is given in *N* units, *Traction\_Torque* in *Nm* units, the *Steer\_Angle* in radians and the *Adhesion* is a non-dimensional value from 0.0 (ice) to 1.0 (dry tarmac). These four vectors are used to control the experiment during the simulation. The outputs from vehicle simulation are: *Wheel\_Speed* vector and *Vehicle\_Speed* vector (in heading direction, *m/s*). Heading angle in radians and global x-axis and y-axis position (*m*). The ABS controller uses the output values to compute the final *Brake\_Force*. All output and input values are 8 byte floating-point variables or vectors.
- b) **Distribution unit** — the distribution unit is introduced to reduce the effect of brake faults. This unit uses as its input the requested *Brake\_Torque* sent by the pedal node and *Fault\_Vector*, which is derived from TTP/C model (using membership

service provided by the TTP/C protocol complemented at application level with comparison of messages from both replicas). Each bit in this array corresponds to one wheel — a value of 1 signals that the brake system at this wheel failed. The distribution unit computes the output *Brake\_Force* from the requested *Brake\_Torque*, the multiplier factor (set for each wheel in the actual situation), and from the fact whether the wheel is front or rear (the brake force distribution is 65% for front wheels and 35% for rear wheels).

- c) **ABS controller** — The ABS controller does not compute the final brake force, it only uses the input brake force obtained from the Distribution unit and then decides whether the *Brake\_Force* will be zero or the input value. Its decision depends on the second and the third input values — the *Wheel\_Speed* and the *Vehicle\_Speed*; the difference between these two values starts or stops the function of the ABS controller. These two modules (distribution and ABS controller) run with period of 4 milliseconds, it means that these simulations have eight times longer period than the vehicle simulation. This is convenient for our TTP/C simulation because the cluster cycle has the same length, which makes scheduling easier.

**TTP/C Model of Four-Wheel Brake by Wire**

The simulation is run on 10 nodes divided into two categories. At every wheel brake are two replicated controlling nodes (8 nodes altogether), and at the brake pedal different two nodes measure the pedal’s position. The vehicle motion, brake force distribution and ABS controller status is updated externally in a separate thread. This structure of the application was obtained

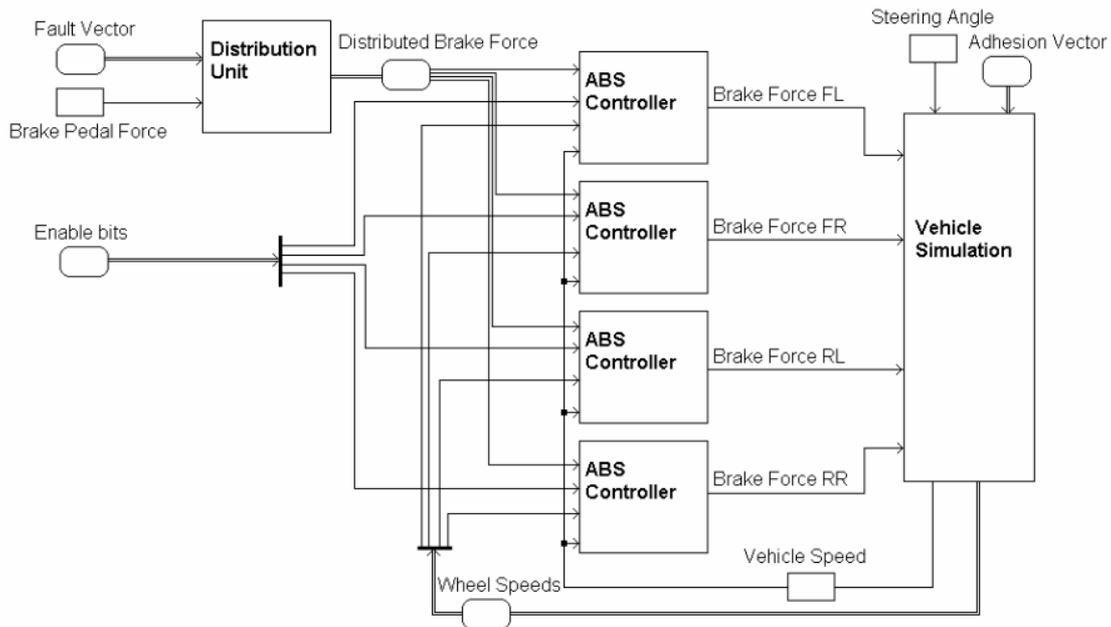


Figure 1: Detailed Structure of Four-Wheel BBW Simulation

from Volvo Company and it is difficult to divide into four (or eight) parts. The distribution unit and the ABS controller routines can run on every wheel node. This division means that the simulation status (internal variables of the model) is updated only at one place in the simulation, but the function that computes the *Brake\_Force* for a wheel is distributed among all (eight) wheel nodes.

The application task of a wheel node (executed in real-world directly by the node's host computer) can be divided into four steps:

- The first step is reading of the values of *Brake\_Force* sent by other wheel nodes during the last TDMA round and their comparison. The node compares corresponding values sent by wheel nodes of other wheels. This comparison is used to set up the fault vector for wheel node — i.e. its own view of the cluster situation.
- The second step is reading the *Brake\_Torque* sent by two pedal nodes. In case that the read values are identical, their value is used for the new *Brake\_Force* computing function; otherwise brake force equal to 0 is used.
- The third step is *Brake\_Force* computing. In this phase the node uses the functions developed by Volvo. These functions use *Brake\_Torque*, their individual image of *Fault\_Vector*, their

*Wheel\_Speed* and *Vehicle\_Speed* as an input.

- In the last step the application writes the actual value of *Brake\_Force* into the CNI for the next transmission slot.

### Cluster and Message Structure

As was mentioned before, the model runs 10 nodes coupled into five pairs. For better understanding of the structure we can look at figure 2.

We divide the cluster cycle into two TDMA rounds. In the first TDMA round all nodes send N-frames (frames with application data content) on the bus; in the second round all nodes send I-frames (these frames contain no application data, they contain current cluster state that can be used for reintegration). Each sending slot is 200  $\mu$ s long.

Multiple I-frames help the nodes to reintegrate after a breakdown in situation when multiple nodes break down as a consequence of a stream of faults. We could send I-frames by pedal nodes or by wheel nodes only. However, it is quite dangerous to send I-frames only by the pedal nodes, because both pedal nodes could fall out and then no node could reintegrate into the cluster. The second option is irrelevant due to a different reason: the danger that there will be no running node that will send I-frames is very low, but we need much more information about how the brakes really work, than the information about the brake pedal status. In a real

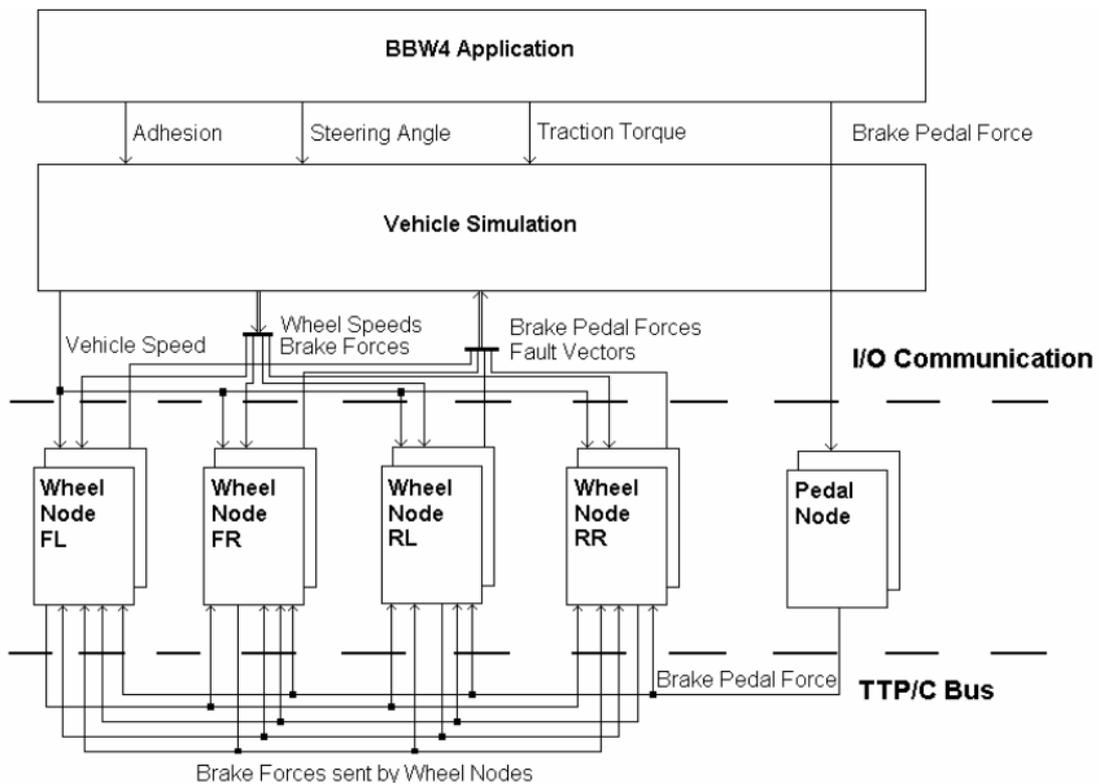


Figure 2: BBW4 TTP/C Cluster Structure and Data Flow

application the brake pedal value change will be slow if we know that the ABS controller period is only 4 milliseconds long, so there is no reason to refrain from sending of I-frames by the pedals nodes.

### Fault Detection Mechanisms

Fault detection is the main task of safety critical applications when faults are introduced into the system. The application has to be fault-tolerant or at least be able to minimize the influence of the fault. Our application is the second case, because when we use only two nodes for each wheel we cannot develop a fault-tolerant application for each wheel, but we can detect a fault by comparing outputs from wheel nodes, and correct the fault consequences by changing the application parameters using the distribution unit.

*The fault detection mechanism in our application is used twice.*

- An application task uses the data sent on TTP/C bus. The *Fault\_Vector* is used in the distribution unit to minimize the influence of a fault.
- In the vehicle simulation thread that simulates actuators on each wheel that receive the value of *Brake\_Force* from nodes on the wheel sent by I/O communication from the nodes. This unit needs to have two identical values on its input; otherwise it cannot decide which value to use and performs no action.

#### *Fault Detection in Wheel Node Task.*

Three fault detection routines are executed during the wheel node task. The first serves to detect a fault in the actual node or its replica. The wheel node reads messages sent by both nodes from CNI Message Area and compares them. The result of the comparison is *equal* or *not equal*. When the messages are different, the wheel node increases total fault counter together with continuous fault counter. Total fault counter is information about the wheel stability only, but the continuous fault counter is used to limit future failures — when the continuous fault counter reaches 5 the node automatically restarts itself (then a self-test can be performed). These counters can be also incremented in one other case — when the wheel node cannot read valid data from its message stored in CNI.

The second test is used to create node's own overview of the whole cluster situation. The node tries to detect wheels that do not work properly, i.e. both nodes are broken down or the nodes send different data or send invalid data. The obtained information is used for *Fault\_Vector* set-up — each node has its own copy of this vector.

The third test is used for *Brake\_Torque (BT)* signal validation. The wheel node compares the values sent by

Pedal nodes and sets a new *BT* value if they are equal or sets the *BT* value to 0 if they are different.

### Types of Tests

The tests have to show that it would be possible to stop the vehicle in spite of faults injected into TTP/C nodes or the bus and in the best case the car would not slip from the straight direction. BBW4 minimizes the influence of faults by a brake force distribution mechanism. The use of this mechanism is enabled by the TTP/C cluster services — membership service, node replication, channel (and frame) replication and time synchronization.

A fault injection experiment can be organized in several ways:

- *White-box FI* — the faults are targeted into exact locations and/or injected at exact points in time. Such experiments are used often to verify a particular hypothesis (e.g. clock synchronization).
- *Black-box FI* — the faults are injected at random intervals and hit random targets in the modeled system. This experiment organization is suitable to prove general resistance to FI. A fault model that is close to reality is essential.

In our case a variant of black-box FI has been applied primarily — pseudo-random stochastic FI.

#### *Stochastic Fault Injection.*

This method utilizes streams of faults (mostly Gaussian or Poisson streams) that are controlled by a pseudo-random number generator. Our basic fault model is a short *burst of single-bit flips* (sequence of several bit-sized faults) that is repeated within a stream with mean period larger than the duration of a single burst. This fault model simulates reoccurring transient problems of the chosen node (e.g. EMI effects) or transient malfunction of node's sensors, depending on the target of FI.

### Selected results

A simple fault injection experiment was chosen for demonstration. It was stated that the brake force distribution mechanism should be able to minimize unwanted effects of a brake failure. To work properly the mechanism requires an accurate and up-to-date fault vector. Braking force to be applied to individual wheels is determined for the current communication period using fault vector from previous period. *This means that when the fault vector changes frequently the force distribution mechanism's performance can degrade significantly and the braking distance may be longer than when a permanent brake failure occurs.* This theory can be easily verified by the model when we run experiments with different frequencies of faults. Very high frequency of faults can be used to simulate a

permanent failure (the period must be shorter than the fastest node recovery time, which is 1 TDMA round).

Table 2 below summarizes the output received from braking trajectory measurement for different fault injection frequencies (the fault is simultaneous shutdown of both replicas at the rear left wheel). Observations of the car behavior in the presence of faults have revealed that the distribution mechanism over-compensates the failed brake in time, so the vehicle starts drifting in the opposite direction (then the effect is reversed, so the car moves from left to right and back). This makes measurement of current Y-Axis position almost useless, so the table below lists only the maximum deviation in the Y-axis (max. lateral movement).

The intensity of fault injection is given as a ratio of fault-injected TDMA rounds to total number of TDMA rounds. For example “1:10” ratio means 1 fault in 10 TDMA rounds. The special value “0:-” means that no FI was performed and the ratio 1:1 means that a fault is injected in every TDMA round (this is a simple model of permanent failure).

FI Ratio [faulty : total]	Traveled distance [m]	Max. lateral movement [m]
0 : -	56.67	0.00
1 : 10	60.53	0.41
1 : 5	75.57	0.38
1 : 3	79.65	0.38
1 : 2	128.64	2.45
1 : 1	89.73	1.29

Table 2: Braking results for different FI frequencies

The first row in table 2 contains measurement of the braking process under optimal conditions, i.e. no fault injection was applied. This provides us with reference values, so we can compare the values retrieved with different fault injection settings. The lower rows in the table gradually increase fault intensity and we can see that the measured travelled distance and lateral movement increase accordingly — except for the FI ratio “1:2”. In this case the overall braking distance and maximum lateral deviation is significantly larger than for other FI ratios (even for higher fault frequencies). This behaviour is caused by the brake-force distribution mechanism that shows a possibly dangerous instability to certain fault frequencies (the algorithm assumes that current state will be valid for the next communication period and this FI frequency invalidates the assumption). This confirms the previously stated hypothesis (in italics at the beginning of this section) and gives us a rough worst-case braking scenario.

A snapshot of the BBW4 application used to gather the presented results (its visualization) is displayed in figure

3. The application can be easily altered to inject different kind of faults, e.g. to inject into different combination of nodes or to inject only during certain protocol execution phase.

## ADVANTAGES AND POSSIBLE DEPLOYMENT

The method of simulation-based verification of safety critical systems is not new and many sources deal with simulation modeling at different architectural levels. What makes our method unique is its wide usability:

- *Functional reference model:* the C-language model of the system’s interconnection protocol can be used from the early stages of development as an exact definition of the protocol (e.g. more exact than a written semi-formal specification). Parts of the code can be verified separately. This enables to perform various experiments in early stages of new protocol version preparation. In later stages the exact C-model serves as reference for any HW based implementation.
- *Executable model:* The functional protocol reference model can be embedded within a simulation environment (in our case the C-Sim library) to provide the basis for building executable models of the whole system including applications. An application can be developed and debugged relatively easily using the model even when the modeled device is not yet available. Using a wide set of simulation applications (specially designed for testing, i.e. not necessary real-world applications), it is possible to generalize the obtained results. We can easily implement different fault tolerant mechanisms (active replication, TMR, repeated execution, etc.) and evaluate the effectiveness of these different approaches.
- *Real-world applications:* The ANSI standard of C language, which is used, enables to link the model with any C-written application. This is important because today, most industrial applications (for embedded computers) are written in C language. There are two possible ways to utilize this: We can take an existing application and verify it under arbitrarily severe conditions (unlikely to happen in the normal operating environment). The second utilization is to develop a completely new application using the model and after thorough verification and debugging port this application into the real device. This porting requires usually only minor modifications to the source code. The simplicity of the porting process (resulting from the same used programming language) reduces the probability that a fault in the application will be introduced during the transformation.
- *Portability and performance:* The simulation can run on any system that provides a C language compiler (this means almost every computer

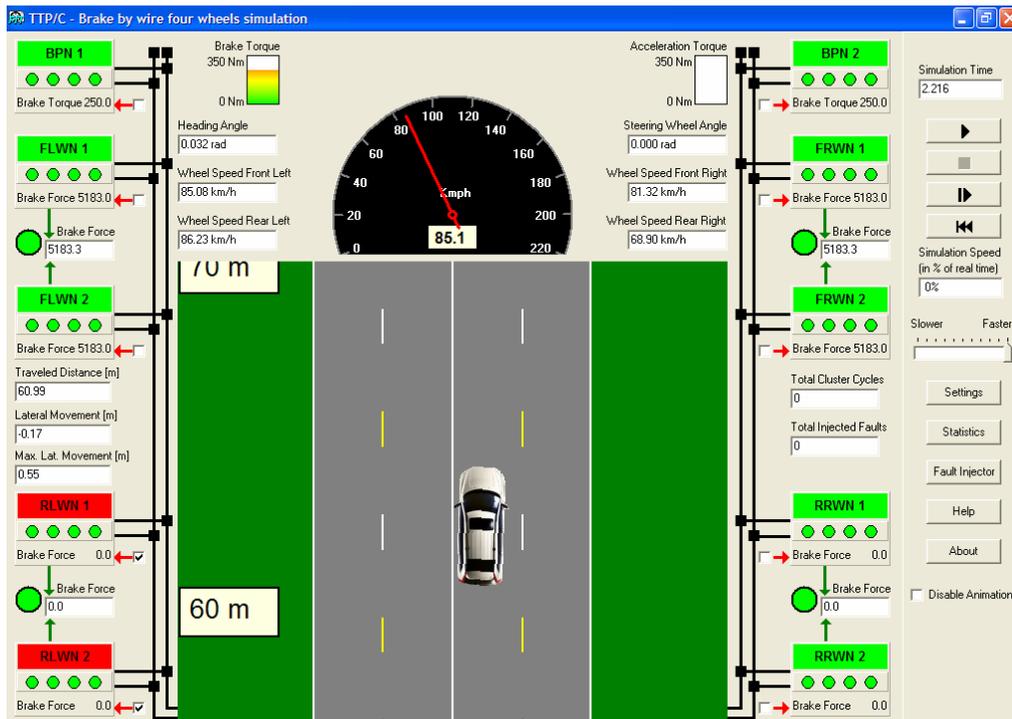


Figure 3: Captured screen from the visualized BBW4 application

system). The experiments were done on several platforms, including: PC with Linux, Windows NT and Windows XP, supercomputer Digital AlphaServer 8400 5/300 with Digital UNIX V4.0E. The performance of the model is determined mainly by the applications design (i.e. number of nodes, computation complexity). For example, on the a 1.4GHz PC the synthetic application sine wave (4 nodes) executes approximately four times faster than real-time and the real-world application BBW4 with 10 nodes executes at half speed of real-time.

- *Analyzability of results:* Because C-Sim based simulation is fully deterministic, we are able to analyze discovered problems at arbitrary level of detail. This enables to pinpoint to source of a problem exactly, i.e. to find whether the problem is caused by a faulty implementation or by a flaw in the system design.
- *Abstraction level* — Limits possible fault injection targets. In our case we are able to perform FI into any memory field of TTP/C that is defined in the official specification, into transmitted messages and into any application defined field. It is impossible to influence internal registers (not covered by the specification) and code.
- *Fault Nature* — only memory based faults are possible, moreover permanent faults are difficult to simulate.

- *Set-up difficulty* — depends on the application and availability of a ready-to-use solution (as the BBW4 made by Volvo). To build a model from scratch is very time-consuming.
- *Reusability* — different for all parts of the model. The C-Sim tool is highly reusable and the TTP/C C reference model can be without any difficulty used for any other application. The real-world application is bound to a particular purpose and cannot be used anywhere else except the real device.

## CONCLUSIONS

The presented case study (the BBW4 application) shows many of the benefits that a simulation can offer. The most obvious one is that a HW implementation would not even be possible in advance. Moreover, the simulation tool enables us to add a visual user interface to the model as well as a fault injection capability. Such an interface can be used either for more sophisticated experiments or for demonstration purposes, as it interactively displays the current state of simulation.

It is obvious that evaluation and verification based on simulation can never provide complete assurance of the safety of the modeled system or application. In reality this cannot be guaranteed by any single verification method. The advantages of our approach are clearly stated in section *Advantages and Possible Deployment*.

Usefulness of the described approach and mainly the mentioned case-study (BBW4) was proven within the EU FIT project (Final report of the FIT project 2002).

The model of a newer version of the TTP/C protocol (the C2 chip) is currently developed and in the future we plan to adapt the testing tools to this new model. Also the current BBW4 application should be extended to allow a wider range of experiments.

## ACKNOWLEDGMENT

The research was in part supported by a grant of 5<sup>th</sup> Framework Program Information Societies Technology: IST-1999-10748 Fault Injection for Time Triggered Architecture (FIT). Theoretical part of work was supported by the Ministry of Education of the Czech Republic, project no. MSM-235200005: Information systems and Technologies.

## REFERENCES

- Ademaj, A.; P. Grillinger; P. Herout; and J. Hlavička. 2002. "Fault Tolerance Evaluation using two Software Implemented Fault Injection Methods". In *IEEE International On-Line Testing Workshop (IOLTW 2002)*, Isle of Bendor, France, July 2002, pp. 21-25.
- Final report of the FIT project, IST-1999-10748.
- Grillinger, P. and S. Racek. 2002. "Transient faults robustness evaluation of safety critical systems using simulation". In *Baltic Electronic Conference (BEC 2002)*, Tallinn, Estonia, October 2002.
- Heiner, G. and T. Thurner. 1998. "Time-triggered architecture for safety-related distributed real-time systems in transportation systems". In *Proceedings of FTCS-28*. Munich, Germany, pp. 402-407.
- Hlavička, J.; S. Racek; and P. Herout. 2001. "Modeling a Fault-Tolerant Multiprocessor System". In *IEEE Conference EUROCON 2001*. Bratislava (Slovakia), July 2001, pp. 544-547.
- Hlavička, J.; S. Racek; and P. Herout. 2000. "Evaluation of process controller fault tolerance using simulation". In *Simulation Practice and Theory*. Volume 7, Issue 8, 15th March 2000, pp. 769-790.
- Herout, P.; S. Racek; and J. Hlavička. 2002. "Model-based dependability evaluation method for TTP/C based systems". In *Proceedings of European Dependable Computing Conference (EDCC-4)*. Toulouse, France, October 2002, pp. 271-282.
- Krejzek, T. 2002. *Verification of Application Reliability in TTA*. Master Thesis. Czech Technical University in Prague, June 2002.
- Kopetz, H. 1997. *Real-Time Systems, Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997
- Laprie, J.C. 1992. *Dependability: Basic concepts and terminology*. Springer-Verlag Wien New York, 1992, 265 pp.
- Lönn, H. 2001. *Brake by wire status report*. Status report of the EU FIT project, Prague.
- Manzone, A. et al. 2001. "Fault tolerant automotive systems: An overview". In *Proceedings of 7th Int'l On-Line Testing Workshop*. Taormina, Italy, July 2001, pp. 117-121.
- <http://www.c-sim.zcu.cz> — Home pages of the C-Sim simulation tool.
- <http://www.cti.ac.at/fit> — Home pages of EU project Fault Injection for TTA (FIT).

## AUTHOR BIOGRAPHIES

**PETR GRILLINGER** was born in Czech Republic and went to the University of West Bohemia in Pilsen where he studied computer engineering and obtained his degree in 2001. He finished his master thesis as a part of the EU project FIT and now continues his PhD study of simulation based fault injection techniques at the same university. His e-mail is: [pgrillin@kiv.zcu.cz](mailto:pgrillin@kiv.zcu.cz) and his web-page can be found at the address <http://www.kiv.zcu.cz/~pgrillin>

**PAVEL HEROUT** was born in Czech Republic. He graduated in 1985 at the Institute of Technology in Pilsen in specialization Electronic computers. In 1999 he defended his PhD thesis in computer science. He works as a teacher at the University of West Bohemia in Pilsen and delivers lectures and seminars in subjects Object Oriented Programming, Programming in the C Language and Desktop Publishing. His professional interests are programming languages, simulations and fault-tolerant computing. His e-mail address is: [herout@kiv.zcu.cz](mailto:herout@kiv.zcu.cz) and his web-page can be found at <http://www.kiv.zcu.cz/~herout>

# A DECOUPLED FEDERATE ARCHITECTURE FOR DISTRIBUTED SIMULATION CLONING

Dan CHEN<sup>1</sup>, Stephen John TURNER<sup>2</sup>, Boon Ping GAN<sup>1</sup>, Wentong CAI<sup>2</sup>, Junhu WEI<sup>2</sup>

<sup>1</sup> Singapore Institute of Manufacturing Technology  
Singapore 638075

E-mail: {dchen, bpgan}@simtech.a-star.edu.sg

<sup>2</sup> School of Computer Engineering  
Nanyang Technological University  
Singapore 639798

E-mail: {assjturner, aswtcai, asjhwei}@ntu.edu.sg

## KEYWORDS

High Level Architecture, Runtime Infrastructure, distributed simulation cloning, decoupled federate architecture, fault tolerance.

## ABSTRACT

Distributed simulation cloning technology is designed to perform “what-if” analysis of existing High Level Architecture (HLA) based distributed simulations. The technology aims to enable the examination of alternative scenarios concurrently within the same simulation execution session. State saving and recovery are necessary for cloning a federate at runtime. However it is very difficult to have a generic state manipulation mechanism for any existing federate, as these can be developed independently and freely. The correctness of replicating a running federate significantly depends on the Runtime Infrastructure (RTI) software. The distributed simulation also needs fault tolerance to provide robustness at runtime. This paper proposes a decoupled federate architecture to address the above issues. A normal federate is decoupled into two processes, which execute the simulation model (virtual federate) and the local RTI component (physical federate) respectively. The decoupled approach interlinks the two processes together via Inter-Process Communication. The virtual federate interacts with the RTI through the standard RTI service interface supported by a customized library. The decoupled architecture ensures the correct replication of federates and facilitates fault tolerance at the RTI level. At the same time, it provides user transparency and reusability to existing federate codes. Benchmark experiments have been performed to study the extra overhead incurred by the decoupled federate architecture against the normal federate. The encouraging experimental results indicate that the proposed approach has a performance close to the normal one in terms of latency and time synchronization.

## 1. INTRODUCTION

Distributed simulation is an important technology that facilitates simulation programs executing in a distributed environment. Geographically distributed simulation models can be linked together to construct a large-scale simulation federation. Distributed simulation technology has a variety of applications, one of which is supply-chain simulation. It meets the pressing need of simulating a supply-chain, as this often involves multiple companies across enterprise boundaries and simulation models that are developed independently (Gan et al. 2000; Turner et al. 2001).

The High Level Architecture (HLA) defines the rules and specifications to support reusability and interoperability amongst the simulation federates. The Runtime Infrastructure (RTI) software supports and synchronizes the interactions amongst different federates conforming to the standard HLA specifications (Dahmann et al. 1998). HLA-based distributed simulation provides interoperability and reusability of the independent simulation federates. However in the context of a traditional distributed simulation, one simulation session can only yield one single set of results for analysis. To perform “what-if” analysis, one has to repeat the execution of the simulation to examine alternative scenarios or decision strategies using different rules and parameters. Therefore the simulation analyst may choose some best solutions based on all the possible results. Basically it is a time-consuming and onerous task in which a lot of computation is repeated unnecessarily.

During the simulation, a federate will meet some points (decision points) at which there occurs a critical change of system states, and it is faced with different choices to proceed (Chen et al. 2003a). Instead of executing all the choices one by one in a linear way, the simulation cloning approach offers users the flexibility to examine these different choices concurrently. At a

predefined decision point, cloning of a federate may be triggered at runtime, following which the federate can replicate itself into multiple clones to explore different possibilities (Chen et al. 2003a). Each clone explores one particular path together with its partner clones spawned from the other federates in the original scenario. Thus, users are able to analyze multiple alternative results concurrently using the same simulation models within a single simulation run.

However it is challenging work to ensure correct and efficient simulation cloning especially in the context of distributed simulation. For example, it needs state saving and recovery at both the simulation model level and the RTI level, rather than simply starting another federate instance. A distributed control mechanism is needed to coordinate the federates within the same distributed simulation session. Furthermore, the correctness of cloning significantly depends on the platform and RTI software the simulation federates use. As the local RTI component is not designed to be replicated, direct cloning of a federate can lead to unpredictable and uncontrollable failure at the RTI level. Thus it requires us to design a reliable and correct federate cloning approach.

One of the key benefits of HLA-based simulation is reusability (Dahmann et al. 1998), which raises another critical issue of reusing the existing code of user federates while adopting simulation cloning technology. Considering the complexity and variety of simulation models, it is difficult to have a generic cloning solution that keeps the consistency of any simulation federate while cloning. However a middleware approach makes it possible to monitor the system state of a federate at the RTI level (Chen et al. 2003a).

Moreover simulation federates running at different locations are liable to failure, and the failure of one federate can lead to the crash of the overall distributed simulation. Cloning more and more federates inside one single federation may increase the risk of such failure steadily, thus we need to investigate the fault tolerance issue in simulation cloning and apply it to the distributed simulation technology.

This paper introduces the idea of decoupling the local RTI component from a normal HLA federate. Basically a normal federate contains the simulation model and the local RTI component. The proposed approach separates these two modules into two independent processes, namely a virtual federate and a physical federate. The virtual federate inherits the code of the original simulation federate while associating a “virtual” RTI component with it, which still provides the simulation model with a standard interface of RTI services. The real RTI services

are accessed through the physical federate working in the background. An Inter-Process Communication channel bridges the two processes together into a simulator in a general sense. All the RTI calls employed by a virtual federate call services via the corresponding physical federate. This approach ensures the intactness of existing simulation federates. Cloning a federate means replicating multiple virtual federates and starting new physical federates with recovered system states at runtime. As the virtual federate contains no real RTI component, the decoupled approach avoids the risks incurred by copying a running federate.

The decoupled architecture isolates the failure of local RTI components from the simulation federates. In the case of an RTI component crash, a new federate or even a new federation can be resumed according to the stored states at the RTI level. One does not need to start the whole simulation from scratch, thus the decoupled approach provides fault tolerance in this sense. All these advantages can be achieved without interrupting the execution of the user’s simulation.

To investigate the overhead incurred by the decoupled approach, this paper presents a set of benchmark experiments on latency and synchronization performance. Results are reported and compared between the decoupled federates and normal federates. The experimental results indicate a promising performance of the decoupled federates in both benchmarks.

The rest of this paper is organized as follows: Section 2 outlines the distributed simulation cloning technology and related work. Section 3 discusses the decoupled approach in detail for both design and implementation. Section 4 describes the benchmark experiments and analyzes the results. In section 5, we conclude with a summary and proposals on future works.

## **2. DISTRIBUTED SIMULATION CLONING TECHNOLOGY**

### **2.1 Related Work**

Hybinette and Fujimoto first employed the simulation cloning technology as a concurrent evaluation mechanism, in the context of parallel simulation (Hybinette and Fujimoto 2001). The motivation for this technique was to develop a parallel model that supports an efficient, simple, and effective way to evaluate and compare alternative scenarios. The method was targeted for parallel discrete event simulators that provided the simulation application developer with a logical process (LP) execution model.

Schulze et al introduced a cloning approach to extend the flexibility of system composition to run-time (Schulze

et al. 2000). Their approach included the parallel management of different time axes in order to provide forecast functionality. Internal cloning and external cloning techniques were suggested to clone the federates at run-time.

As our design targets the users who may have their own existing complex simulation models, we have the additional aim to provide reusability and transparency while enabling simulation cloning. Our research and discussion are based on HLA-compliant distributed simulations. Providing easy utilization and deployment is another major concern. Distributed simulation cloning technology should be a much more powerful and flexible decision support tool than traditional “linear” simulation. Our approaches focus on the control of a large-scale distributed simulation using the cloning technology.

## 2.2 Distributed Simulation Cloning

Simulation cloning technology involves research issues such as trigger conditions, cloning operation, distributed coordination, state saving and recovery, scenario management and interactive control, etc. We define some terms in distributed simulation cloning as follows.

Cloning of a distributed simulation may happen at some critical points that are defined by a simulation analyst. At one of those points, a federate may face different choices that perform alternative actions. These points are known as **decision points**, which comprise trigger conditions and candidate actions for a federate to perform. A decision point usually represents the location in the execution path where the states of the system start to diverge in a cloning-enabled simulation. From the decision point onwards, a federate may spawn multiple executions to exploit alternative scenarios concurrently.

A federate is said to perform **active cloning** if it makes clones on its own initiative. As there exist multiple interoperating federates in distributed simulations, when one federate splits into different executions, the partners who interact with this federate may have to spawn clones as a result of the active cloning, thus **passive cloning** happens. The clones created from the same root federate are referred to as **sibling clones**. Each clone is an independent simulation federate, and it cooperates with some clones of other federates to form an independent simulation scenario. Those clones within the same scenario are known as **partners**.

In order to save computation, our proposed approach merely requires cloning the federates whose states will change at a decision point and keeping other federates intact. Thus the simulation is replicated incrementally;

such an **incremental cloning** approach shares computation between federates in multiple scenarios. Although new scenarios have been created due to the active cloning, a clone is capable of executing in multiple scenarios, and these are known as **shared clones**.

When a federate is cloned, we can create multiple federations to meet the demand of executing alternative scenarios or generate new federates within the original federation without intervening in the execution of any other scenario. The former approach is called a **Multiple-federation Solution**, and the latter approach a **Single-federation Solution**. A single-federation solution offers advantages in cloning control and cloning sharing and is adopted for our research. In order to manage concurrent scenarios within a single federation, we propose to use the Data Distribution Management (DDM) (Morse and Petty 2001) mechanism to partition scenarios. To provide reusability to existing simulation federates, a middleware approach is adopted to hide the implementation of any cloning related modules. Thus transparency to simulation federates is achieved in distributed simulation cloning.

## 3. CLONING FEDERATES

### 3.1 Problems in Cloning Federates

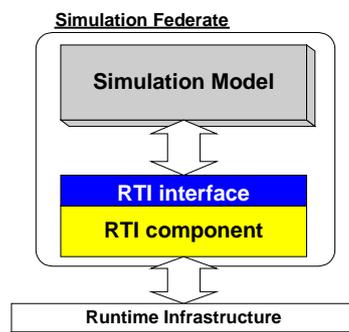


Figure 1: Abstract Model of a Simulation Federate

A normal simulation federate can be viewed as an integrated program consisting of a simulation model and local RTI component, as shown in Figure 1. As mentioned above, active cloning of a federate occurs at a decision point to enable different candidate actions to be performed. “Cloning” implies that the new clones of one particular federate should have the same features and states as the original federate both at the RTI level and at the simulation model level. This is to ensure the simulation state consistency. For example, at the RTI level, clones must have subscribed to the same object classes and registered the same object instances etc. At the simulation model level, the clones should have the same program structure, data structures, objects and variables; all these program entities should have identical states.

Immediately after the active cloning, the clones will be given some particular parameters or routines to execute in different paths.

One possible solution is to introduce a state saving and recovery mechanism to the simulation federates, allowing the simulation federate to store snapshots of all the system states. When cloning occurs, new federate instances are started and initialized with stored states. However, users model their simulations in a totally free manner. It is unlikely that a generic state saving and recovery mechanism can be provided that will be suitable for any simulation federate. Even given such a mechanism, it is unlikely that all simulation developers will use the same standard package to model their simulations. Without the ability to customize the user's simulation code, it is almost impossible to make snapshots of all system states of any federate. Furthermore, the principle of reusing existing federate code increases the difficulty of this task. On the other hand, the standard HLA specification makes it relatively easy to intercept the system states at the RTI level. Using a middleware approach, one may save and recover the RTI states while enabling transparency.

However some operating systems enable the user to duplicate a running process. In UNIX, some system calls such as **fork** can create a new process that is an exact copy of the calling process (Stevens 1999). This suggests the possibility of cloning a federate at runtime using such a process duplication mechanism. Thus the correctness of cloning depends on the platform and RTI software that the simulation federates adopt. However, the local RTI component is not designed to be duplicated, thus forking a federate can lead to unpredictable and uncontrollable failure at the RTI level. The failure of the local RTI component prevents the simulation execution from proceeding correctly.

In HLA-based distributed simulation, the crash of one federate can result in the failure of the overall simulation federation. As more and more clones will participate in the existing federation as a result of simulation cloning, fault tolerance becomes another important concern. From the above discussion, we can see that the simulation model and the local RTI component have very different characteristics. Therefore, it seems that we can make a distinction between these two modules for cloning a federate.

### 3.2 Decoupled Federate Architecture

To tackle the problems involved in replicating running federates, this section introduces the decoupled federate architecture to separate the simulation model from the

local RTI component. The design and implementation of this approach will be covered in detail.

#### 3.2.1 Virtual Federate and Physical Federate

In the context of the decoupled architecture, a federate's simulation model is decoupled from the local RTI component. A virtual federate is built up with the same code as the original federate. As HLA only defines the standard interface of RTI services, we are able to substitute the original RTI software with our customized RTI++ library without altering the semantics of RTI services (Chen et al. 2003a). Figure 2(B) gives the abstract model of the virtual federate. Compared with the original federate model illustrated in Figure 1, the only difference is in the module below the RTI interface, which remains transparent to the simulation user.

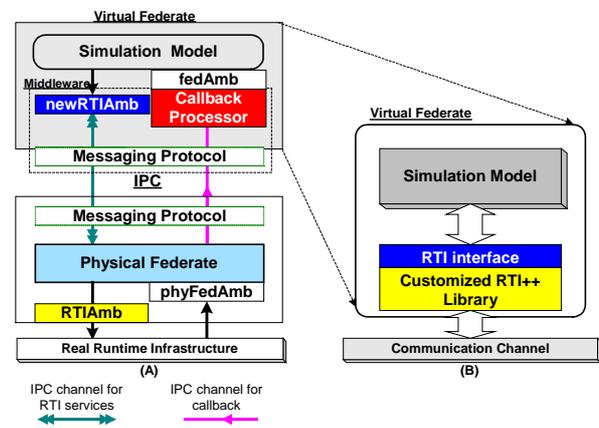


Figure 2: Decoupled Federate

A physical federate is specially designed as shown in Figure 2(A). The physical federate associates itself with a real local RTI component. Physical federates interact with each other via a common RTI. Both virtual federates and physical federates operate as independent processes. Reliable Inter-Process Communication (IPC) or other out-of-band communication mechanism bridges the two entities into a simulator in a general sense (Stevens 1998). Using the decoupled approach, cloning of a simulation federate can be done by forking the virtual federate process and starting an additional physical federate instance with restored system state at the RTI level.

All the components inside the dashed rectangle form a **Middleware** module between the simulation model and the IPC. Within the virtual federate, the **newRTIAmb** contains customized libraries while presenting the standard RTI services and related helpers to the simulation model. This module is also designed to contain all other management modules for cloning purpose (Chen et al.

03b). The **fedAmb** serves as a common callback to the user federate, which is freely designed by the user and independent of the decoupled approach. The **newRTIAmb** handles the user's RTI service calls by converting the method together with the associated parameters into IPC messages via the **Messaging Protocol**. The protocol defines a mapping between an IPC message type and the RTI method it represents. For example, an RTI\_UPDATE message indicates that the virtual federate has invoked the RTI method *updateAttributeValues()*. The IPC conveys these messages to the physical federate for processing in a FIFO manner immediately.

The physical federate is designed to convert an RTI call message generated from the virtual federate into the corresponding RTI call through its own messaging protocol layer. The **RTIAmb** module executes any RTI service initiated by the simulation model prior to passing the returned value to the IPC. The **phyFedAmb** serves as the callback module of the physical federate to respond to the invocation issued by the real RTI. Within the **phyFedAmb** module, the messaging protocol is employed to pack any callback method with its parameters into IPC messages. The IPC enqueues the callback message to the **Callback Processor** module at the virtual federate. Through the messaging protocol, the callback processor activates the corresponding **fedAmb** method implemented by the user. From the simulation users' perspective, a combination of one virtual federate and its corresponding physical federate operates as a simulation federate in the context of the decoupled architecture. The federate combination performs an identical execution to the normal simulation federate using the same code in the virtual federate. In future discussion, we will explicitly use "normal federate" to refer to a traditional federate that directly interacts with the RTI. By default, in the discussion of this paper a clone or a federate contains a virtual federate process and a physical federate process.

### 3.2.2 Inside the Decoupled Architecture

As discussed above, the decoupled approach interlinks a virtual federate and the physical federate into a simulator that performs an identical simulation to the corresponding normal federate. This section gives the details of how an RTI service call is executed and the callback is invoked in the decoupled federate architecture.

Figure 3 depicts the procedure where a simulation model initiates an RTI call and waits for a return from the real RTI, using the *updateAttributeValues* method as an example. The procedure is as follows:

- The virtual federate invokes the redefined *updateAttributeValues* method.

- Inside the *updateAttributeValues* method, the *packMsg* routine extracts the data stored in the *AttributeHandleValuePairSet (AHVPS)* and packs them together with other parameters such as the associated timestamp, object instance handle and tag into an RTI\_UPDATE message.
- The IPC enqueues the RTI\_UPDATE message to the physical federate. The virtual federate switches to waiting mode for the returned message.
- Once the physical federate receives the IPC message, it invokes the *unpackMsg* routine to process it according to the associated type, RTI\_UPDATE.
- A new *AHVPS* object and related parameters are recovered based on the IPC message and passed to the *RTI::updateAttributeValues*, which invokes the real RTI service.
- On the accomplishment of this *RTI::updateAttributeValues* call, the physical federate acknowledges the virtual federate with an IPC message containing the returned value.
- The *updateAttributeValues* call finishes and the data retrieved from the acknowledgement message is returned to the simulation model.

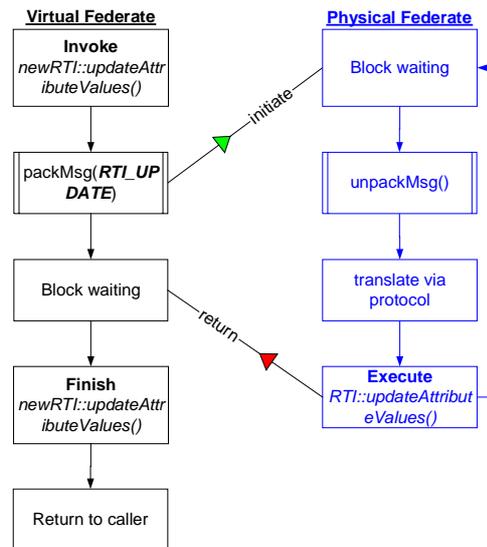


Figure 3: Executing an RTI Call in the Decoupled Architecture

From the user's point of view, the initiation and accomplishment of an RTI call are identical to the original normal federate. The semantics of RTI services are kept intact in the decoupled approach.

The RTI software has an interface that provides flexible methods to the user for packing update data and leaves the transmission details transparent. The user can

create update data of variable lengths. However most IPC mechanisms have limitations in message size and buffer size. For example, the **Message Queue** based on Solaris defines the maximum queue length as 4096 bytes (Stevens 1998). The message sender and receiver must agree with each other on the same message length. If a fixed message size is defined for IPC messaging, it may incur some unnecessary overhead. A fixed large size is inefficient in transmitting small messages. On the other hand, a fixed small size increases the overhead for packing, delivering and unpacking a large number of small packets in the case of processing large messages. Thus a protocol is proposed for messaging between the virtual federate and physical federate. We define a small message size (MSG\_DEF) and a large message size (MSG\_LG) for assembling user data into packets. A special “PREDEFINE” packet is used to notify the receiver if large or multiple packets are to be sent for a single data block. Figure 4 gives the messaging details based on this simple protocol.

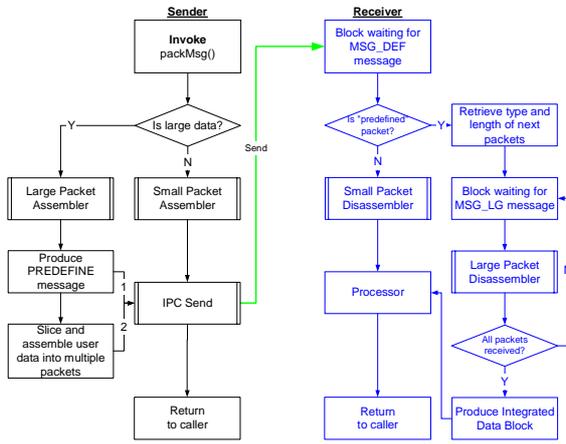


Figure 4: Messaging between Virtual Federate and Physical Federate

The RTI communicates with a federate via its federate ambassador provided by the user (DMSO 2002). A federate must explicitly pass control to the RTI by invoking the *tick()* method. For example, the RTI delivers the Timestamp Order (TSO) events and Time Advance Granted (TAG) to a time-constrained federate in strict order of simulation time, which coordinates event interchange among time regulating and time constrained federates in a correct and causal manner. Therefore, the decoupled architecture should guarantee that (1) the federate ambassador at the user federate works in a callback like manner and (2) callback methods are invoked in the correct order. Figure 5 depicts how to realize these functionalities. To ease discussion, we

assume the physical federate will get the callbacks shown in Figure 5. This procedure is illustrated by the following steps:

- The Virtual federate invokes the routine *newRTI::tick()* and the latter sends out an RTI\_TICK message to the physical federate.
- The Physical federate calls the real RTI *tick()* according to the RTI\_TICK message.
- The local RTI component acquires control and delivers events to the phyfedAmb module of the physical federate in a strict order.
- In each callback method invoked, the data sent by the RTI is enqueued to the callback IPC channel. The routine inside the *newRTI::tick()* accesses the queue for the virtual federate.
- As long as the RTI\_TICK\_DONE message is not detected, the callback processor continuously processes the messages in a FIFO order while activating the corresponding method in the fedAmb module based on the messaging protocol.
- At the physical federate side, once the RTI finishes its current work and passes control to the physical federate, the latter returns an RTI\_TICK\_DONE message to the virtual federate.
- On receiving the RTI\_TICK\_DONE message, the virtual federate accomplishes the *newRTI::tick()*, and control is returned to the caller immediately.

The real RTI starts to take charge only when the physical federate explicitly invokes *RTI::tick()*. On the other hand, the *newRTI::tick()* can only return when the real RTI finishes its work. The communication channels linking the virtual federate and physical federate work in a FIFO manner. Thus the order of each callback method invoked at the physical federate is identical to the sequence in which the callback processor at the virtual federate processes the data. From the user’s perspective, the callback mechanism based on the decoupled approach executes the equivalent operations to the normal federate. It guarantees consistency in presenting interactions from the real RTI to the simulation model and also ensures user transparency.

The decoupled architecture requires an additional IPC communication layer although it performs exactly the same computation as the corresponding normal federate. The external communication may incur some extra overhead. To investigate the overhead incurred by the decoupled approach, a series of benchmark experiments has been performed to compare with the normal federates. Section 4 reports and analyzes the experimental results in terms of event transmission latency and synchronization efficiency.

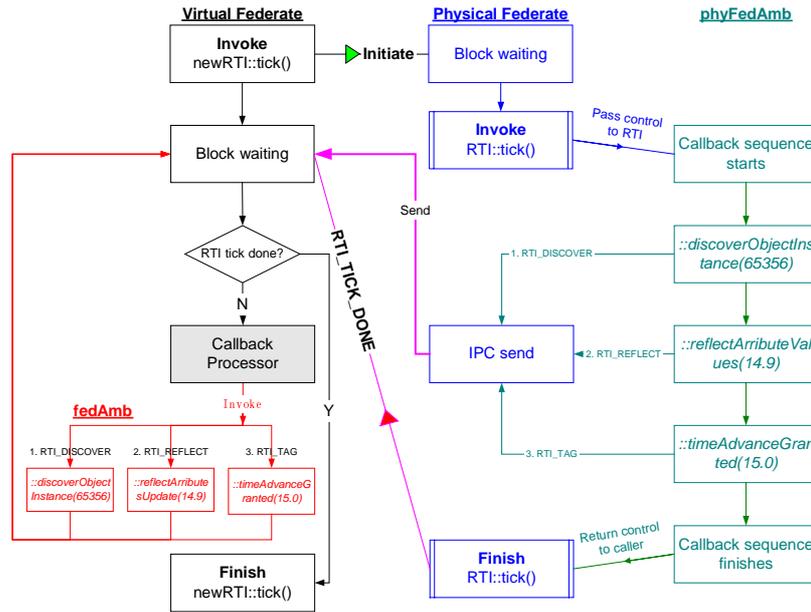


Figure 5: Conveying Callbacks to the Virtual Federate

### 3.2.3 Fault Tolerance

In an HLA-based distributed simulation, the participating federates running at different locations are liable to failure. A lot of factors may contribute to the failure of a federate, for example, network congestion, platform crash of the RTIEXEC process (DMSO 2002) etc. It is also difficult to handle such failure during runtime because most RTI implementations operate as a black box. In a large-scale distributed simulation, the crash of one federate can lead to failure of the overall simulation. More and more federates will participate in the same federation with the cloning of federates, which increases the possibility of simulation failure. Although users model the simulation federate properly, an RTI failure still induces simulation collapse. Simply restarting a new federate to substitute the crashed one is not applicable since the consistency of the overall simulation state is lost. Considering the complexity and distribution of the individual simulation models and the number of federates in a large-scale distributed simulation, it is costly to restart the overall distributed simulation. As fault tolerance (Danami and Garg 1998) is needed in a distributed simulation, we propose using the decoupled approach to address the potentially unpredictable faults at the RTI level. In this study, the fault tolerance aims to minimize the wasted distributed computation and to provide user transparency. In other words, the user does not have to intervene into the running simulation to deal with RTI failure.

The middleware approach enables the interception of RTI services invoked by the simulation model. The

system state at the RTI level is accessible using the middleware approach (Chen et al. 2003a). Thus we can retrieve the “features” of a federate, such as the object classes subscribed and published as well as whether the federate is time constrained or time regulating. Furthermore we can log the “operation” history of a federate. The middleware can track the object instances registered and each attribute update to any object instance. All these operations are hidden beneath the newRTIamb interface. Based on the stored information, a crashed federate can be replaced by a new federate with inherited system states from the system state log. By “plugging” the new federate back to its virtual federate, the distributed simulation can continue without being interrupted by the previous RTI failure. The approach can also take advantage of the federation save and restoration mechanism provided by RTI services. This mechanism is indicated as in Figure 6. The same model of state saving and recovery used in cloning federates can also provide this fault tolerance.

Figure 6 gives a model of how fault tolerance can be achieved with the decoupled architecture, in which one of the running federates (marked as *m*) is highlighted for study. As illustrated in Figure 6(A), at runtime the middleware intercepts the invocation of each RTI service method. The interceptor logs all the RTI system states into stable storage. Some RTI states are relatively static, such as the federate identity, federation information, the aforementioned declaration data and time features. The static states also include the registered or deleted objected instances. Some other RTI states are highly dynamic, such

as granted federate time, sent and received interactions, updated and reflected attribute values of object instances, etc.

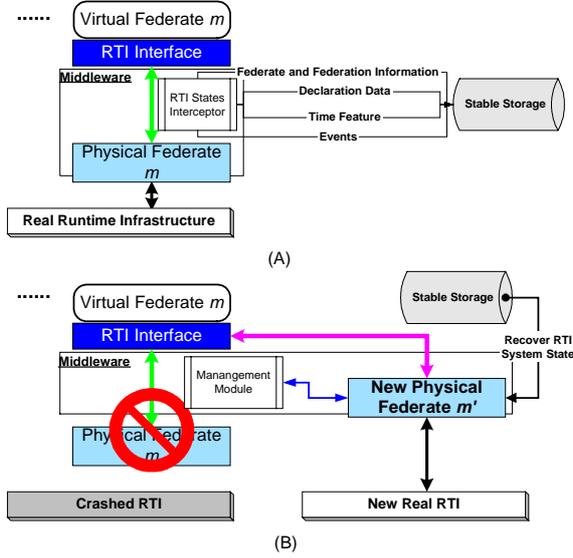


Figure 6: Fault Tolerance using Decoupled Federate

As soon as the middleware detects the RTI failure, no matter whether it is due to a local physical federate or incurred by other federates or for some other unpredictable reasons, the management module within the virtual federate will cut off the connection from its physical federate and terminate it (as shown in the left side of Figure 6(B)). Subsequently the management module will initiate a new physical federate instance  $m'$  and have it join the existing federation or possibly a new federation with another RTIEXEC process. When the whole federation fails, other virtual federates can also perform the same action and form a new workable federation together in the same way. After that, the physical federate reads in the RTI state from the stable storage. It invokes the corresponding RTI services with restored parameters to recover the features of the old federate and resumes the dynamic system states in the snapshot obtained from the stable storage. Finally the virtual federate continues execution with the new physical federate. Thus, the physical federate works as a plug-and-play component, it can be replaced and transplanted at runtime.

#### 4. BENCHMARK EXPERIMENTS AND RESULTS

In order to investigate the overhead incurred in the decoupled architecture, we perform a series of benchmark experiments to compare the decoupled federate with a normal federate. The performance is compared in terms of latency and time advancement calculation. Latency is

reported as the one-way event transmission time between one pair of federates. The time advancement performance is represented as the time advance grant rate.

#### 4.1 Experiment Design

The experiments use three computers in total (two workstations and one server), in which the server executes the RTIEXEC and FEDEX processes. The federates that run at one independent workstation are enclosed in a dashed rectangle. In our case  $fed A[i]$  and  $fed B[i]$  ( $i \geq 0$ ) occupy workstation 1 and workstation 2 respectively. The computers are interlinked via a 100Mbps-based backbone.

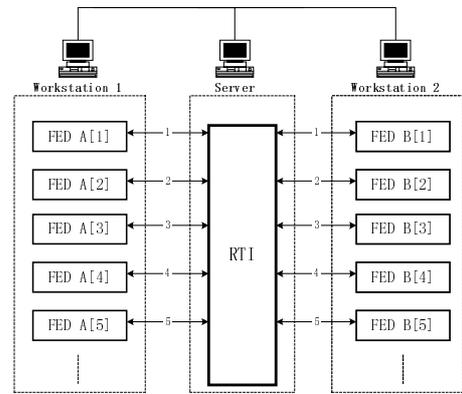


Figure 7: Architecture of Benchmark Experiments

The server (Ultra-Enterprise) has following specification:

- sparcv9 processor (\* 6) operating at 248 MHz
- 2048 Mbytes of RAM
- Sun Solaris OS 5.8
- GCC 2.95.3
- DMSO RTI NG 1.3 V6 for the SunOS-5.8 operating system and the gcc-2.95.3 compiler

The workstations (SunBlade 1000) have the following specification:

- sparcv9 processor operating at 900 MHz
- 1024 Mbytes of RAM
- Sun Solaris OS 5.8
- GCC 2.95.3
- DMSO RTI NG 1.3 V6 for the SunOS-5.8 operating system and the gcc-2.95.3 compiler

The experiments emulate the simulation cloning process by increasing the number of identical federates. As shown in Figure 7,  $fed A[1]$  and  $B[1]$  form a pair of initial federate partners, which represent the federates to be cloned.  $Fed A[i]$  and  $B[i](i > 1)$  stand for the  $i$ th clones of the two original federates respectively. The architecture

is used through all the benchmarks experiments and for both normal federates and decoupled federates.

A DDM based approach is used to partition concurrent scenarios (Chen et al. 2003b). For the latency benchmark, each pair of federates have an exclusive point region associated to any event being exchanged. The federates are neither time regulating nor time constrained. In one run, each federate updates an attribute instance and waits for an acknowledgement from its partner (from *fed A[i]* to *fed B[i]*, and vice versa) for 5,000 times with a **payload of 100, 1000 and 10,000 bytes**. The time interval in the ping-pong procedure will be averaged and divided by 2 to give the latency in **milliseconds**. A federate merely reflects the events with identical region to itself. In other words, *fed A[i]* only exchanges events with *fed B[i]*.

As for the time advancement benchmark, all federates are time regulated and time constrained. Each federate has lookahead 1.0 and advances the federate time from 0.0 to 5,000.0 with timestep 1.0 using *timeAdvanceRequest* (DMSO 2002). The results report the rate that the RTI issues *timeAdvanceGranted* (TAGs/Second).

#### 4.2 Latency Benchmark Results

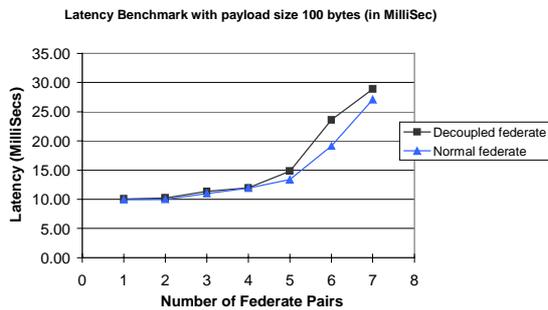


Figure 8: Latency Benchmark on Decoupled Federate vs Normal Federate with Payload 100 Bytes

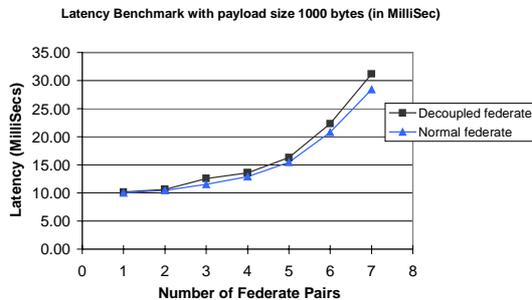


Figure 9: Latency Benchmark on Decoupled Federate vs Normal Federate with Payload 1000 Bytes

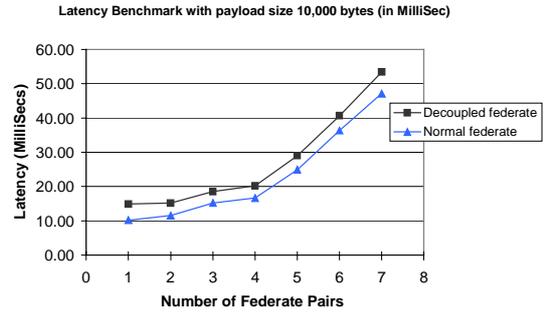


Figure 10: Latency Benchmark on Decoupled Federate vs Normal Federate with Payload 10,000 Bytes

The latency benchmark experiments report the latency with three different payload sizes. From Figure 8 to Figure 10, we can see that no matter whether the payload size is small or large, the latency increases steadily with the number of federates. The increment becomes obvious when the number of federates exceeds 4 pairs (8 federates in total). As indicated in Figure 8 and Figure 9, when the payload is not greater than 1000 bytes, the latency varies from about 10 milliseconds for one pair of federates to about 30 milliseconds for 7 pairs of federates. The decoupled federate and normal federate show similar results in this situation, and the decoupled federates incur only slightly more latency than the normal ones. As shown in Figure 10, when a bulky payload as large as 10,000 bytes is applied, the decoupled federates incur about 5 milliseconds extra latency to the normal ones. However the extra latency remains nearly constant with the number of federate pairs. The latencies for both types of federates increase more rapidly than the small payload cases. This is due to the extra overhead incurred by Inter-Process Communication, which becomes obvious with bulky data transmission between the physical federate and virtual federate. When the payload size and the number of participating federates are not too large, the decoupled federate has a similar performance to the normal federate in terms of latency.

#### 4.3 Time Advancement Benchmark Results

In the time advancement benchmark, the TAG rate decreases with the number of federates for both decoupled and normal federates. The rate decreases less rapidly when the number of federate pairs is greater than 4 (8 federates in total). The TAG rate is about 300 times per second for one pair of federates down to about 40 times per second for 7 pairs of federates. The results indicate that the performance of these two types of federates is very similar in terms of time advancement.

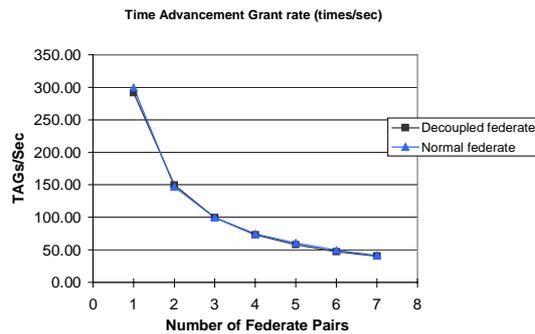


Figure 11: Time Advancement Benchmark on Decoupled Federate vs Normal Federate

## 5. Conclusions and Future Work

In this paper, we have investigated some issues in cloning federates in distributed simulations. In order to overcome the problems of replicating a federate instance and providing fault tolerance to a distributed simulation, a decoupled federate architecture is proposed. This approach decouples the simulation model from the RTI local runtime component in a normal simulation federate. A federate is separated into a virtual federate process and a physical federate process, where the former executes the simulation model and the latter provides RTI services at the backend. A standard RTI interface is presented to support user transparency at the virtual federate, while the original RTI component is substituted with a customized library. The decoupled architecture enables a relatively generic method of replicating the simulation model. It also facilitates state saving and recovery at the RTI level for cloning a federate and fault tolerance. The proposed approach guarantees the correctness of executing RTI services calls and reflecting RTI callbacks to the simulation model.

Benchmark experiments have been performed to investigate the overhead incurred by a decoupled federate architecture. The experimental results are compared for a decoupled federate and normal federate in terms of latency and time advancement performance. The results indicate that the decoupled architecture incurs only a slight extra latency in the case of a bulky payload and has a very close performance of time advancement compared with a normal federate.

The decoupled architecture can provide other advantages to distributed simulation technology. The potential application avenues are as follows:

- Using a communication channel between the virtual federate and physical federate, we are able to

distribute the computational complexity of one federate with a heavy load in a cluster computing environment

- Physical federates can be more independent, which allows the further optimization of the computation. For example, the physical federates can monitor the federation and optimize the computation by migrating the virtual federates to other nodes

For our future work, we need to further explore the mechanism of the cloning operation to ensure simulation consistency. Another challenge is the interactive manipulation of cloning-enabled simulation in a distributed environment, where users are offered the flexibility to control and update the cloning online.

## REFERENCES

- Chen, D.; B. P. Gan; S. J. Turner; W. Cai; N. Julka; and J. Wei. 2003. "Evaluating Alternative Solutions for Cloning in Distributed Simulation". *Proceedings of the 36<sup>th</sup> Annual Simulation Symposium* (Orlando, Florida, USA, Mar), 201-208.
- Chen, D.; B. P. Gan; S. J. Turner; W. Cai; and J. Wei. 2003. "Data Distribution Management in Distributed Simulation Cloning". *Proceeding of 2003 European Simulation Interoperability Workshop*, (Stockholm, Sweden, June), paper no. 03E-SIW-024.
- Dahmann, J. S.; F. Kuhl; and R. Weatherly. 1998. "Standards for Simulation: As Simple As Possible But Not Simpler, The High Level Architecture for Simulation". *Simulation*, 71:6 (Dec), 378-387.
- Danami, O. P. and V. K. Garg. 1998. "Fault-Tolerant Distributed Simulation". *Proceedings of the 12<sup>th</sup> Workshop on Parallel and Distributed Simulation* (Banff, Albert, Canada), 38-45.
- DMSO. 2002. RTI 1.3-Next Generation Programmer's Guide Version 5 (Feb), *DoD, DMSO*.
- Gan, B. P.; L. Liu; S. Jain; S. J. Turner; W. Cai; and W. Hsu. 2000. "Distributed Supply Chain Simulation Across Enterprise Boundaries". *Proceedings of the 2000 Winter Simulation Conference* (Orlando, Florida, USA), 1245-1251.
- Hybinette, M. and R. M. Fujimoto. 2001. "Cloning parallel simulations". *ACM Transactions on Modeling and Computer Simulation*, Volume 11, (Oct), New York, USA, 378-407.
- Morse, K. L. and M. D. Petty. 2001. "Data Distribution Management Migration from DoD 1.3 to IEEE 1516". *Proceeding of the Fifth IEEE International Workshop on Distributed Simulation and Real-Time Applications* (Cincinnati, Ohio, USA, Aug), 58-65.
- Schulze, T.; S. Straßburger; U. Klein. 2000. "HLA-federate Reproduction Procedures In Public Transportation Federations". *Proceedings of the 2000 Summer Computer Simulation Conference* (Vancouver, Canada, Jul).
- Stevens, W. R. 1999. "UNIX Network Programming, Inter-Process Communications". Vol. 2, 2<sup>nd</sup> Edition, Prentice Hall.
- Turner, S. J.; W. Cai; and B. P. Gan. 2001. "Adapting a Supply-chain Simulation for HLA". *Proceeding of the Fourth IEEE International Workshop on Distributed Simulation and Real-Time Applications* (San Francisco, California, USA), 67-74.

# ADVANTAGES AND DISADVANTAGES OF BUILDING BLOCKS IN SIMULATION STUDIES: A LABORATORY EXPERIMENT WITH SIMULATION EXPERTS

Edwin C. Valentin, Alexander Verbraeck, Henk G. Sol  
Faculty of Technology, Policy and Management  
Delft University of Technology  
P.O. Box 5015, 2600 GA Delft, The Netherlands  
E-mail: {edwinv, alexandv, henks}@tbn.tudelft.nl

## KEYWORDS

Simulation studies; building blocks ; transportation systems; decision support

## ABSTRACT

Many logistic problems are solved using simulation, however these studies often take too much time and cost too much. One of the reasons for this is the lack of a clear structure of simulation models. To solve this problem we postulate, based on our research, that simulation building blocks can be used to provide fast and easy construction of simulation models that are easy to maintain and to extend. We defined a set of laboratory experiments to evaluate whether building blocks really provide the benefits we expect. In this paper we describe a laboratory experiment in which simulation experts were asked to perform a simulation study and to provide as much support to the problem owners as possible. The experts were divided into two groups: a group with and a group without building blocks. The outcome was nothing like we expected. None of the experts managed to reach an acceptable level of performance. The experts using building blocks faced a lot of errors due to sloppy user input and the experts using plain simulation constructs were still configuring their models at the end of the time allowed for the experiment. The participants using building blocks mainly complained about documentation and the training material, but felt that they understood the building blocks and could, in future, carry out a high-quality simulation study more quickly.

## INTRODUCTION

Simulation is often used as a research methodology for problem solving. Many different books exist that show how this research methodology can be used when dealing with logistic problems (Law and Kelton, 1999; Kelton et al, 2002; Banks, 1999; Harrington and Tumay, 1999). The books explain when simulation is useful and what the different involved actors should do. Yet even though there is extensive literature on how a simulation study should

be performed, lots of pitfalls can still be identified and these can cause simulation studies to fail. This results in dissatisfied problem owners, projects that miss their deadlines and overspending of budgets. Keller et al (1991) summarize the pitfalls of failed simulation studies as:

- low salesmanship of the simulation expert to the problem owner, i.e. the problem owner does not understand what the simulation expert wants or is trying to do,
- low skills on behalf of the simulation expert, mainly regarding specific domain knowledge and statistical background,
- lack of time to complete the study, so the study is abandoned before all the necessary experiments and statistical tests have been performed.

Robinson (1999) also describes a set of pitfalls specific to simulation studies that cannot be solved using a clear process for a simulation study. These problems result from the *structure* of a simulation model. Robinson lists reasons for these pitfalls:

- the simulation model development is started from scratch, reuse is not applied,
- the implemented simulation model does not fit with the conceptual model,
- the simulation environment leads to unstructured complex models,
- the simulation model is too inflexible and has a limited set of options for experimentation.

The result of these problems is that it takes too long to construct a simulation model, and the models are difficult to use for experiments. When a simulation model is used for experiments to evaluate different scenarios, the simulation model often needs to be extended or adjusted, and a rigid structure makes it difficult to change the model to incorporate the needed extensions. Another problem that is observed is that the problem owner has difficulties understanding the structure of the model, and relating the structure to the real world system.

A possible solution to these problems is to use reusable blocks to form a simulation model, because – when designed well – these blocks can represent an element of the system in the way the problem owner expects, both in structure and in behavior. We use the term building blocks (Valentin and Verbraeck, 2002) to describe the concept of designed reusable simulation modeling blocks. The concept of building blocks encompasses the idea of decomposing a prototypical system within a domain and implementing the observed domain elements in a standard simulation environment. The expected benefits of simulation building blocks are a higher recognizability of simulation models, easier construction and adjusting of simulation models and an ability to transfer a simulation model to an environment where there is less experience in simulation and experimentation. These benefits should result in a better support for problem owners, because they will receive more insight into their system in less time.

We have implemented sets of building blocks for problems in different domains, such as modeling passenger flows at airports, luggage handling at airports, information flows in supply chains, transaction management in international banking and traffic flows at container terminals. Then, we performed simulation studies in these different domains using the building blocks to construct our models. These case studies taught us much about the structure of building blocks and pointed to several advantages of using such blocks. In this paper we describe a set of laboratory experiments to assess the added value of building blocks by comparing simulation studies using building blocks with studies using a standard simulation environment. The group of analysts consists of simulation experts, who have an experience of many years using the chosen simulation package. Earlier laboratory experiments looked at the ease of adjusting a simulation model to run simulation experiments, and at the construction of a simulation model using building blocks. Both experiments were performed by novices and showed us that some parts of simulation studies can benefit from using building blocks, but that activities where there where no building blocks used caused the novices more problems than we expected. We therefore wanted to see whether the benefits of using building blocks remain when we study simulation experts instead of novices. The research question for this third laboratory experiment was:

*Can simulation experts provide better support to a problem owner by using building blocks than by using the standard constructs of a simulation environment?*

We give more background on building blocks and the main lessons we hoped to learn from this laboratory experiment in section two. In section three we describe the set-up of the laboratory experiments, and in section four we discuss the outcome of laboratory experiments. We conclude this paper by relating the laboratory experiment described in this paper to future planned research and experiments.

## **BACKGROUND**

Simulation environments increasingly provide us with constructs, which allow us to later reuse parts of our model. The main arguments provided by the simulation vendors to support these features are faster model construction and an ability to reuse previous work. In our research projects we tried to work with the provided concept. The first simulation study that we performed in a domain by using (reusable) objects was a success, but follow-up simulation studies in the same domain were much harder and did not reuse the objects as much as we expected (Verbraeck et al, 1998; Hooghiemstra and Teunisse, 1998). We also had problems with transferring the objects to other modelers, who did not understand the objects or tended to ‘misuse’ them, or use them in a different way than we had intended.

The experiences we had with these cases caused us to come up with the idea of building blocks. Our research is based on the on-going research of the BETADE research program (Verbraeck et al, 2002). BETADE defines the concept of building block as:

*A building block is a self-contained, interoperable, reusable and replaceable unit that encapsulates its internal structure and provides useful services to its environment through precisely defined interfaces*

In the software engineering the similar definitions are used for software components. Within the BETADE research program is argued that a BuildingBlock applies in many more domains than only software. Therefore a software component is an implementation of a building block in software. Based on the knowledge of the BETADE research program and our experience drawn from the above mentioned cases, we filled in the concept of simulation building blocks using a structure consisting of different levels of abstraction and different types of blocks (Valentin and Verbraeck, 2002). When we used these new building blocks in a set of case studies the expected benefits of faster model construction, structured models and a reduced need for experienced simulation experts seemed to be achieved; however, we had nothing with which to compare our results. The main questions were: Was it *really* faster? And: did it *really* reduce the need for

experienced simulation experts to successfully carry out a study? Because we could not answer these questions we also could not say whether building blocks really provide improved support for problem owners.

From the literature on software engineering research we learned that expert and novice software developers use software components in a different way. The experts are more hesitant to use components and prefer to construct models or components themselves, because they are not sure whether they can trust a component made by others. Novices are glad the components are available and see them as their best option. We expected the same kind of outcome in simulation studies, and we wanted to evaluate whether experts and novices used the simulation building blocks in accordance with our conceptual definitions. The planned set of laboratory experiments was designed to show us the difference between building blocks and constructs of standard simulation environments and whether we needed to improve the conceptual model of building blocks in some way.

## LABORATORY EXPERIMENT

### General Information

Our goal for the whole range of laboratory experiments was to identify whether simulation studies are carried out more effectively when building blocks are used, rather than constructing the model from the elements of a standard simulation environment. Figure 1 is a simple representation of the expected effects of building blocks on the outcome of a simulation study, the '+' and '-' signs in the figure denote the expected relationships, which we have evaluated in different laboratory experiments. Process descriptions of simulation studies (Law and Kelton, 1999; Kelton et al, 2002; Banks, 1999; Harrington and Tumay, 1999) show that the number of actions a simulation expert has to perform varies with the phase of the project. Thus we needed different kinds of experiments to evaluate the sub-processes to permit conclusions to be drawn for all the '+' and '-' relationships. We describe in this paper the third laboratory experiment in the range of evaluating building blocks in predefined settings.

We used two sets of building blocks with different levels of "domain specificity" in the different laboratory experiments. More precisely, we used one set of building blocks and constructs taken from the simulation environment Arena (Kelton et al, 2002). Arena is one of the most popular commercial simulation environments at the moment, and it was one of the first simulation environments that allowed for the development of building blocks by simulation experts. We use this environment to teach simulation to our students at the faculty of Technology Policy and Management of Delft University of Technology. This in turn has helped us to get a reasonable set of novices who are willing to participate in the experiments. Finally the vendors of the simulation environment were willing to participate in laboratory experiments and to provide experienced simulation experts for experiments to test the effectiveness of simulation building blocks. Experts from Rockwell Software were studied in this laboratory experiment. These experts are developers of the simulation software Arena and internal consultants with several years of experience.

We required a case study for the laboratory experiments aimed at measuring the relationships shown in figure 1. It was also necessary that the case could be modeled within a limited time, that it showed repetition to provide a trigger for the use of building blocks, and that it was suitable to be modeled using the simulation environment Arena. We ended up with a design problem for an advanced automatic public transportation system. This system, based on a simulation project performed by Brandt (1999), has been proposed to provide public transport between the cities of The Hague and Ypenburgh in the Netherlands. We judged that we could reuse all the data from this study to provide input and output, and a good (verified and validated) set of performance indicators.

The map of the expected route between the cities of The Hague and Ypenburgh is shown in figure 2. This simulation study was designed to handle a new and advanced automatic transportation system. The route is fixed, but there are still a lot of open design choices that need to be evaluated using simulation.

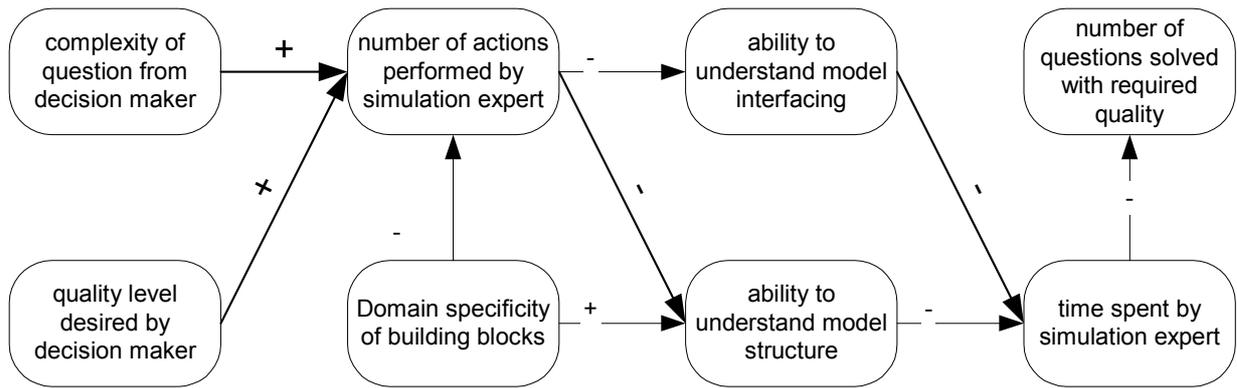


Figure 1: Causal diagram simulation study

Some of the choices are:

- type of vehicle; monorail versus cable mover, large versus small vehicles, fast versus slow
- number of vehicles
- daily pattern of the vehicles
- number of platforms at stations
- number of tracks between stations

These choices need to be evaluated for a large set of scenarios. A small set of variables needed to be varied in the scenarios to test the proposed solution under a wide range of assumptions. The variables are: arrival pattern of passengers, origin-destination relationships of passengers, effects of new offices or leisure activities in the region attracting new passengers, effects of transfers from and to the conventional public transportation (bus, tram, and train).

We developed a set of building blocks that fit the building block concept of Valentin and Verbraeck (2002). The set of building blocks consists of blocks for the physical infrastructure, e.g. *track*, *platform*, *station* and several building blocks for *control* and *generators* for passengers and vehicles. Using the building blocks we were able to develop the same models that Brandt developed using the Arena simulation language without building blocks. We used the same performance indicators and data input and evaluated the outcome. The key values, among them traveled kilometers for vehicle and passengers, utilization of vehicle and wait time were the same, with a 95% confidence interval, as in the original model.

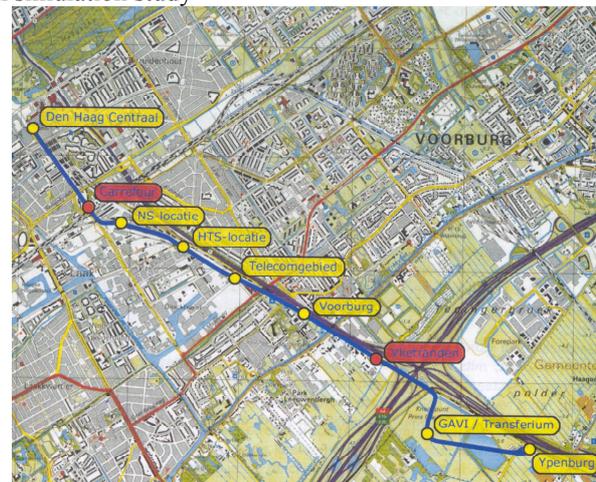


Figure 2: Expected route for the SkyShuttle transportation system

### Set-up Laboratory Experiment

In a first simulation experiment, we gave the Delft students a pre-developed simulation model, either with or without building blocks, for the transportation system problem. The students had to perform different simulation experiments using these models. The main outcome of this laboratory experiment was that the novices in simulation using building blocks could adjust the models faster for experiments than the other group, this group thought that their models were easy to understand and maintain, but they completely forgot about the need to validate their models.

In the second laboratory experiment we asked participants with a similar background and knowledge as in the first experiment to develop a full simulation model from scratch for the problem situation, including the preparation of the experiments. The students using building blocks could develop a simulation model quite quickly, but had a lot of problems with the user-input and producing of a good and valid initial set-up. The students using simulation constructs worked hard, but forgot to abstract from their problem working in more detail than the students using building blocks.

Based on both observations we concluded that simulation experts using building blocks is usually faster, because they can quickly set-up their model, but using standard simulation constructs is not so bad, because modelers can reduce and abstract their initial model much better. The second laboratory experiment triggered a third laboratory experiment using simulation experts, which is described in this paper in more detail. We expected that simulation experts would perform better than novices, and we had them work on the complete simulation study where the simulation model had to be built from scratch – with and without building blocks. We limited the participants' time to 8 hours, let them construct a simulation model at the level of detail they preferred and let them perform the experiments that they assumed to be key. Afterwards we evaluated the performance of the simulation experts by showing the outcome of the study to real problem owners.

Eight employees from the company Rockwell Software participated in the experiment. Rockwell Software bought the company Systems Modeling that developed the simulation environments SIMAN and Arena. A sample of 8 persons is small, but given the expertise of these people we were glad they could participate. All of the participants involved had been working for at least 5 years in simulation, some of them even for 20 years. The group consisted of developers of the Arena core code, developers of templates in Arena and consultants that use Arena in commercial projects. Even though the group consisted only of simulation experts, the experts were not comparable, e.g. a junior consultant can not be compared to a developer of the Arena core who has published more than 20 scientific articles about discrete event simulation. The group was divided into four comparable couples:

- two developers of Arena software with more than 15 years of experience and a PhD in computer science,
- two expert Arena user with  $\pm 10$  years of experience,
- two developer of commercial Arena Templates (Call Centers and High Speed Packaging) with  $\pm 7$  years of experience,
- two junior consultant mainly using Arena, including VBA with  $\pm 5$  years of experience in simulation projects.

One person of the couple, chosen at random, received building blocks developed in Arena including all documentation (but not the source code). The partner person was expected to work with the Arena basic constructs.

The participants were given a maximum of 8 hours to develop their model and to run any number of experiments they assumed to be necessary. The experiments were meant to demonstrate the validity

of the simulation models. If they finished in less than 8 hours, the participants could continue doing other things. None of the participants was able to participate for a straight 8 hours in a row, so they divided the laboratory experiment over a period of three days.

All the participants received documentation about the concepts that could be used for the simulation models. These concepts formed the basis of the building blocks that were given to the experts using building blocks. It was made very clear to all participants that the conceptual model was just an overview and they were not forced to comply with the conceptual models. The group using building blocks received some extra material about the background and technical implementation of the building blocks. The time that participants with building blocks needed to understand this material was included as part of the 8 hour allowed for the complete laboratory experiment.

At the end of the laboratory experiment the participants were required to provide a set of deliverables.

1. A simulation model based on the building blocks or the Arena basic constructs.
2. An extension to the presentation of the SkyShuttle team that could be used to explain the problem solution to the problem owners at the Municipality of The Hague
3. A filled in questionnaire about their satisfaction with using either the building blocks or the Arena constructs and their expectations of the model development assignment
4. A log-file describing their activities of the 8-hour period.

#### **Plans for Evaluation**

The evaluation we planned to do using the results of the participants consisted of three steps. The first step dealt with the problem owner of the SkyShuttle project. The problem owner needed to feel supported by the simulation expert, based on model outcome, visualization, experiments and useful model abstractions. The second step was to judge the quality of the simulation model, using simulation experts, on level of detail, completeness, model structure and ease of adjustment. The third step was to evaluate the questionnaire and log files created by the participants.

We did not have real problem owners for this evaluation, because we were working with a slightly adapted case compared to the original study. However, we asked the problem owners of the initial study that triggered this setup for the laboratory experiment to participate, and some other experts drawn from the field of transportation who have used simulation models in their projects. We developed a

list of more than 50 items that a problem owner might be interested in. We planned to let the problem owner set the priority of these items and then judge the work of each of the participants to determine how well it dealt with the top 15 items. We planned to do the same thing with the simulation expert, we first showed them a large list of items, then we let them prioritize and score the top 15 items for the final simulation models of each of the participants. The two basic lists were developed together with R. Sadowski, the chair of the annual Arena-modeling contest for undergraduate students.

Based on the material of the problem owner, the simulation experts, the questionnaire and the log-files we expect to see that:

- the problem owner would judge the visualization and performance indicators of the simulation models using building blocks to be more valuable than the visualization and performance indicators of the simulation models based on the basic constructs,
- the simulation models based on the building blocks would contain more details than the models using basic constructs,
- the first simulation model based on the basic simulation constructs of Arena would differ more in detail, quality and animation, from the final simulation model, compared to the differences between the first and final model using building blocks,
- the participants using the building blocks would be more positive about the quality of their simulation models, technically and visually, compared to the model developers using basic constructs,
- the participants using the building blocks would assume they had better met the problem owner's needs regarding visualization, performance indicators and preparation for future experiments compared to the assumptions made by model developers using basic constructs.
- the participants using building blocks would agree more with the statement that they had had enough time compared to the model developers using basic constructs.

## **OBSERVATIONS LABORATORY EXPERIMENT**

Unfortunately we could not apply the evaluation plan as we designed it. The participants in the laboratory experiment did not succeed in finishing the simulation study. None of the participants performed a full range of simulation experiments and none of them provided a good design for the SkyShuttle system. The participants provided different reasons for failure and combined with our observations this lead to some additional conclusions, which we will

discuss below. We would have preferred to use the objective problem owners and simulation experts, but because the work was not finished, it did not make sense to bother these volunteers with the unfinished outcomes produced by the participants.

### **Observations during the laboratory experiment**

During the laboratory experiment we made notes of the things we noticed regarding the processes and the models of the participants. We then evaluated these observations with the participants, remarkably similar ideas were identified and registered, this allows us to speak of the participants using building blocks as one composite individual and the participants using the standard Arena concepts as another.

The participants using building blocks started directly, clicking the simulation model together based on the diagrams shown in the documentation. They copied the data from the Excel sheets, unfortunately this copying was done by retyping instead of copying all the text at once, and once they were finished they tested the model to see if it worked. This first attempt to run a full model took two and a half to four hours of work.

The participants using building blocks were convinced their model was correct and as a result they were surprised when they received error messages. The errors ranged from "reserved name" to "linker errors" and "undefined symbol". The participants spent the following hours solving their problems. The participants used different ways to do this. Two of the participants dived into the example models and tried to see what was different, they performed the test assignments and got stuck with the examples, because their results differed by 0.4 % with the mentioned values of the main performance indicators. The other participants constructed the whole model twice before they noticed they had made a typing error in one of the names of one of the elements in their simulation model.

Once the participants using building blocks got the model running, but the model contained deadlocks and produced an odd outcome. Both problems were due to an invalid configuration and should have been solved by applying different parameter settings, e.g. more tracks, different platforms, or a different vehicle frequency. However, the participants doubted the quality of the building blocks and started to debug the simulation model using the SIMAN-command view. As a result they did not succeed in performing experiments. These participants stated that they now understood the working of the building blocks and would be well able to perform the test if they were asked to do it again.

The participants without building block all started with a good walk-through of the problem description and the provided conceptual model. Probably they followed the conceptual model so closely that they did not think to deviate from it, because the provided simulation models were very comparable to the conceptual models provided to act as an example. However, the conceptual models contained a lot of details like the scheduling of vehicles, the behavior of doors and destination schedule of passengers. As a result of not applying reduction to their models, none of the participants using the basic Arena constructs succeeded in developing a complete working model in Arena.

The main reason for the lack of outcome was a lack of reduction, but some additional reasons were individual choices in setting up the of their simulation models as well. One of the participants used VBA-code to automatically construct the simulation model, but getting the VBA-code correct took much more time than he expected. Another participant had been working with the development of new features for Arena for a couple of years, and was not used to the available SIMAN constructs, so he lost a lot of time evaluating different concepts to model his vehicles, in the end he had this working, but did not have time to implement the passengers in his simulation model. The last two participants working with the Arena constructs noticed the long list of desired experiments and made sure their models were very flexible for any kind of layout and vehicle parameter setting. This flexibility lead to concepts that did not allow easy communication with passengers and this made the implementation of passengers in the simulation model very hard, a task they had not completed at the end of the experiment.

#### **Observations based on the Time Logging Form**

Most of the important parts of the log-form have already been discussed in the previous sub-section. The participants using building blocks needed a lot of time to get their simulation model working the way they wanted. The participants using the Arena constructs spent almost all their time on model development. One of the participants pointed out in his log-form that he spent 5 minutes on experimentation in the first hour, which he used to think out the different experiments he wanted to perform. In the evaluation with all participants they claimed that they all spent some time to overview the kinds of experiments they needed to perform.

#### **Outcome of the Questionnaire**

The questionnaire showed clearly that the participants were short on time and it also showed that all the participants expected they would do much better if they would have had more time. The main difference was that the participants using building blocks thought they needed 2 to 4 more

hours, while the participants using Arena constructs thought they would need 10 to 30 more hours to get the simulation study finished.

From the questionnaire it could be seen that the participants using building blocks expected that the problem owners would like their work better. They expected to be able to easily do any possible experiment and to easily visualize the simulation model in such a way the problem owner would understand what was going on. The participants using the Arena constructs were less optimistic. They assumed they could do most of the desired experiments, but had to conclude that they would ignore some of the issues, as they were not prepared for them and could not extend their simulation model in that direction.

#### **Outcome of Evaluation with Participants**

After all the participants had handed in their material, we arranged a meeting to discuss the preliminary analysis and what we had expected. During this meeting we discussed the outcomes of the participants, mainly why they did not succeed in finishing on time. The participants using Arena concepts agreed that they followed the conceptual model and the suggested experiments too much. They did not think of doing a quick and more global study first, followed by a more detailed analysis. The participants using the building blocks complained about the documentation. They received error messages, which they could not understand. They complained about a lack of quick explanations, something like a Frequently-Asked-Question list to help them through their main problems. They also complained they did not have the code, so when they encountered problems, they could not check the source code of the building blocks to see whether the developers of the building blocks had made any mistakes.

### **CONCLUSIONS OF THIS LABORATORY EXPERIMENT**

Our overall observations are that the experts using building blocks had a high conceptual mismatch and hesitation before fully trusting the building blocks using them. When we compare this with the novices in the first two laboratory experiments (adjusting a simulation model and develop a simulation model) the novices did not have an opinion of their own how to model such a system, they just did what they were asked to do. They did not have any conceptual idea about how a transportation system should work, so they did not want to understand the building blocks. Based on the desire for additional technical information, we can conclude that simulation experts need to be fully convinced of the technical superiority of building blocks before they use them, this will allow them to conclude that they are wrong

when errors are reported, instead of the building blocks. This process of producing conviction should be performed using hands-on training, additional explanations or Frequently-Asked-Question lists.

The participants using the Arena concepts wanted to show off their expertise with Arena and show they could model any conceptual model. They were convinced of the quality of their generic simulation tool so they did not want to abstract too much. However, pragmatic problem solving collided with their drive for high-quality solutions and this resulted in no solution within the time parameters of the experiment.

## FUTURE RESEARCH

Even though the outcome was not as convincing as we expected before we started the laboratory experiment, we can still conclude that building blocks show a higher efficiency for the support of problem owners using a simulation study. This laboratory experiment showed that simulation experts using building blocks achieve more results than simulation experts that start with standard simulation constructs, and we expect that the difference may even be larger. An important indicator for this is the expected time the experts would need to successfully complete the study: the building block group mentioned 2-4 hours, compared to 10-30 hours for the basic simulation group.

One of the main outcome was that the documentation and training for the experts using building blocks was not good enough, we need to convince the experts that building blocks are the concept of choice before allowing them to work with building blocks. This might sound harsh, but the main thing that these technical experts wanted, was insight into the building blocks. They experienced the building blocks as black boxes and they did not fully accept what was going on.

Finally, the participants of this laboratory experiment were mainly developers of the simulation environment Arena and not consultants that work daily in a simulation environment. Perhaps the differences between the participants using building blocks and those using Arena constructs would change if simulation consultants performed the laboratory experiment.

Both ideas, using improved technical documentation and different kind of experts, need to be tested to gather additional knowledge about the efficiency of using of building blocks in simulation studies.

## ACKNOWLEDGEMENT

We are grateful for the voluntary participation of the simulation experts from Rockwell Software in Sewickley (PA, USA) and Rockwell Automation in Rotterdam (The Netherlands).

## REFERENCES

- Banks, J. "Introduction to simulation", In: P.A.Farrington; H.B.Nembhard; D.T.Sturrock and G.W.Evans (Eds.) *Proceedings of the 1999 Winter Simulation Conference*, p.7-13., 1999
- Brandt, M. *SkyShuttle, Den Haag in beweging*, Master thesis Delft University (in Dutch), 1999
- Harrington, H.J.; K.Tumay. *Simulation Modeling Methods: to reduce risks and increase performance*, McGraw-Hill, 1999
- Hooghiemstra, J.S.; M.J.G. Teunisse. "The Use of Simulation in the Planning of the Dutch Railway Services". In: D.J. Medeiros; E.F. Watson; J.S. Carson and M.S. Manivannan (Eds.), *Proceedings of the 1998 Winter Simulation Conference*, p.1139-1145., 1998
- Law, A.M. ; D.W.Kelton. *Simulation Modeling and Analysis*, McGraw-Hill, 3<sup>rd</sup> edition, 1999
- Kelton, W.D.; R.P.Sadowski ; D.A.Sadowki. *Simulation with Arena*, McGraw-Hill, 2<sup>nd</sup> edition, 2002
- Keller, L.; C.Harrell ; J.Leavy. "The three reasons why simulation fails". In: *Industrial Engineering*, Volume 23, Issue 4, p27-31., 1991
- Robinson, S. "Three sources of simulation inaccuracy (and how to overcome them)". In: P.A.Farrington; H.B.Nembhard; D.T.Sturrock; G.W.Evans (Eds). *Proceedings of the 1999 Winter Simulation Conference*, p1701-1708., 1999
- Valentin, E.C.; A.Verbraeck. "Simulation using building blocks" In: F.J.Barros, N.Giambiasi (Eds.) *Proceedings conference on AI, Simulation and Planning*, p65-71., 2002
- Verbraeck, A.; Y.A. Saanen; E.C. Valentin. "Logistic Modeling and Simulation of Automated Guided Vehicles". In: A. Bargiela, E. Kerckhoffs (Eds.) *Simulation Technology: Science and Art. 10th European Simulation Symposium and Exhibition*, p. 514-519, 1998
- Verbraeck A.; Y.Saanen; Z.Stojanovic; B.Shishkov; A.Meijer; E.Valentin; K.van der Meer. "Chapter 2:What are building blocks?" In: A.Verbraeck and A.Dahanayake (Eds.) *Building blocks for Effective Telematics Application Development and Evaluation*, TU Delft press, 2002

# IMPLEMENTATION OF CONTINUOUS-TIME DYNAMICS IN SCICOS

Masoud Najafi

Azzedine Azil

Ramine Nikoukhah

INRIA, Rocquencourt,  
BP 105, 78153 Le Chesnay cedex, France

**KEYWORDS:** Hybrid systems, Synchronous language, DAE, Simulation software, Numerical solver.

## Abstract

Scicos is a software environment for modeling and simulation of dynamical systems. The underlying formalism in Scicos allows for modeling very general dynamical systems: systems including continuous, discrete and event based behaviors. This paper presents some aspects of the simulation software, especially the features which have to do with the hybrid nature of the models. We study in particular the implementation issues concerning the efficient use of ODE (Ordinary Differential Equations) and DAE (Differential/Algebraic Equations) solvers in the simulator.

## INTRODUCTION

Scicos is a toolbox of the scientific software package scilab [Bunks et al. 1999, Chancelier et al. 2002]. Both Scilab and Scicos are free open-source softwares ([www.scilab.org](http://www.scilab.org), [www.scicos.org](http://www.scicos.org)). Scicos includes a graphical editor for constructing models by interconnecting blocks, representing predefined or user defined functions, a compiler, a simulator, and some code generation facilities.

## Scicos Formalism

The formalism used in Scicos is based on the formalism of synchronous languages, in particular *Signal* and its extension to continuous-time systems [Benveniste 1998]. We do not give a full presentation of the formalism in this paper (for more see [Nikoukhah and Steer 1996-a, Nikoukhah and Steer 1997, Nikoukhah et al, 1999]). We simply review some aspects which specifically have to do with the continuous-time behavior to lay the ground for presenting the specific issues related to continuous-time simulation.

Even though there exists an inheritance mechanism in Scicos formalism which provides a data-flow type of behavior, Scicos is fundamentally event-triggered. This has made the continuous-time extension non-trivial. The ba-

sic idea consisted in treating the continuous-time events just like an ordinary event [Nikoukhah and Steer 2000].

Events are signals which activate system components in discrete event-driven environments. We extend the notion of event to obtain what we call an *activation signal* which consists of a union of isolated points in time and time intervals [Nikoukhah and Steer 1996-a, Djenidi et al. 2001]. Each Scicos signal is then associated with an activation signal specifying time instances at which the signal can evolve, see Fig.1.

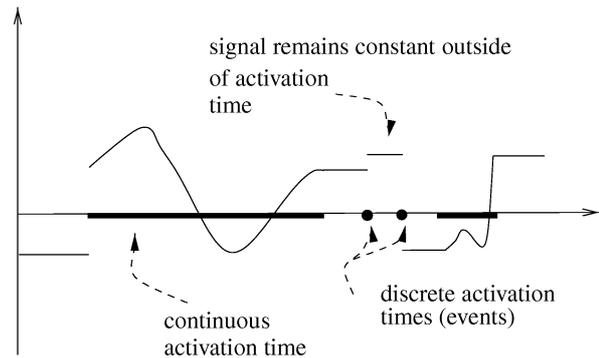


Figure 1: A typical Scicos signal. Thick line segments represent the activation times.

The fundamental assumption is that over an activation interval, in the absence of events, the signal is *smooth*. Scicos compiler propagates the activation signals through the model in order to obtain activation information about all the signals present in the system. This information is valuable for the simulator which has to properly parameterize and call the numerical solver.

It would be unrealistic to imagine that a formalism can be developed independent of the properties of the numerical solver. That is why it is important to study these properties and identify precisely what properties are important and must be taken into account in the development of the formalism and the implementation of the modeler and simulator.

## Solver properties

Scicos uses two numerical solvers: `Lsodar` [Hindmarsh 1980, Petzold 1983-a] and `Daskr` [Petzold 1983-b, Brown et al. 1998]. `Lsodar` is an ODE solver which is used when the Scicos diagram does not contain *implicit* blocks. Implicit blocks are blocks which contain implicit dynamics of the form

$$\begin{aligned} 0 &= f(\dot{x}, x, t, u) \\ y &= g(\dot{x}, x, t, u) \end{aligned}$$

where  $u$ ,  $x$ ,  $y$ , and  $t$  represent respectively block's input, state, output, and the time;  $\dot{x}$  represents the time derivative of the state  $x$ . Note that the implicit nature of the block has to do with the absence of an explicit expression for  $\dot{x}$ , the input and output are still explicitly defined. If a diagram contains an implicit block, the overall continuous-time system to be solved becomes implicit and `Daskr` is used by Scicos simulator.

The solver properties discussed here are those of the two solvers mentioned above, however these properties are common to most modern solvers. The most important of these properties is that these solvers require that the system be sufficiently smooth over an integration period. This means that Scicos simulator must make sure to stop and reinitialize the solver at each potential point of non-smoothness (discontinuity, discontinuity in the derivative, etc...).

The other important property is that these solvers are variable step, meaning the discretization is not regular in time and more importantly, the solver can take a step forward in time and then later step back so that the evaluation calls do not constitute an increasing time sequence.

Another important property is the possibility to set constraints on solver. For example constraints on the relative and absolute error tolerances can be imposed globally or on each state variable separately. Time constraint can be imposed to forbid the solver to advance time beyond a given time called the *stopping time*. Normally the solver is allowed to step beyond the final integration time and returns the value at final time by interpolation. This increases the efficiency when the integration is restarted.

Finally there are complex issues related to re-initialization as far as the implicit blocks are concerned and the use of `Daskr`. We do not discuss these issues here.

## Parameterization of the solver

Besides obvious parameterizations such as setting various error tolerances, maximum step size, etc..., the simulator must automatically start and stop the simulation when necessary.

When an event occurs, the continuous-time simulation must stop so that the event-triggered components of the system can be activated. Once this is done, the

continuous-time simulation can proceed. So events times are the times of stop and restart of the continuous-time simulation. Although they are similar in this way, the events originating from different sources, perform different tasks. Hence they have different importances and properties. In continue, two important properties of events will be introduced.

## Event criticality

Although all events force the simulator to stop and restart the integration, they are however not equal in importance. Consider Scicos diagrams in Fig. 2 and Fig. 3. In the first diagram, the event generated by the *Event Clock* drives the *Scope*. The integrator (`1/s`) receives on its regular input port the sine function which is generated by the *ever-active, sinusoid generator* block (ever-active means the block is always active<sup>1</sup>). The output of the integrator is sampled by the *Scope* at its activation times (times at which it receives an activation on its activation input port, (i.e., the times of the events generated by the *Event Clock*). In this case, the solver must be stopped at each of these times, however, since the corresponding events do not produce any non-smoothness on the input of the integrator, there is no reason to re-initialize the solver (do a *cold restart*). We say this event is *not critical*.

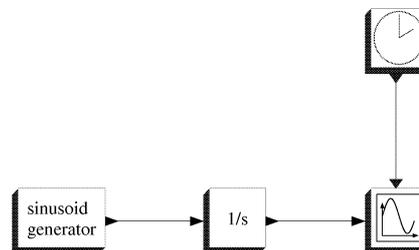


Figure 2: Non-critical event.

In the second diagram, the event activates the *random generator* block which outputs a random variable. This output remains constant until the next event reactivates the block. The integrator then receives a piecewise constant signal to integrate. Thus the event in this case creates a discontinuity at the input of the integrator. Clearly in this case the solver must be re-initialized. This event is *critical*.

Furthermore, an event can be critical in another way: an event can cause a jump in the internal state of a block. One such example is the integrator with reset on event. Clearly if a jump is produced in the state, the solver must be restarted cold.

<sup>1</sup>The activations in Scicos, in general, come through activation signals via activation ports, however, for the sake of simplifying diagram construction, ever-active activations are not explicitly drawn.

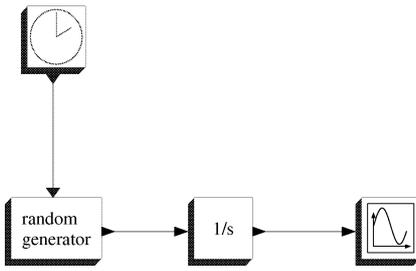


Figure 3: Critical event.

### Event predictability

We have seen that events can be put into two categories: critical and non-critical. But they can also be put into two other categories: *predictable* and *non-predictable*. Consider the following system:

$$\dot{x} = \begin{cases} x\sqrt{1-t} & \text{if } t \leq 1 \\ 0 & \text{if } t > 1 \end{cases}$$

To model this system in Scicos, we need to generate an event at time  $t = 1$  in order to change the dynamics of the system. Clearly this event is critical. See Scicos diagram illustrated in Fig. 4.

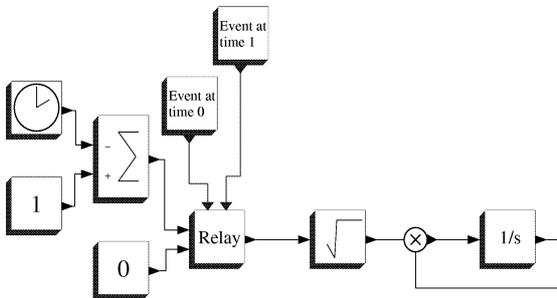


Figure 4: Predictable event.

But this event is also predictable. We know ahead of time its occurrence time ( $t = 1$ ). In general, most events in a Scicos diagram are generated by *Event Clocks* and their times are predictable.

Non-predictable events are *zero-crossing* events. These events are generated when a signal crosses zero and their activation times are not known in advance. A predictable event can be considered and modeled as a non-predictable event. For example in the above example, the event at time 1 can be obtained by using a zero-crossing test on the function  $1 - t$ . But that would be inefficient for two reasons: first, the zero-crossing tests are additional work for the solver. Second, to detect a

zero-crossing, the solver has to go beyond the crossing and perform iterations to pin-point exactly the location of the crossing. In the above example, this results in an error since for  $t > 1$ ,  $\sqrt{1-t}$  is not defined. See Scicos diagrams illustrated in Fig. 5 and Fig. 6. In the Scicos diagram illustrated in Fig. 5, the Zcross block defines a zero-crossing surface. The solver has to go beyond the surface in order to find the exact time of crossing. That is why it attempts to evaluate the value of  $\sqrt{1-t}$  for  $t$  larger than 1. The same system is implemented in a slightly different way as illustrated in Fig. 6. Here, the *if-then-else* block redirects its activation signal to one of its output activation ports depending on the value of its regular input. If this latter is positive, the activation goes through the *then* port, if not, it goes through *else*. The *if-then-else* and the *event-select* blocks are the only blocks which redirect activation signals without creating delays. These blocks are referred to as *Synchro blocks*. They are the counterparts of conditional statements *if-then-else* and *switch* in programming languages such as C. *Synchro* blocks have built-in zero-crossing surfaces that generate an event when the activated output port changes. Because such a change may produce discontinuity and thus the solver must be informed. The presence of this zero-crossing forces the solver to step beyond the  $t = 1$ , and as a result, the simulation fails again.

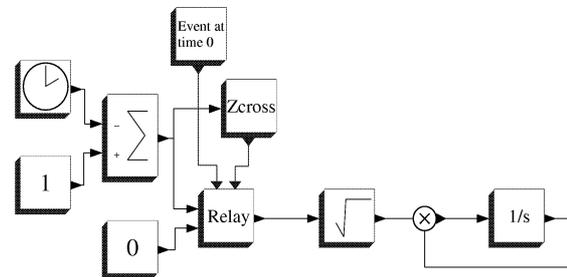


Figure 5: The simulation fails in this case.

If the event is treated as predictable, the simulator, using the fact that the upcoming event at time 1 is critical, sets the critical time to 1 preventing the solver to step beyond  $t = 1$ .

### Critical event classification

In the previous sections the importance of Critical events has been shown. Here the definition, the classification algorithm, and its applications in simulation will be discussed.

**Definition 1** A discrete event is critical if upon activation, either an input of a block, containing zero-crossing

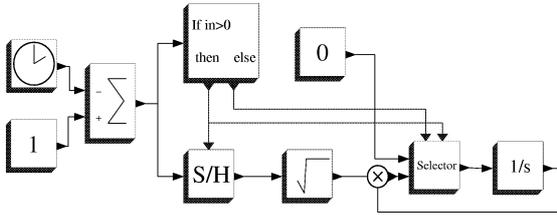


Figure 6: The simulation fails also in this case.

surfaces or continuous-time states, is updated, Or the continuous-time states of a block jump.

Before explaining critical event classification algorithm, the following items should be noted.

- The critical property concerns only discrete events. Therefore to avoid considering a continuous-time event as critical, all pure continuous-time activation links must be excluded from the classification procedure.
- Block having direct input-output relationships (direct feed-through) and ever-active blocks can pass on the discontinuities arriving at their inputs to the following blocks. These blocks are called DTB (Discontinuity transmitting block).
- Synchro blocks that do not have an input event port and inherit the continuous-time activation, are referred to as Continuous Synchro (CS) blocks.

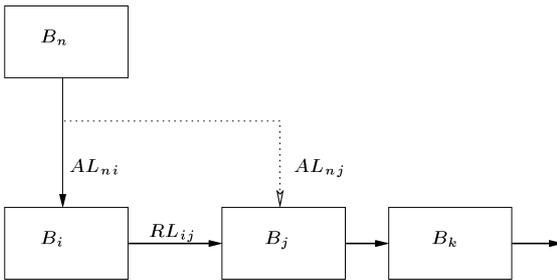


Figure 7: Event inheritance

#### Algorithm:

1. Propagate the events through DTB's to determine the blocks that potentially subject to discontinuity at their inputs. For example in Fig. 7, IF  $B_n$  block activates  $B_i$  block (here i.e.  $AL_{ni}$ ), AND IF a regular link between  $B_i$  and  $B_j$  blocks exists (i.e.  $RL_{ij}$ ), AND IF the  $B_j$  block is a DTB, THEN an explicit activation link between  $B_n$  Block and  $B_j$  block (i.e.

$AL_{nj}$ ) is established<sup>1</sup>. Repeat this process until no more activation link can be established.

2. Find all CS blocks (Synchro blocks not activated explicitly) and store them in CS-list.
3. Identify all the blocks activated by CS blocks and add them to the list of DTB's. For example, in Fig. 8, the  $B_j$  block is activated by  $CS_n$ . so it should be added it to the list of DTB's. The reason is that the block  $B_j$  receives a continuous-time activation through  $CS_n$  therefore it can pass the eventual discontinuities at its input port into the  $B_k$  block.

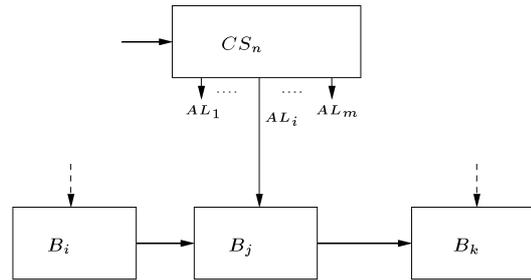


Figure 8: Continuous Synchro Block

4. Remove all activation links originating from CSs (e.g. in Fig. 7 all  $AL_i$  links) and then remove CSs from the diagram.
5. Go to step 1 and repeat until in step 2 CS-list becomes empty.
6. For each event source, search through all the blocks it activates. If any of them contains continuous-time states or zero-cross surfaces, then flag the event as critical.

At later stages of compilation, Synchro blocks (i.e., *if-then-else* and *event-select*) are duplicated if they have multiple sources of activation. So each Synchro, at the end, is activated just by one activation source [Nikoukhah and Steer 1996-a, Nikoukhah and Steer 1997]. This process is done after critical event classification and the newly generated events inherit the property of the original events.

#### Use of critical event classification in simulation

The classification of the events allows us to avoid unnecessary cold restarts. Without it, at every event time, the solver should be cold restarted to make sure that the numerical solver uses consistent information.

<sup>1</sup>All the changes made in the diagram are discarded at the end when the critical events are identified so that other phases of compilation can be carried out normally

The lack of consistency is avoided in two ways. If the critical event is fired immediately, a cold restart is performed. Because in such a case, a discontinuity may occur in a signal (or in its derivative) fed to a block containing continuous-time states or zero crossing surfaces. This discontinuity may cause the solver to fail during restart. This is particularly a problem in Scicos because Scicos uses a BDF (backward differentiating Formula) method to integrate the continuous-time systems.

Another situation where inconsistency can occur is when a critical event is programmed for a future time. If the solver had been allowed to look beyond this time, the new critical event may affect values which are already used by the solver, i.e., values beyond this time. This change of model can result in a failure of the solver which expects to receive consistent informations. To see more precisely what happens in this case, note that after each event at time  $t(i)$ , when the integration resumes, the start-time  $t(i)$ , the end-time  $t(i + 1)$ , and the time beyond which the solver is not allowed to explore  $t_{\text{stop}}(i)$  are passed to the solver to integrate the system from  $t(i)$  to  $t(i + 1)$ . Note that the solver may advance the time beyond  $t(i + 1)$  and compute the solution at  $t(i + 1)$  by interpolation. Let  $t_{\text{max}}$  be the largest value of  $t$  for which the solver has requested an evaluation of the function; clearly  $t(i + 1) \leq t_{\text{max}} \leq t_{\text{stop}}(i)$ . After the processing of the event(s) at time  $t(i + 1)$ , the integration resumes from  $t(i + 1)$ . If the events at  $t(i + 1)$  program a critical future event at a time  $t_c$  earlier than  $t_{\text{max}}$  then the function evaluations done in the previous integration period are no longer valid. One way to make sure that such a situation does not occur is to force a cold-restart if  $t_c < t_{\text{stop}}(i)$ . Note that in this case we set  $t_{\text{stop}}(i + 1) = t_c$  if not  $t_{\text{stop}}(i + 1) = t_{\text{stop}}(i)$ .

## Simulation optimization

To each event, we associate a list specifying the blocks which are to be activated when the event fires and the order in which they should be activated.

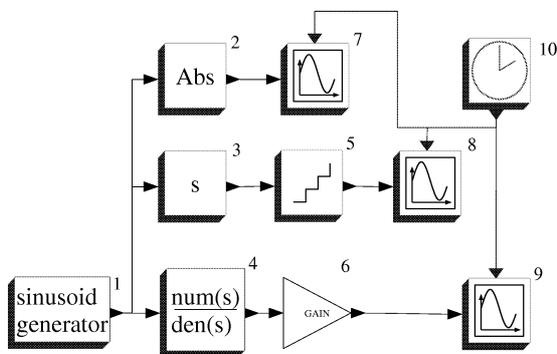


Figure 9: Execution order.

The same holds for continuous-time activations (as it was said before, Scicos treats the continuous-time activation similarly to the way it handles discrete events). A continuous-time activation is a call to the system to update its continuous-time components. For that, there is an activation list and an order in which the activations must be realized. We call it *CORD* list.

For example, in Fig. 9, to update the system at the requested times, the blocks  $\{(1), (3, 4), (2, 5, 6)\}$  should be called to update their outputs. During the continuous-time simulation (when the solver is at work), the 3, 4 blocks are called to deliver their updated residuals  $(f(\dot{x}, x, t, u))$ . Note that these blocks are the blocks in *CORD* which contain continuous-time states.

Note that the *CORD* can be used both for updating the continuous-time parts of the system at specific times and also during the integration by the solver. So unlike discrete events, continuous-time events are evoked in three distinct situations:

1. In the simulator, to update the continuous-time parts at a requested time.
2. By the differential equation solver, to update the continuous-time parts so that system residuals can be correctly evaluated.
3. By the differential equation solver, to update the values of the zero-crossing surfaces.

It turns out that in the second and third cases, we don't necessarily need to activate all the blocks in the *CORD* list. In the second case, we only need to update the outputs of blocks which affect directly or indirectly the input of a state bearing block. Same for the third case except that blocks containing zero-crossing surfaces must be considered. For example in Fig. 9, the 2, 5, 6 blocks which are in *CORD* don't have any effect on the residuals. Thus, they can be excluded from *CORD* and be stored in a new list. The new list, *OORD* list, is comprised of all the blocks which their outputs may affect the input to a block containing a continuous state. The blocks affecting the zero-crossing surfaces (in this case, *Abs* and the *Quantizer* blocks) are 1,2,3,5. We call this list *ZORD*.

The use of *OORD* and *ZORD* optimizes the number of calls to each block in the integration phase. This optimization is very important for simulation efficiency because the solver requires many function evaluations (depending on the stiffness/nonlinearity of the system and the required precision). In a typical application, for every evaluation due to a discrete event, the solver requires hundreds or even thousands of continuous-time function evaluations.

## Conclusion

In this paper, we have presented the mechanism by which properties of events (critical, predictable) are extracted

automatically and used to generate an appropriate parameterization for the numerical solver. We also discussed a technique for increasing simulation efficiency.

## References

- [Bunks et al. 1999] C. Bunks, J. P. Chancelier, F. Delebecque, C. Gomez (ed.), M. Goursat, R. Nikoukhah and S. Steer, *Engineering and Scientific Computing with Scilab*, Birkhauser, 1999.
- [Chancelier et al. 2002] J. P. Chancelier and F. Delebecque and C. Gomez and M. Goursat and R. Nikoukhah and S. Steer, *An introduction to Scilab*, Springer-Verlag, 2002.
- [Benveniste 1998] A. Benveniste, *Compositional and Uniform Modeling of Hybrid Systems*, IEEE Trans. Automat. Control, AC-43, 1998.
- [Nikoukhah and Steer 2000] R. Nikoukhah and S. Steer. *Conditioning in hybrid system formalism*, ADPM, Dortmund, Germany, Sept. 2000.
- [Hindmarsh 1980] A. C. Hindmarsh, *Lsode and Lsodi, two new initial value ordinary differential equation solvers*, ACM-Signum Newsletter, no. 4, 1980.
- [Petzold 1983-a] L. R. Petzold, *Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations*, SIAM J. Sci. Stat. Comput., No. 4, 1983.
- [Petzold 1983-b] L. R. Petzold, *A Description of DASSL: A Differential/Algebraic System Solver*, in *Scientific Computing*, R. S. Stepleman et al. (Eds.), North-Holland, Amsterdam, 1983.
- [Brown et al. 1998] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Consistent Initial Condition Calculation for Differential-Algebraic Systems*, SIAM J. SCI. COMP., NO. 19, 1998.
- [NIKOUKHAH AND STEER 1996-A] R. NIKOUKHAH AND S. STEER, *Scicos a dynamic system builder and simulator*, IEEE INTERNATIONAL CONFERENCE ON CACSD, DEARBORN, MICHIGAN, 1996.
- [NIKOUKHAH AND STEER 1996-B] R. NIKOUKHAH AND S. STEER *Hybrid systems: modeling and simulation*, COSY: MATHEMATICAL MODELLING OF COMPLEX SYSTEM, LUND, SWEDEN, SEPT. 1996.
- [NIKOUKHAH AND STEER 1997] R. NIKOUKHAH AND S. STEER, *Scicos: A Dynamic System Builder and Simulator; User's Guide - Version 1.0*, INRIA TECHNICAL REPORT, RT-0207, JUNE 1997.
- [NIKOUKHAH ET AL,1999] S. STEER, R. NIKOUKHAH, *Scicos: a hybrid system formalism*, ESS'99, ERLANGEN, GERMANY, OCT. 1999.
- [NIKOUKHAH AND STEER 1999] R. NIKOUKHAH AND S. STEER, *Hybrid system modelling and simulation*, FEM-SYS'99, MUNICH, GERMANY, MARCH 1999.
- [DJENIDI ET AL. 2001 ] R. DJENIDI, R. NIKOUKHAH AND S. STEER, *A propos du formalisme Scicos*, MOSIM'01, TROYES, FRANCE, APRIL 2001.

# Multi-Agent Simulations of Evolution and Speciation in Cichlid Fish

Ross Clement

Department of Artificial Intelligence and Interactive Multimedia  
Harrow School of Computer Science, University of Westminster  
Northwick Park, Middlesex HA1 3TP, UK  
Email: clemenr@wmin.ac.uk

## ABSTRACT

An agent-based simulation has been built to model speciation in cichlid fishes in the Great Lakes of Africa. A real natural system has been chosen as the target of simulation, rather than a generalised system. This focusses research towards open problems in cichlid biology, and provides a library of field research to drive the design and parametrisation of the simulation. Visualisations of the end results of simulations are presented to confirm that the simulated fish actually do speciate. A further experiment suggests that the potential for organisms to adapt has strong effects on the competitive exclusion principle, but that this cannot be used to explain the patterns of cichlid species found on rocky reefs in African lakes. The simulation has been written in pure Java rather than a general agent-based modelling platform. The reasons for using Java and a number of alternative platforms are described.

## INTRODUCTION

This paper describes the use of a purpose built multi-agent simulation system (TDLP – “The Digital Lake Project”) to investigate open problems in the speciation of cichlid fish in the African Great Lakes. Previous simulation systems used to investigate speciation (e.g. Kondrashov & Kondrashov 1999; Dieckmann & Doebeli 1999; van Doorn & Weissing, 2000) attempt to find 'general' rules for speciation, by modelling highly abstract ecologies. In general, these simulations model three properties of individuals: trophic preference/adaptation (preferred foods), some sexual signalling phenotype (e.g. colour), and a sexual preference (e.g. the preferred mate colour of a female). These simulations were typically used to support the contentious concept of *Sympatric speciation* (Via, 2001; Turelli *et al*, 2001). Sympatric speciation is a form of speciation where a single ancestor species will divide into two (or more) daughter species, without any division of the ancestor species into geographically isolated subpopulations. TDLP originally was an implementation of the aforementioned models which added geographical details, to allow the investigation of conditions under which sympatric speciation would, and would not, occur. It has since evolved into a much larger and more detailed system, intended to study speciation in a real natural ecology.

The cichlid fish have been chosen as the biological system to be modelled by TDLP to allow cichlid field biology to inform the choice of model form and parameters. Also, it is by no means certain that all organisms speciate by the same methods. E.g. Sturmbauer (1998) discusses possible effects of the

specific environment found in the African Great Lakes on cichlid speciation. Comparing and contrasting what is known about cichlid speciation and the biology of the galaxid fish of New Zealand suggests that speciation occurs by different methods. Diandromous (living part of their lives in salt water and part in fresh water) Galaxids appear to speciate when they become land-locked. The strong effect of major geological events on galaxid evolution can be seen in genetic research such as that in (Waters *et al*, 2001). Cichlids do not appear to need such an obvious physical barrier for speciation to occur, with the best example being the cichlids of Lake Barombi-Mbo (Schliewen *et al*, 1994), where speciation has occurred in a small, smooth walled, crater lake. These cichlids are often quoted as being the most likely example of real-world sympatric speciation. Making the choice of cichlids allows Cichlid biology (e.g. Barlow, 2000) to be used to guide simulation design and choice of parameters, removing the ambiguity (and, we believe, insolvable problem) in choosing parameters for a 'generalised' system. Additionally, these biological studies provide a benchmark with which to compare the performance of TDLP. A relatively recent survey of cichlid biology (and open problems) can be found in (Coleman, 2001).



Figure 1: A Typical Cichlid from Lake Victoria<sup>1</sup>

Multi-agent system are frequently used in the study of ecological system (e.g. Mamedov & Udalov, 2002, Parrott & Kok, 2002). These systems are frequently presented as general systems useful for a wide variety of experiments. However, any such systems invariably make design choices for the users, with the most obvious example being that ecological simulation systems typically hard-code the names of species to which individual agents belong. This is clearly inappropriate for a system intended to model speciation, where the number and form of species will change over time, and pass through situations where species boundaries are not clear-cut.

<sup>1</sup> (c) 2003 M Pederson. (<http://www.cichlidrecipe.com>).  
Used by permission.

In the remainder of this paper, the current state of TDLP is described, and its use is illustrated by two experiments. First, we demonstrate the speciation (one ancestor species splitting into two) is occurring during 'typical' runs of the simulation, and show that the resulting groups of fish are what would be typically called 'different species'. To show the generality of the system, further experiments investigating the relationship between the speed by which fish can evolve, and the speed by which competitive exclusion occurs, are described.

## CHOICE OF PLATFORM

TDLP is implemented in Java, the choice of language being made to allow the maximum degree of flexibility during the development and evolution of the project itself.

Considering potential agent-based platforms, we survey the systems of Mamedov & Udalov (2002), Parrot and Kok (2002). The system of Mamedov & Udalov (CENOCON) is "a computer tool to build individual based models for simulation of population interactions". Even at this high abstract level, there is a clear mismatch between the application domain of CENOCON, and that of TDLP. That is, modelling an ecology of (essentially) unchanging organisms in CENOCON, and modelling an ecology of changing (through evolution) organisms in TDLP. In CENOCON, animals are modelled by 38 individual properties, including detailed information about food preferences, growth parameters, body fat levels, gut capacity, and many others. These properties are all relevant to evolution, but the system as a whole lacks a genetic (or other) basis for allowing these to change over time. Animals in CENOCON are also labelled with a species name, which is entirely incompatible with the purpose of TDLP, where species are expected to emerge from a single-species population as a natural consequence of their ecology, and the environment they find themselves in.

Parrot & Kok's unnamed generic modelling system (hereafter PK) has many similarities to CENOCON, in that individuals are modelled in terms of digestion, body composition, food preferences, and species are again fixed and labelled with a species name. Over 100 properties are used to model animals, though these include dynamic properties such as the direction an animal is moving at any particular time. Again, the possibility of the generic properties that define an organism changing over time is not allowed (as befits the aim of modelling ecologies).

TDLP combines both ecological modelling (as we wish to measure the effect of ecological factors on speciation and evolution) and a varying genetic model. In terms of the sophistication of the ecological factors modelled, both CENOCON and PK are superior to TDLP, and may well be used to guide the future development of TDLP in this area.

If high-level biological/ecological modelling systems are not appropriate for investigating African cichlid biology, then an alternative is to use a more general agent-based modelling framework such as RePast. In some ways, the

arguments on whether to use a system such as RePast are due to learning curves, and the perceived (rather than actual) advantages of RePast. The central motivation of this research is to learn as much as possible about the biology of the cichlids of the African great lakes, not to work on agent-based modelling itself. Hence, the choice of implementation was between the known (Java) and the unknown (Agent tools), and driven by a high degree of motivation to dive straight into the biological details. RePast uses a similar method for performing time-based updates to TDLP, with event the name of the *step()* method being the same. However, agents in TDLP are stored at differing abstract levels. Hence, time does not pass by a high level co-ordination agent repeatedly calling the *step()* method of individual fish. An intermediate level class (*Population*), not clearly matching the definition of either an agent or a container class, is called by the true container class (*Environment*) once per time period. The *Population* class then calls a number of methods of individual *Fish* agents, requesting single steps in such features as metabolism, feeding, and breeding. While it is possible to implement structures like this in RePast (as the classes in RePast can be sub classed and used alongside any other Java classes), there is still insufficient confidence that RePast would be a significant advantage over pure Java. For example, the amount of code in TDLP concerning generic agent functions is a small percentage of the total. This differs from smaller and simpler models (which have important roles in biological and other simulations) where the proportion of generic code is higher.

An alternative to using Java would have been the adoption of a specialised agent-based modelling platform (Gilbert & Bankes, 1999) such as RePast (<http://repast.sourceforge.net/>). The decision to use Java rather than a system such as RePast was mainly driven by the author's long experience with Java, and lack of experience with specialist platforms. A further factor was a lack of confidence that existing platforms would be flexible enough to handle future developments in the simulation system. At the time the choice of implementation language was made, even the nature of these future developments (such as the inclusion of the simulation of behaviour) was unknown. Research in the field biology of African Cichlids (e.g. Kocher, 2003) throw up metaphorical 'curve balls' affecting the future course of TDLP. Work on BTEExact's iCRM system for modelling the effects of policy on Customer Relationship Management (Baxter, 2003) initially used RePast, but later switched to Java to allow finer control on the simulation, including the user interface (Baxter, *priv. comm.*). It is expected that TDLP will be reimplemented from scratch in the near future, but at present there is not sufficient evidence to choose between the extremes of reimplementing in C or C++ to improve speed, or to use a dedicated agent platform to improve ease of simulation design and implementation. However, experience so far suggests that programming itself is not a bottleneck limiting experimentation.

Ginot *et al* (2002) describe MAS, another ecological multi-agent simulation system. They discuss the importance of making the agent-based simulation

platform available to non-programmers. In the specific case of TDLP, the author of this paper has far greater experience and knowledge of programming (particularly in Java) than in cichlid biology. Hence, without an intended audience of non-programming biologists, another of the major advantages of generic platforms does not apply.

Given that no generic approach to agent modelling appears perfectly matched to the aims of TDLP, it might seem attractive to see TDLP as the first prototype of a new, generic agent-based platform for investigating speciation problems. This approach is criticised in the "Conclusions and Future Work" section of this paper.

TDLP simulates an ecology in discrete time steps roughly approximating one week each. A side project examined the possible role of social learning behaviour in partially dividing populations as a precursor to speciation (Clement, 2003a). Behaviour occurs (and can be learnt) in time periods much quicker than a week, and future version of TDLP may need to be rewritten to work on much finer grained time scales. Currently, the system is not spatially explicit other than modelling isolated populations in rocky reefs (as occurs in all of Lakes Victoria, Tanganyika, and Malawi). However, on these reefs, species tend to segregate according to water depth (e.g. Seehausen, 1996; Konings 2001). Models that investigate this characteristic of lake ecology will need to be investigated in the future, requiring more detailed spatial representations.

The system makes extensive use of Java interfaces (Sun Computers, 2003). Much of the code accesses abstract agents (such as a generic *FoodSource*) expressed by interfaces. An interface defines a set of methods (including their arguments and return types), and any class (here agent) written that includes these methods (and declares itself to do so), can be manipulated by the code. This allows agents to be substantially changed (or replaced by new agents), without forcing across-the-board rewrites of the code, and allowing configuration files to load different types of agents fulfilling similar roles (e.g. *Constant food source* agents and *Regenerating food source* agents), with the remainder of the system (e.g. *Ecology* and *Fish* agents) being agnostic as to the inner workings of the *Food Sources*.

It is not intended that the system will become infinitely variable. There is a danger that in trying to build a general-purpose system, that research may enter an open-ended development phase without ever achieving results proposing interesting biological concepts. Focussing on one specific natural system helps avoid this problem.

Visualisation of results is a particular problem in simulation. Species are a naturally fuzzy concept, in that there is no generally agreed definition of the concept of a 'species'. (See (Paterson, 1993) for a discussion on defining species.) And even when individual species concepts are applied, real organisms may not fit these categories exactly, let alone fit them exactly during the process of speciation. This creates problems for tracking the process of speciation over time. The research in this paper concentrates on 'final results', i.e. the set of species

present when the biological system being modelled reaches an equilibrium. Work on detailed visualisation of the process (rather than just the result) of speciation (using fuzzy sets to represent species) is being undertaken in parallel to the main simulation system, and has been reported in (Clement, 2003b). Current research (this paper included) tends to use ad-hoc visualisation techniques, which often depend on properties of the individual simulation. E.g. if a food source is placed at an abstract co-ordinate (0.0) in a single dimensional 'trophic space', and another is placed at (1.0), then graphing the trophic adaptations of fish in a population can show lack of speciation (one group, presumably adapted near (0.5)), or speciation (two groups, one near (0.0) and one near (1.0)). However, these approaches do not adequately visualise the generation of large numbers of species, nor to co-ordinate systems higher than two dimensions.

The system is highly agent-orientated, over and above the typical 'individual-based' simulations often used in Biology. As well as data about individuals, the actions of the simulation are devolved to agents initialised according to a configuration file. E.g. an agent satisfying the '*Food Source*' interface supplies methods by which energy from that food source is supplied to fish in the local population. This is to allow radically different simulations to be performed by loading different types of agents when the system is configured.

The configuration file, like much of the system, reflects the system's growth from a much smaller program. Agent parameters are described in an ad-hoc language, with multiple 'files' included in a 'meta-file' capable of storing multiple experiments and sharing components between experiments. It is planned that this file format will be replaced by an XML format in the future.

The following sections describe the simulation system by describing each of the agent types in turn.

## Fish

The basic living entity in the simulation is a fish (prey such as algae, or invertebrates, are modelled as the more general *Food Source* agent type). Each fish is modelled by a number of parameters defined by a 'lineage'. Lineage parameters are *lineage name* (basic type such as *cichlid*, *catfish*, or *cypriniform*), *sublineage* (a species name), *size at birth*, *maximum size*, *growth increment*, *typical lifespan*, *lifespan standard deviation* (hereafter *stddev*), *sexual maturity time*, *breeding interval*, *initial energy supply*, *maximum energy (food) store*, *minimum (starvation) energy store*, *minimum energy reserves if breeding*, *typical brood size*, *brood size standard deviation*. Each of these properties is described in the following section:

- *Lineage name*. As described elsewhere, this allows the specification of species that can interbreed (same lineage name) and those that cannot (different lineage name).
- *Sublineage*. This parameter (e.g. species name) is not used in the simulation, other than during output of results.
- *Size at birth*. The size of fry when born. This is

specified in a generic 'unit', where 1.0 means a full-grown adult fish.

- *Maximum size*. The maximum size of an adult fish. Growth stops when this point is reached.
- *Growth increment*. The amount that a growing fish grows per time step.
- *Typical lifespan*. The average lifespan of a fish that dies of old age (which will be quite different from the real average lifespan when factors such as starvation are considered).
- *Lifespan standard deviation*. The standard deviation of the typical lifespan. These two parameters are used to choose random numbers from a normal distribution to assign lifespans to newly born fish. Fish reaching this age (having not died previously for other reasons) die of old age.
- *Sexual maturity time*. The age (in time steps) at which fish become sexually mature.
- *Breeding interval*. The interval between females breeding (contingent on sufficient energy being available)
- *Initial energy supply*. The amount of energy that a fish is born with. Removed from the mother's energy supply to prevent breeding being an (unrealistic) creator of energy in the model.
- *Maximum energy supply*. The maximum amount of energy that a fish can store.
- *Minimum energy store*. The minimum amount of energy that a fish needs to be storing before it starves to death.
- *Minimum energy reserves if breeding*. The minimum energy that a fish may have if it is to breed (females only, males only have to be alive).
- *Typical brood size*. The brood size of a fish. I.e. how many offspring are created when a female breeds.
- *Brood size standard deviation*. The standard deviation in brood sizes.

These properties control the ecological properties of the Fish themselves. These properties are typically identical across simulations, with generic values chosen for each, to approximate the small omnivorous rock dwelling cichlid *Cynotilapia afra*. E.g. parameters used in experiments for this paper are:

- *lineage* cichlid
- *sublineage* cynotilapia\_afra
- *lifespan* 100
- *lifespan\_var* 20
- *breedinginterval* 10
- *firstbreeding* 80
- *spawn* 10
- *maxsize* 1.0
- *initialsize* 0.03
- *growthincrement* 0.01
- *initialfood* 0.1
- *maxfood* 8.0
- *minfood* 1.4
- *minbreedfood* 4.0

In addition to these hard-coded, numerical, properties, the lineage also defines the plug-in models that control the

three phenotypes (*trophic adaptation*, *sexual phenotype (male colour)*, *(female) sexual preference*). These are the same properties modelled in the generic sympatric speciation experiments of Kondrashov & Kondrashov, and others. In the full modelling system, a number of plug-in models can be used for phenotypes, though the only models used at present are a *continuous phenotype model*, where the phenotype is represented directly as an  $N$ -dimensional real number, and the *genotype model*, where the model represents genes and chromosomes, potential mutations and their average frequencies, and a rule base to convert diploid genotypes to  $N$ -dimensional real number phenotypes. Gender of fish is also set by some model, which for all the experiments described in this paper is a *random* selection of gender with equal probabilities for male and female offspring. The parameters for cichlids used in experiments are:

- *trophicphenotype* continuous trophic.txt
- *sexualphenotype* continuous trophic.txt
- *preferredphenotype* continuous trophic.txt
- *gender* random

file trophic.txt

- *location* 0.5 0.5
- *var* 0.1

Indicating a single continuous phenotype (location (0.5,0.5), var 0.1, stored in a separate location) for initial values for the three phenotypes controlling *trophic*, *sexual*, and *preference* factors of the fish, and *gender* is random male or female.

Lineages, and sub-lineages, are important concepts for modelling African cichlids. Many African cichlids (especially those in the very young Lake Victoria, see (Seehausen, 2002)) are extremely closely related, and can easily mate and produce fertile offspring. It is thought possible (e.g. Goldschmidt, 1996) for several species of cichlids to merge into a single species, containing all the genetic variability of the original species. Hence, when creating an ecology, we may want to create closely related species which have different initial properties, but have the ability to interbreed, in certain circumstances (such as muddy water with poor visibility so that females cannot choose appropriate mates). This can be simulated by creating lineages with the same lineage name, but otherwise independent properties. On the other hand, it is sometimes desirable to create species that can never interbreed, no matter what the circumstances (e.g. *cichlids* and *catfish* in the competitive exclusion experiments described in this paper.)

The simulation program reads the model from a text configuration file, and creates the initial population. The simulation then proceeds from time 0, to a defined ending time. At each time step (approximating one week each), the population agent calls methods of the fish agents to simulate various life processes and events. At each step, each fish metabolises a unit of energy equal to its size (i.e. 1 unit of energy is sufficient to support one standard size fish for one week). If energy on hand falls below 0, then the fish starves. Otherwise, if the fish has reached

the end of its natural lifespan, it dies from old age. If the fish survives, is female, is due to breed (has reached sexual maturity, and has reached the start of a new breeding period), and has sufficient energy reserves, then

$$p(\text{fem chooses } m) = \frac{f(|\text{fem.preference} - m.\text{sexual}|, \text{fem.preference}, \sigma_{\text{fem.preference}})}{\sum_{m' \in \text{Males}} p(\text{fem chooses } m')} \quad (1)$$

where; *fem* is the female breeding, *m* and *m'* are potential (male) mates, and  $\sigma_{\text{fem.preference}}$  is the stddev in the females sexual preference (i.e. how choosy the female is),  $|\text{f.preference} - m.\text{sexual}|$  is the Euclidean distance between the female's preference phenotype and the male's sexual phenotype (expressed as *N*-dimensional points), and  $f(x, m, s)$  is the standard function defining a normal distribution of mean *m* and standard deviation *s*. The phenotypes (and possibly genotypes) of offspring are chosen in a plug-in-dependent manner). E.g. for *continuous phenotype models*, the values of the phenotypes of the father and mother are averaged, and a random factor (generated at random from a normal distribution centred at 0, and with a standard deviation of *breeding sd*) added to represent variability in young. See (Rice, 1998) for an example of modelling evolution with continuous numeric phenotypes. The fish then broods young (holds the eggs, then young in its mouth, as is common in African cichlids, losing the energy expended in creating these young), and releases them into the general population. *Genotype models* create haploid gametes from each the mother and father, including crossover, and then creates diploid offspring as per standard diploid genetics. Note that almost all of this behaviour is performed within the *Fish* agent. However, *Fish* agents are aware of other fish in their own population, e.g. when choosing mates and producing offspring.

Assignment of energy from food sources to fish is a function of the food source agent, and is described in §2.2. The interface between fish objects and food source objects is quite minimal. The fish must expose an *N*-dimensional phenotype describing their trophic adaptation in an *N*-dimensional trophic space. This phenotype can be based on any underlying representation. Also, the fish is offered energy from a food source, and return the amount eaten (in case their stomachs become full, and the amount eaten is less than

it breeds. The female chooses a mate from among the available (in the same location, of the same lineage, and male) mates according to the following probabilities:

the amount available).

### Food Sources

Food sources are represented by an interface, with the underlying food source being chosen in the configuration file. *Continuous food sources* supply a fixed amount of energy to fish in the population every time interval. Each food source is assigned a location (a *N*-dimensional point) in the 'trophic space' so that it can be evaluated how suitably adapted fish are for harvesting the food source. *Continuous food sources* also include a *random stddev*, indicating the degree to which a fishes success in harvesting food is due to luck. This factor was necessary to prevent unrealistic scenarios such as a population of fish growing, then all fish in the population dying at once when the fish received equal (and insufficient) amounts of energy. Other types of food source have been implemented, such as *Regenerating food source*, where the amount of energy available in each time period depends on the amount remaining from the previous time period. Only *continuous food sources* have been used in the experiments described in this paper.

An example food source is:

- *name* diatoms
- *amount* 40
- *foodsigma* 0.4
- *randomsigma* 0.1
- *location* 0.0 0.0

It is assumed that fish compete for the energy available from food sources, and that fish better adapted to the food source (both the fishes trophic adaptation and the 'location' of the food source in this space are *N*-dimensional numbers) obtain proportionally more energy from the food source. Available energy is given out according to the relative competitive advantage of the fish as follows:

$$\text{competitive}(fs, fish) = \frac{f(|fs.location - fish.trophic|, 0, \sigma_{fs.variance}) \times N(1.0, \sigma_{fs.random})}{\sum_{fish' \in \text{Fish}} \text{competitive}(fs, fish')} \quad (2)$$

Note that  $N(1.0, \sigma_{fs.random})$  represents a normally distributed random number representing the degree to which a fish has been lucky or unlucky with respect to the food source in question. It is assumed that  $\sigma_{fs.random}$  is small, and that  $N(1.0, \sigma_{fs.random})$  will never generate a negative number.

Two types of food sources are available for us at present. *Constant food source(s)* assume that there is a constant supply of energy from the food (e.g. *diatoms*) at all times. This is not realistic, as constant grazing of fish on diatoms will reduce the number of diatoms, and restrict their ability to breed and produce more diatoms for the next time cycle. In reality predator-prey relationships are complex, and can produce cyclic, and unpredictable

behaviour (e.g. random walks to extinction). Current experiments use *Constant food source(s)*, as is typical (explicitly, or implicitly) in much work on simulating evolution). However, a *Regenerating Food Source*, where rules relate the amount of a food source remaining at the end of one time period to the amount remaining at the next, has been implemented. Initial experiments with this food source type show that common patterns in predator/prey population dynamics can be reproduced.

### Populations

A population is just a set of fish, and the actions performed by a population agent typically just call methods of the fish in that population. E.g. a population may call the *breed()* method of all fish (male and female, mature and immature) and let the fish themselves decide what will and will not happen. Configuration parameters control the lineages of fish present in the population, as well as the total initial number of fish. This is modelled by a single parameter for the total number of fish, and weights for each lineage present. E.g. if there are 60 fish, and weights of 0.4 for *cichlids* and 0.6 for *catfish*, then 24 *cichlids* and 36 *catfish* will be present in the initial population. The population sizes for the experiment are described on a per-location (roughly corresponding to a rocky reef) basis.

### Locations

A location represents one reef (rocks, not coral). It is modelled as a set of food sources, a population, and a set of probabilities describing the likelihood that a fish will migrate to another location during a single time period. Like population agents, location agents basically perform few operations of their own. Their main role is to coordinate food sources with populations, and marshal migrating fish. Sturmbeyer (1998) describes possible effects that reefs, and their environment (e.g. changes in water level exposing and covering reefs) may have on speciation, and hence representing this environment is important. However, experiments reported in this paper have not made use of this feature of TDLF.

- location=lionscove
- populationSize=40
- foodSources=lionsfoods.txt
- lineages=lineage.txt
- location=nkhalireef
- populationSize=40
- foodSources=nkhalifoods.txt
- lineages=lineage.txt

Note that both locations make use of the same 'lineage' file, meaning that both locations start with identical populations.

Fish can migrate from one reef to another, providing that probabilities are defined describing the probabilities of each fish making a transition for each time step.

- from lionscove to nkhalireef 0.03
- from nkhalireef to lionscove 0.03

### Experiment One: Speciation

In order to make claims about the progression of speciation in a simulated system, it is of course necessary to demonstrate that the changes occurring in simulated populations is actually speciation. This has been questioned after previous conference presentations of the system, and The system was run with parameters approximating the following ecology. Two clearly distinct food sources (0.0, 0.0) and (1.0, 1.0) were present, with the initial population of fish being placed exactly half-way between them (0.5, 0.5). Trophic, sexual, and preference values were represented by two dimensional co-ordinates, to allow easy graphing of the distributions of the final population. Fish were given ample opportunity to adapt (in terms of the variability and malleability of their phenotypes), and the simulation was run for 40,000 time steps (a time step is intended to represent one week meaning that the entire simulation lasts roughly 770 years). Phenotypes of the final population were extracted from the simulation trace, and visualised by conversion (normalisation of coordinates, and plotting of numbers identifying fish) to an *xfig* format file. (*xfig* (<http://www.xfig.org>) is a drawing and diagram creation tool for Unix computers which uses a simple text-based format for storing diagrams.) The three plots for the three phenotypes given in Figures 2, 3, and 4.

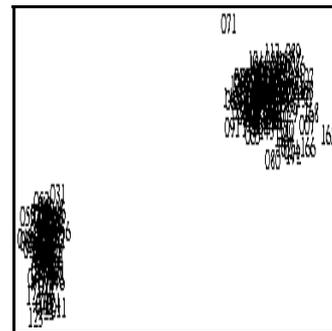


Figure 2: Sexual Phenotypes

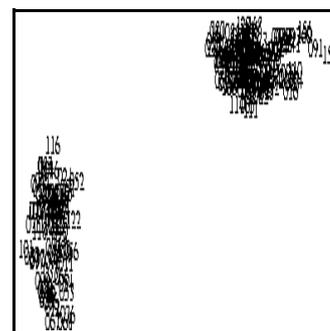


Figure 3: Preference Phenotypes

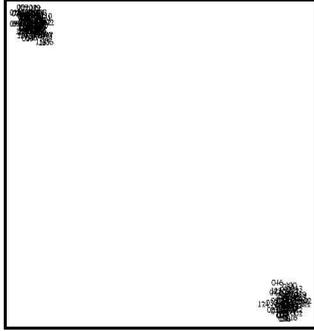


Figure 4: Trophic phenotypes

In order to confirm speciation, it is necessary to show that individual fish are not randomly assorted into the groups, but that there is a strong covariance (or high mutual information) between membership in the groups. Arbitrarily labelling groups for the three phenotypes (*sexual, preference, and trophic*): *As*, *Bs*, *Ap*, *Bp*, and *At*, *Bt*, and counting the membership of all eight combinations of groups given in Table 1. These results clearly show that group membership is consistent across all three phenotypes. In particular, fish belonging to the 'high' group for sexual preference (*As*) all belonged to the 'high' group for sexual phenotype (*Ap*), showing a preference for intra group mating. Similar results occur for *Bs*, and *Bp*, and these groups also segregate according to trophic adaptation. It is felt that these results confirm that speciation has occurred in the simulation.

These figures all show clear separation of each of the phenotypes into two groups (of fish) consistent with speciation. It is particularly interesting to note that clustering around food sources is much tighter than the (emergent covariance) for colour and sexual preference phenotypes.

	<i>ApAs</i>	<i>ApBs</i>	<i>BpAs</i>	<i>BpBs</i>
<i>At</i>	96	0	0	0
<i>Bt</i>	0	0	0	136

Table 1: Group membership

### Experiment Two: Competitive Exclusion

The competitive exclusion principle (see e.g. Begon *et al*, 1996) states that two species competing for the same resources cannot co-exist. Eventually one of the species will go extinct leaving the other species to dominate the environment. Cichlids in African lakes appear to contradict this principle as rocky reefs (and other environments) appear to support large numbers of species eating more or less identical diets. Parallel work (Clement, 2003b) investigated the role behaviour may play in dividing up resources among apparently competing species. The experiments reported here investigate the possibility that species that have lost the ability to adapt (due to inbreeding among small populations) may take much longer to exclude other species from an environment (or be excluded).

Food sources were placed at single dimensioned coordinates (0.25), and (0.75). These food sources were given *foodsource stdevs* of 0.3, sufficiently large to make the preferred strategy of fish a generalist adapted to (0.5). Two different lineages of fish were created (labelled *cichlid*, and *catfish*, the labels preventing any interbreeding or the collapse of the two species into one). *Cichlids* were given an initial trophic adaptation of (0.25), exactly specialised for the first food source, and *Catfish* were exactly specialised for the second (0.75) food source. In the first experiment, both groups of fish were given a *breeding stdev* (equal for all three phenotype) of 0.1, allowing rapid adaptation. After simulation, the time at which competitive exclusion was complete (one of the lineages completely disappeared with the death of the last fish from that lineage) was recorded. Each experiment was repeated 10 times, and the average time of competitive exclusion was calculated. Experiments for *breeding stdevs* of 0.08, 0.06, 0.04, 0.02, 0.01, and 0 were performed. The results of all experiments excluding *breeding stdev* of 0 are plotted in Figure 5.

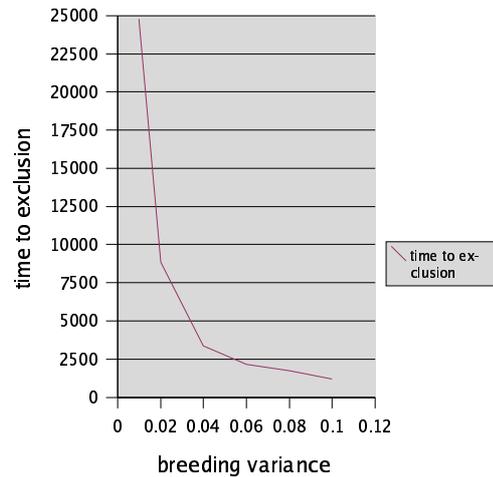


Figure 5: Competitive Exclusion Experiments

At a *breeding stdev* of 0 (i.e. the fish have no ability to adapt) competitive exclusion does not occur, even when the simulation was run to 400,000 steps (approximately 7700 years).

### CONCLUSIONS AND FUTURE WORK

The (virtual) fish simulated in TDLP clearly undergo a process that closely parallels speciation in the real world, producing species groups that are differentiated in terms of diet, and have developed a method (emergent covariance between colour and preference) of maintaining different species within a single location. The model only allows pre zygotic barriers to speciation (i.e. fish of different species do not mate and produce zygotes), rather than post-zygotic barriers (e.g. where matings may occur but the resulting offspring are infertile), based on colour male colour and female choice. This is acceptable, as this matches the situation for real African mouth brooding cichlids, where fish tend not to mate outside their species, but would produce viable

offspring if they did. Difficulties remain, particularly in visualising the process of speciation, rather than the result, of speciation. Every aspect of the simulation is far from a perfect match to the natural analogues, though this will always be necessary due to the near infinite complexity of real natural environments (especially lakes with the surface area of Switzerland). However, the simulation appears sufficiently realistic for investigation of conjectures concerning cichlid speciation and evolution.

The investigation of the effect on adaptability on competitive exclusion suggests that the ability to adapt increases the likelihood of competitive exclusion occurring. Adaptive organisms can evolve towards a global optima for harvesting a number of food sources, forcing out other species that might also gravitate towards the same optima. However, if the species (fish) are initially specialists in different food sources, and cannot adapt, then competitive exclusion will not occur.

While appearing interesting in itself, this result does not help explain the situation found in the African lakes. Cichlids are plainly quick at adapting to new food sources, as shown by the young Lake Victoria, where in ~14,000 years one or two generalist species appear to have evolved to fill nearly every available niche (Seehausen, 1996), though recent work (discussed in (Kocher, 2003)) suggests that Lake Kivu may be involved in the seeding of Lake Victoria, and therefore the evolutionary system being modelled may be much more complex than originally believed. One of the original conjectures on which this experiment was based was that the mixing of two fish with a very small ecological difference might lead to a stable system where the two species monopolise their own specialist food sources, removing the evolutionary pressure towards the single optimum, and exclusion of one of the species. Very similar African cichlids do co-exist in a single geographic location, and something prevents these species from excluding each other. But, it does not appear to be the factor conjectured in the exclusion experiment.

The concept may help explain other concepts. E.g. one frequently asked question about cichlids in the African Great Lakes is why cichlids dominate these lakes, to the expense of other fish families such as catfish and the carp families. Future experiments are planned to investigate whether differing relative speeds of adaptation will strongly affect the likelihood of one species winning the competitive exclusion race.

Using a real natural system as the basis for TDLP has many advantages over abstract systems. E.g. the number of parameters and model types that need to be considered can be reduced to only those that apply to the natural system in question. E.g. the use of male colour and female choice as method by which matings occur.

Furthermore, by choosing African cichlids as the natural system, well-known open problems in cichlid biology (such as the speed, and scale, of speciation, as well as the unexpected coexistence of species adapted to apparently identical niches). One of the outcomes of the exclusion experiment emphasises this point. The pattern found,

where the ability to adapt sped competitive exclusion, is a potentially valid concept in biology. But, it isn't needed (or helpful) in explaining the open question of why so many cichlid species co-exist, with no observable ecological differences. In fact, it adds to the mystery. Hence, the further consequences of this concept (apart from those mentioned above which may help explain the dominance of cichlids over other species in these lakes) do not need to be investigated.

One major problem with an agent-based simulation such as TDLP is the huge number of potential experiments that can be performed. Hence, it is necessary to carefully pick and choose what experiments are to be performed in order to prevent huge amounts of time being given over to small variations of the same experiment. The concentration on a real system (with real open problems) helps provide this focus. Hence, in this research we are attempting to rein in the temptation to attempt to model all and every possible influence on cichlid speciation. E.g. Kooi & Kooijman (2000) show how invading (predator) species can stabilise the populations of potential prey species, reducing the tendency towards competitive exclusion. Predation is a very important factor in cichlid ecology, but it has not yet been modelled. The reason for this is that in African lakes, the primary natural predators of cichlids are other cichlids. Since these cichlids evolved from common ancestors, modelling this reality will introduce considerable complexity into TDLP. The separation between sexual factors (colour) and trophic factors (food sources) disappears, a predator may adapt itself to better see a prey species, by increasing the number of photo receptors in the eye which detect the wavelength of the prey species breeding colouration.

At present, repeating Kooi & Kooijman's studies is a major future subproject. Kooi & Kooijman use mathematical modelling to support their results. Our intended repetition of their results will be as an agent-based implementation, most probably using RePast. This study is expected to answer the following questions. (i) Whether agent-based and mathematical modelling investigations of the same properties will give the same results. (ii) If the results agree, the degree of sensitivity of the agent-based model to changes in model parameters. (iii) Throw further light on the effects of predator-moderated competitive exclusion. And, (iv) allow the comparison of implementing agent-based models of speciation in both pure Java, and an agent-modelling platform.

As mentioned when discussing the choice of platform, it is interesting to discuss whether TDLP should be rewritten as a generic tool for agent-based simulations of speciation problems. One of the great difficulties of building a system such as TDLP is that a great deal of biology needs to be studied and incorporated to make the simulation results useful. In creating a general—purpose tool, the required amount of biological knowledge that would need to be incorporated (or at least understood) grows to seemingly unmanageable levels. TDLP contains many short-cuts which are not even applicable across all African cichlids (e.g. that young are cared for only by the

female, and that mate choice is female-driven). Furthermore, Genetic models assume sexual reproduction only, which is not universal over all organisms. As can be seen by studying the Galaxids of New Zealand, even among fish, the factors underlying speciation may be quite different. If another researcher were to attempt to use TDLP for research into the speciation of snails, or stick insects, even a single mismatch between the biological system and the facilities provided by TDLP (such as asexual reproduction), could make TDLP unusable. Because development and research time is not unlimited, a broadening of the intended audience of TDLP could result in a system which is theoretically capable of a wide range of tasks, but is ideal for none.

Within the biological research surveyed, the most advanced agent-based simulation platforms found are those in Ecology. Research has progressed to a point where review papers on individual-based (agent) simulation are common (e.g. Bercé, 2002). But, the most advanced of those found (CENOCOM, PK, MAS) tended to be publications describing the agent-based systems themselves, not papers describing new results found using these simulations. This may be a property of the age of these systems (typically very recent), or a symptom of a larger problem where the providers of agent-based simulation platforms being a different group of individuals from the potential consumers of these systems.

In terms of the future choice of generic agent-based platform versus low-level programming language, an important factor that has not yet been considered is repeatability of experiments. While plain Java is a convenient language for the implementation of TDLP, it may be very inconvenient for any non-programming researcher who might wish to repeat or further investigate any of the experiments performed with it. E.g. a large number of the biological details (such as Female-only mate preference) are not stated in the (text) configuration file, but buried within 14,000 lines of Java code. Due to the way TDLP developed, many classes have redundant methods included, which would make the mechanisms involved less than obvious for any observer. If a generic agent platform or tool were used, there would be a (at least partial) separation of biological model parameters, and genetic modelling tool code, which should improve accessibility of the model itself.

Compared to much ecological modelling, one of the major problems modelling speciation is the implicit nature of the factor being investigated. Particularly in African cichlids, species membership can often be a 'fuzzy' concept, with considerable argument as to what is a separate species, and what is a race. At present a considerable amount of work is being put into extracting, and visualising the process of speciation. This is based upon a clustering of Fish from a stable, final population, and the mapping of these clusters onto a set of potential ancestor species. (Clement, 2003b). From traces of these ancestor species (represented by fuzzy sets), phylogenetic trees can be extracted and plotted (Clement, 2003c). This work is in a fairly advanced state of development, and appears to be moderately well solved.

The bulk of the 'remaining' work for TDLP (though in reality, the 'remaining' work is infinite), is in improving the biological accuracy of the simulation. In addition to the inclusion of predation, other improvements planned are the inclusion of more, and more accurately modelled, food sources. One potential future project is the modelling of cichlid digestive tracts (and teeth) in considerable detail. At present TDLP only models a fish's suitability for a food source as a single (plastic) co-ordinate in a  $N$ -dimensional space. Hence a herbivore might be at a location of (0.3, 0.9), while an insectivore might be at (1.2, 0.1). Optimal locations for food sources (such as algae, and worms) is also given as points in this space, and therefore 'suitability' for a particular food sources decreases according to distance between these points. This is an abstract model approximating real-world properties of the fish and the environment. E.g. herbivore cichlids often have a longer intestine than carnivores, due to the greater difficulty of digesting plant material compared to animal flesh. The optimal length of an intestine therefore depends on the diet of the fish. It isn't a case of longer is better due to the developmental and metabolic costs of a long intestine. The same applies to teeth, with different teeth being better adapted for different methods of harvesting different prey. E.g. the genus *Labidochromis* have pincer-like teeth for picking small food items (either plant or animals) from tight crevices in rocks (Konings, 2001). These teeth would be less effective for scraping algae from a rock. It may be interesting and informative to build a Fish agent which has a realistically modelled digestive tract, and the potential for this tract to vary under genetic (or other) control. As well as the tract itself, fish behaviour would also need to be modelled. This would allow experiments showing the effects on anatomy on evolution and speciation. In theory, a covariance should still emerge between digestive tract form, foraging (or hunting) behaviour, and some sexual lock (colour) and key (female preference) resulting in speciation.

It is as yet unknown exactly how big TDLP (in terms of agents modelled) will have to grow to accurately model speciation. All three great lakes are huge, and their cichlid populations are similarly huge. However, many rock living cichlids live in very constrained habitats (e.g. rocky reefs, sometimes the size of a small room). As well as the Great Lakes, there are other smaller lakes (such as Lake Albert, and in particular, Lake Barombi-Mbo), where populations are much smaller, but speciation still occurs. At present this research is continuing under an assumption that patterns and properties of speciation found in smaller populations will scale up to larger populations, but this remains to be proven.

Future work, in general, is based around three basic concepts; improve the biological accuracy (not necessarily flexibility) of the simulation, improve the visualisation of the results, and continue investigating conjectures concerning cichlid biology. Where this research will benefit from the use of agent-based platforms or tools, these will be used. But, in the immediate future, progress is likely to concern mainly the existing Java implementation.

## ACKNOWLEDGEMENTS

This paper has benefited greatly from the comments and help of many individuals, including George Turner, Nicola Baxter, and the comments of a large number of anonymous referees both for this paper and for other papers.

## BIOGRAPHY

Ross Clement was born in New Zealand. He completed his B.Sc. degree in Cellular and Molecular Biology, and a M.Sc. degree in Computer Science at the University of Auckland (New Zealand). He completed the degree of Doctor of Engineering in Systems and Information Engineering at the Toyohashi University of Technology (Japan) in 1991. After working as a research assistant, studying Genetic Algorithms for Transport Scheduling at the University of Leeds (UK). Since 1993 he has been a Lecturer, then Senior Lecturer in the Harrow School of Computer Science, of the University of Westminster (UK). His research interests are the application of Artificial Intelligence and Computer Science techniques to models and simulation of Evolution. His email address is clemendr@wmin.ac.uk

## REFERENCES

- Barlow, G. W. (2000). *The Cichlid Fishes: Nature's Grand Experiment in Evolution*. Perseus, MA.
- Baxter, N. (2003). Intelligent Customer Relationship Management. OR45 Conference, Keele, UK. <http://www.orsoc.org.uk/conf/or45/Handbook.doc>
- Begon, M., J. L. Harper, and C. R. Townsend. (1996). *Ecology: Individuals, Populations and Communities*. 3rd ed. Blackwell Science, Oxford, UK.
- Berec, L. (2002). Techniques of spatially explicit individual-based models: construction, simulation, and mean-field analysis. *Ecological Modelling* 150: 55-81.
- Clement, R. P. (2003a). Plausible roles for Social Learning in the Speciation and Evolution of Cichlid Fish. *Journal of the AISB* (to appear).
- Clement, R.P. (2003b). Visualising Speciation in Cichlid Fish. *Proceedings of the 17th European Multisimulation Conference*, Nottingham.
- Clement, R.P. (2003c). Visualising Speciation in Artificial Cichlid Fish. Submitted to *Artificial Life*.
- Coleman, R. M. (ed.) (2001). Cichlid Research: State of the art. *Journal of Aquaculture and Aquatic Sciences* 9.
- Dieckmann U, Doebeli M (1999). On the Origin of Species by Sympatric Speciation. *Nature* 400: 354-357.
- van Doorn, G. & F. Weissing F. (2001). Ecological versus sexual selection models of sympatric speciation. *Selection* 2: 17-40
- Gilbert, N., & Bankes, S. (2002). Platforms and methods for agent-based modelling. *PNAS* 99 suppl. 3: 7197-7198.
- Ginot, V., Le Page, G., & Souissi, S. (2002). A multi-agents architecture to enhance end-user individual-based modelling. *Ecological Modelling* 157: 23-41.
- Goldschmidt, T. (1996). *Darwin's Dreampond: Drama in Lake Victoria*. MIT Press.
- Kocher, T. D. (2003). Fractious phylogenies. *Nature* 423: 489-491.
- Kondrashov, A. & Kondrashov, S. (1999). Interactions among quantitative traits in the course of sympatric speciation. *Nature* 400: 351-354.
- Konings, A. (2001). *Malawi cichlids in their natural habitat 3<sup>rd</sup> edition*. Cichlid Press.
- Kooi, B. & Kooijman (2000). Invading species can stabilize simple trophic systems. *Ecological Modelling* 133: 57-72.
- Mamedov, A. & Udalov S. (2002). A computer tool to develop individual-based models for simulation of population interactions. *Ecological modelling* 147: 23-68.
- Parrott, L. & Kok, R. (2002). A generic, individual-based approach to modelling higher trophic levels in simulation of terrestrial ecosystems. *Ecological Modelling* 154: 151-178.
- Paterson, H. (1993). *Evolution and the Recognition Concept of Species*. John Hopkins University Press.
- Rice, S. H. (1998) The evolution of canalization and the breaking of von Baer's laws: modeling the evolution of development with epistasis. *Evolution* 52: 647-657.
- Schliewen, U, Tautz, d, Pääbo, S. (1994). Sympatric speciation suggested by monophyly of crater lake cichlids. *Nature* 368: 629-632.
- Seehausen, O. (2002). Patterns in fish radiation are compatible with Pleistocene desiccation of Lake Victoria and 14,6000 year history for its cichlid species flock. *Proc R. Soc. Lond. B. Biol. Sci.* 269: 491-7.
- Seehausen, O. (1996). *Lake Victoria Rock Cichlids*. Verduijn Cichlids.
- Sturmbauer, C. (1998). Explosive speciation in cichlid fishes of the African Great Lakes: a dynamic model of adaptive radiation. *Journal of Fish Biology* 53 (Supplement A), 18-36.
- Sun Computers (2003). *Creating Interfaces*. <http://java.sun.com/docs/books/tutorial/java/interpack/interfaces.html>
- Turner, G. F. & Burrows, M. T. (1995). A model of sympatric speciation by sexual selection. *Proc. Royal Soc. Series B* 260 (1359), 287-292.
- Turelli, M., Barton, N., & Coyne, J. (2001). Theory and speciation. *Trends in Ecology and Evolution*.16: 330-343.
- Via, S. (2001). Sympatric speciation in animals: the ugly duckling grows up. *Trends in Ecology and Evolution* 16: 381-390.
- Waters, J.M., Craw, D., Youngson, J.H. & Wallis, G.P. (2001). Genes meet geology: fish phylogeographic pattern reflects ancient, rather than modern, drainage connections. *Evolution* 55:1844-1851.

# THE OsMoSys/DrawNET Xe! LANGUAGES SYSTEM: A NOVEL INFRASTRUCTURE FOR MULTI-FORMALISM OBJECT-ORIENTED MODELLING

Marco Gribaudo  
Dipartimento di Informatica

Università di Torino  
C.so Svizzera 185  
marcog@di.unito.it

Mauro Iacono  
Nicola Mazzocca  
Dipartimento di Ingegneria  
dell'Informazione  
Seconda Università di Napoli  
via Roma 29 Aversa (CE) Italy  
{mauro.iacono, nicola.mazzocca}  
@unina2.it

Valeria Vittorini  
Dipartimento di Informatica e  
Sistemistica  
Università di Napoli "Federico II"  
via Claudio 21 Napoli Italy  
vittorin@unina.it

## KEYWORDS

Modelling languages, multi-formalism, object orientation, XML.

## ABSTRACT

Complex systems present a big challenge to the modeller: different subsystems need different modelling techniques, because they have different purposes and different kind of specifications. A multi-formalism modelling methodology can be useful to unify the different aspects of a model. In this paper we propose a system of languages which constitutes the foundation of OsMoSys, a multi-formalism, multi-solution, object-oriented modelling framework. The presented languages system supports automatic generation of GUI for every modelling formalism through the integration of the DrawNET Xe! tool, as well as OO submodel reuse and inheritance, flexible multi-solver solution and result analysis through the OsMoSys solving architecture.

## INTRODUCTION

Modelling complex systems is a challenge for designers. Several modelling techniques have been developed and can be found in literature. These techniques are able to pursue performance prediction, verification or analysis of systems in both the design phase and validation phase. Complexity (in terms of number of subsystems, or behaviours, or heterogeneity of subsystems) exasperates the task of modellers, because different subsystems must meet different kinds of requirements, either functional (e.g. correctness) or non-functional (e.g. dependability), or both. Different submodels need to be developed in order to design and verify the whole system, with the consequent growth of the number of available system views. These views, each one independent from the others, should be kept synchronized during the design cycle of the system. In addition, different models, written in different formal languages, can not automatically share parameters or exchange results, because they have not been thought to interoperate. In order to obtain a comprehensive,

flexible, interoperable, composeable modelling technique, different languages/techniques must be integrated in an unique canvas.

Syntactical integration between formal languages is the first step on the path to a multi-formalism, multi-solver modelling technique. Every formal modelling technique is founded on a formal language. Syntactical integration relies on the definition of common rules behind different grammars which define different formal languages. A second step is semantic integration. Semantic integration defines the meaning of inter-model communication and parameter exchange, and, at the best of our knowledge, should be examined case by case, with some exceptions for formal languages that have common roots (e.g., various kind of Petri Nets evolutions share the basic elements of the languages). Semantic integration also introduces the problem of describing (and retrieving, as a side effect) the results of the evaluation of a submodel, because interoperation between models includes hierarchical composition or dependences in the model evaluation process. The inclusion (or the dependence) of a submodel in (by) another one, written in a different formal language, usually requires the evaluation of some parameters of one component and the transformation of these values into parameters for the other submodel. Automating this evaluation process requires the definition of additional formal languages suitable for the description of results and for the specification of a solution process.

This paper is part of a more complex research project whose purpose is the development of an integrated framework for multi-formalism, multi-solution modelling. Some research results have already been published on this topic and on the methodology behind the framework in literature (Gribaudo and Valente 2000b, Gribaudo and Sessi 2001, Vittorini et al. 2002, Franceschinis et al. 2002a, Franceschinis et al. 2002b, Gribaudo and Valente 2000a, Vittorini et al., Baravalle et al. 2003), as well as applications (Franceschinis et al. 2003, Gilmore and Gribaudo 2003). In this paper we show how the OsMoSys/DrawNET Xe! (OsMoSys in the following) languages family allows the designer to easily develop multi-formalism, object oriented models

together with a general specification of a complete multi-solver resolution process, from the graphical specification of models to the extraction of the results after their evaluation. In this work we will only describe the syntactic integration and resolution process specification: the semantic integration is out of the scope of this paper.

In the following section we will describe the OsMoSys methodology and the overall features of the OsMoSys languages family, their integration in the framework and in the DrawNET Xe! tool. The general organization in levels and layers is then presented, followed by an in-depth analysis of both the syntactic integration facilities and the modelling paradigm; then a brief description of the OsMoSys architecture is given, to show how the results definition and retrieval languages are used together with the resolution process definition languages.

## OsMoSys OVERVIEW

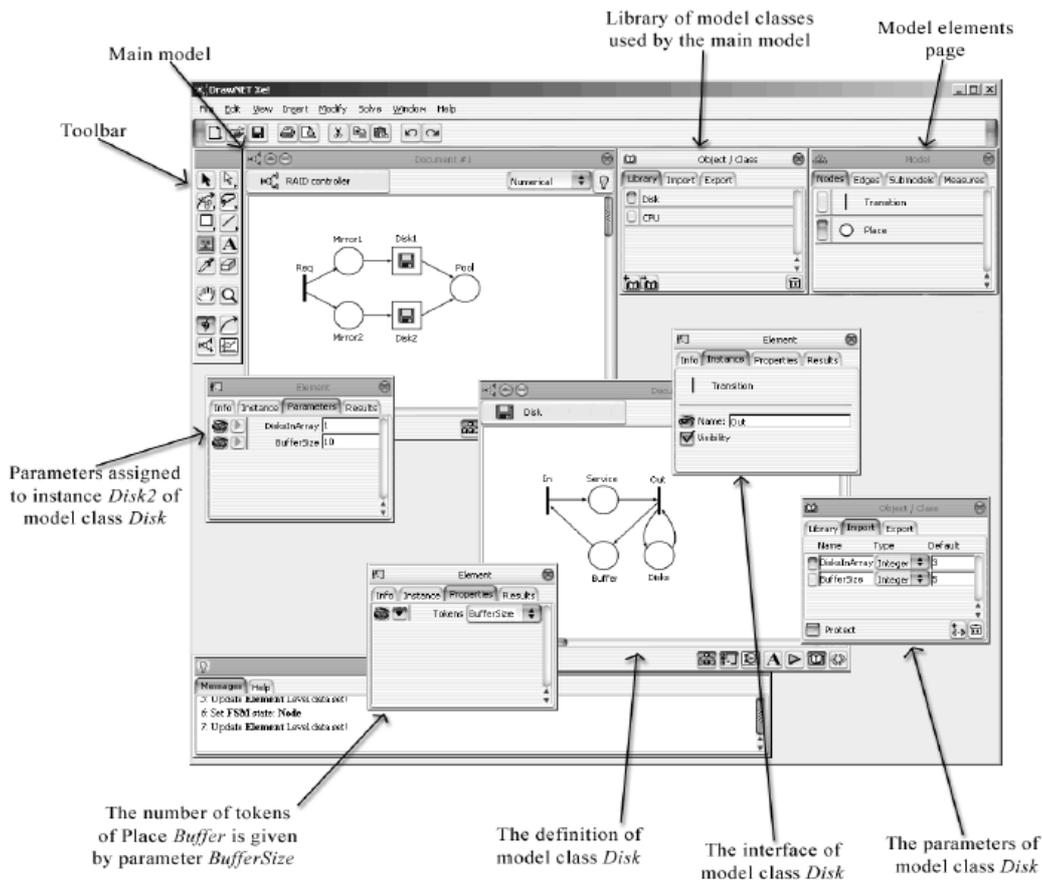
The OsMoSys methodology is the result of a collaboration between research groups at the *Univ. di Napoli* and the *Dip. di Informatica* of the *Univ. del Piemonte Orientale* on the themes of compositional model construction with sub-model reuse and multi-formalism modelling. The methodology is the evolving result of the work done by the two research groups to define formalisms integration and composition strategies and to develop an integration strategy for external solvers. In particular, ongoing research is aimed at applying workflow principles to cope with the problem of multi-solution when analysing/simulating multi-formalism models. A principal role in the work done is played by DrawNET, a multi-formalism GUI developed at the *Dip. di Informatica* of the *Univ. di Torino*. The third version of DrawNET (namely DrawNET Xe!) is currently under release. Its first release is at the origin of the first version of the languages family now used to support the OsMoSys methodology. The evolution of the two projects is now tightly connected because DrawNET Xe! has been adopted as the presentation level of the OsMoSys framework, which in turn is the implementation of the ideas of the OsMoSys methodology.

The principal guideline of the modelling paradigm in OsMoSys is flexibility: submodels in a model can belong to different formalisms and must speak to each other in order to allow the modeller to exploit his/her proficiency to cope at its best with any different problem in the design phase. In addition, the model can be obtained by composition of submodels and submodels can be reused and refined. In order to achieve this goal, the OsMoSys modelling languages family implements some concepts of Object Oriented Development (OOD), like classes, inheritance,

aggregation and instances. Submodels in OsMoSys are classes, which can be developed, stored in a class library and instantiated to build models. Non instantiated submodels are named Model Classes (MCs) in the OsMoSys terminology. MCs can be obtained by scratch, building them with a proper formal language, suitable for the evaluation process they are designed for. New MCs can be also derived from existing classes or can be assembled by aggregation (and partial instantiation) of existing classes. A model is finally a fully instantiated class. All these ideas are supported by DrawNET Xe!: in Figure 1, the tool screenshot shows a model together with a component submodel, interface and parameter definition facilities in order to build model classes, the model class library and the instantiation of a class through parameter specification. Going behind the modelling level, model classes must be written following a proper syntax. This syntax belongs to a proper formal language suitable for some kind of analysis, like Petri Nets (PN) and extensions, or Fault Trees (FT) and extensions et cetera. A syntax is defined by a grammar. In the OsMoSys terminology a grammar is named Model Metaclass (MM). The word *metaclass* defines the two main characteristics of it: as first, it's a class, so it can be obtained by derivation from another metaclass; besides, it is a meta-description, that is a description of a description (a MC is the description of an instance). This technique, known as *meta-modelling*, allows the definition of formal languages and of their syntactical interaction to develop the foundation of multi-formalism models.

These issues were already implemented in the OsMoSys framework (Vittorini et al.) before the introduction of the new languages family presented in this paper (and obviously in DrawNET++, the predecessor of DrawNET Xe!): but the new family exceeds the limitations of its predecessor and completely redefines the modelling and meta-modeling languages improving their clearness and their expression power, as well as DrawNET Xe! is a totally new tool rebuilt on the new languages paradigm.

Brand new features included in the new versions are the ability of describing and retrieving results in the solving architecture, and to show them through the GUI. Three different problems have been solved through the definition of proper languages for results specification: first, a language has been introduced in the family to describe the results themselves; another language has been introduced to define queries that can be sent to the solving subsystem in order to produce results; a third language has been defined to return the values of the results back to the GUI. In addition, proper languages have been introduced to support the specification and the enactment of solution strategies.



**Figure 1: the DrawNET Xe! interface with visual representation of model classes and the library**

## LAYERS AND LEVELS

A more in-depth view on the OsMoSys modelling technique requires a description of the conceptual structure of an OsMoSys model. An OsMoSys model can be viewed by two different points: from the modeller's one or from the structural one.

The modeller can consider an OsMoSys model as the result of the coexistence of three different layers: the conceptual graph layer, the visual representation layer and the solved model layer. The structural view of models is orthogonal to the layers view and consists of two levels: the description level and the metadescription level.

### OsMoSys layers

The conceptual graph layer is the core part of a model and constitutes its abstract representation. A synthetic view of the languages used in this layer is in Figure 2, which will be better examined in the following. This layer contains all the structural and quantitative information that form the model itself. OsMoSys has been designed and exploited for formal languages which models can be represented by a graph. This policy, the only available in the first version of the framework languages family, is adopted because of the presence, as a main model development tool, of the DrawNET Xe! user interface. The DrawNET Xe! user interface in fact implements syntactical validation on models and aids

the modeller in many tasks, as choosing between already allowed multi-formalism techniques as well as selecting the most appropriate multi-solver analysis. This choice is not a limitation for the expression power of the framework, because it is possible to design and implement graph-based representations for usually text-based formal languages (Gilmore and Gribaudo 2003).

A model is in the conceptual graph layer a graph of (nested) graphs: every submodel is represented by a graph itself and submodels can be nested by aggregation with no limitations in the depth level. The nodes of inner graphs are constituted by atomic components of a formal language (integrated into the framework). The OsMoSys implementation of a formal language will be named *formalism* in the following to better highlight the context. At each level, a graph is a submodel expressed in a single formalism. Each graph/submodel is included in another graph/submodel or connected to other graphs/submodels through a special kind of arc: these graphs/submodels can be written in the same or different formalisms. A closer view to this organization will be given in the next section.

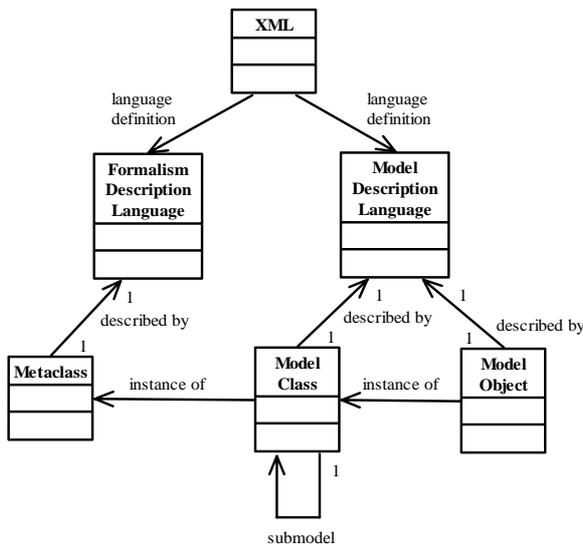
The visual representation layer is in charge of the visual presentation of a model: each element of each formalisms has a graphical version, able to give the user an iconic view of itself. The graphical representation is used by DrawNET Xe! in order to visualize a model and all its parts (Figure 1). The visual description of a model is kept separated from the model abstract description. This is done for two reasons: first, the two different

layers are independent representations of the same reality, both needed for different and disjoint purposes; second, while the visual representation layer is only exploited by DrawNET Xe!, conceptual graph layer is a bridge between DrawNET Xe! and the OsMoSys modelling framework and carries only the information required to store, to solve and to evaluate the model. A useful consequence of this organization is the independence of models from their representation (which can be modified whenever needed).

The solved model layer enriches the model with the results of its evaluation. A synthetic view of the languages used in this layer is in Figure 4, which is commented in the following. This layer contains an abstract description of the results of the pursued analysis. This description is flexible enough to incorporate structured information from any kind of (unknown when a model is written) solving process the model is sent to. This layer is responsible of:

- defining which kind of analysis can be pursued on a certain multi-formalism model;
- defining the structure of the analysis process for every allowed analysis, in terms of algorithms and solvers;
- defining how the allowed analysis can be communicated to the solving subsystem by the DrawNET Xe! user interface;
- defining the format of the results;
- describing the results.

These functionalities constitute a bouquet of services which is the warp of the solving subsystem of OsMoSys. The weft, as we will show later, is constituted by several software objects properly driven and coordinated.

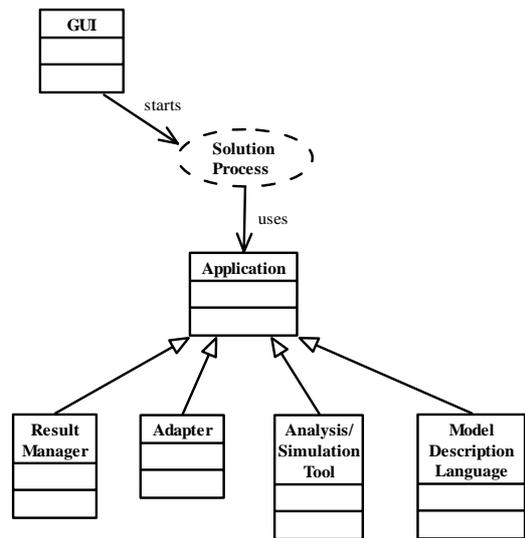


**Figure 2: the UML class diagram of the language system for conceptual graph layer**

### OsMoSys levels

The structural view of models is composed of two levels: the description level and the metadescription

level. These levels are present in each layer of the OsMoSys model layer organization. The description level is the part of the model the user can see, directly or by the DrawNET Xe! interface, including the conceptual graph, the visual representation, the result presentation and the model results queries. All these information are written following proper formalisms (modelling formalisms, visual formalisms, result structuring formalisms and model query languages, respectively). These informations are instances of the formalisms in which they are represented. One of the strengths of OsMoSys since the first version, is that those information are expressed in a language whose keyword are specified by an upper level specification. This composed description of the model is supported by the metadescription level, which defines the formalism grammars and the facilities to implement syntactical connections between submodels written in different formalisms. Figure 2 and Figure 4 respectively show the UML description of the set of languages used for models definition and results definition: the relationship between languages and other elements will be detailed in the following. For example, with reference to Figure 2 and Figure 4, the Formalism Description Language (FDL) is used to specify model metaclasses (formalism grammars), which allows writing model classes (submodels) by the Model Description Language (MDL); the Model Query Language (MQL) is used to write queries to be evaluated on models, with the support of results specifications, written by the Request Definition Language (RDL). This two-levels architecture gives OsMoSys the possibility of completely customizing the modelling formalisms and to express and retrieve results.



**Figure 3: the UML class diagram of the components participating in a solving process**

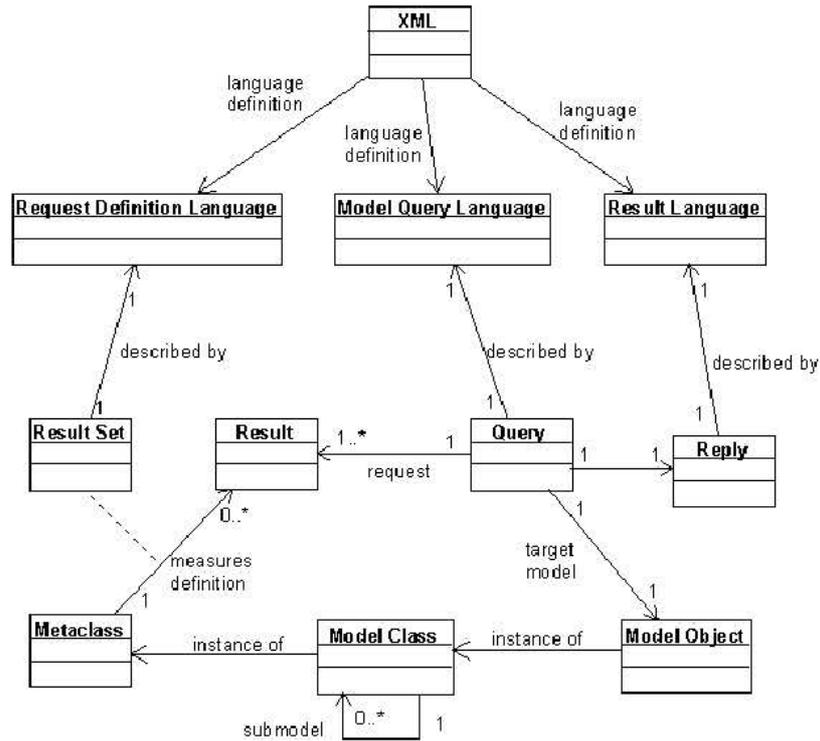


Figure 4: the UML class diagram of the language system for solved model layer

## MODELLING FORMAL LANGUAGES

According to the structural view, we will in brief describe the family of languages of OsMoSys by levels. All languages are based on XML because of its flexibility, portability and universality, as shown in Figure 2 and Figure 4. Three languages are used in the metadescription level of OsMoSys: they are the Formalism Description Language (FDL), the Result Description Language (RDL) and the Formalism Representation Language (FRL), each one used in a different layer of the modelling paradigm.

The FDL is the foundation of the conceptual graph layer. The FDL is used to specify formalism syntax for both the definition of models and the automatic customisation of the DrawNET Xe! interface. FDL is a brand new evolution of the previous version, already existing in OsMoSys/DrawNET++ with all the main features but completely redesigned and redefined. The FDL allows a formalism designer to describe all the elements of a formalism in terms of nodes (*elements* in OsMoSys), edges (*arcs* in OsMoSys) and *constraints*. The FDL grammar itself is defined by an XML dtd in order to keep coherence in the framework.

The central point of FDL is the element. An element is defined by an *elementType* definition and is characterized by properties (which are now typed in the new version). An element can contain other elements, references to other *elementType*s statements or *elementPointers* to other elements. References can be used to introduce type restrictions in the syntax of a model: *elementPointers*, as well as common software pointers, are handles which reference other elements in

the model and in addition are able to add new properties to the target. Pointers are a new extension to OsMoSys, as well as references. Constraints define limitation on other elements in the formalism: now they are independent entities while in OsMoSys they were owned by other elements. Constraints have been enhanced: while in the previous version they only were able to limit the kind and/or the number of *elementType*s participating in a relationship, they can also now express checks or include mathematical/boolean expressions to be evaluated on the constrained elements.

The FDL is object oriented. Every element can be inherited from an existing one and can extend it with new properties or new constraints. Elements can be defined concrete or abstract and visibility qualifiers can be used in models. Constraints can be defined as valid on the original *elementType* or on it and every derived *elementType*. FDL definitions (formalisms) can inherit one from another as well, by adding new elements or constraints and by inclusion of other formalisms. Inclusion is another new feature now available at FDL level. The following is an example of the FDL definition of Petri Nets (PN).

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fdl SYSTEM "fdl.dtd">
<fdl main="PN">
  <include src="base/GraphBased.fdl" />
  <include src="base/Instantiable.fdl" />
  <elementType name="PN" type="private">
    <parent ref="GraphBased" />
    <parent ref="Instantiable" />
    <elementType name="Place">
      <parent ref="Node" />
    </elementType>
  </elementType>
</fdl>
  
```

```

    <propertyType name="Tokens" type="integer" default="0" />
  </elementType>
  <elementType name="Transition">
    <parent ref="Node" />
    <propertyType name="Weigth" type="float" default="1.0" />
    <propertyType name="Priority" type="integer" default="1" />
  </elementType>
  <elementType name="Arc">
    <parent ref="Edge" />
    <propertyType name="Weight" type="integer" default="1" />
    <constraint>
      <check op="isOfKind" ref="from" kind="Transition" />
      <check op="isOfKind" ref="to" kind="Place" />
    </constraint>
    <constraint>
      <check op="isOfKind" ref="from" kind="Place" />
      <check op="isOfKind" ref="to" kind="Transition" />
    </constraint>
  </elementType>
  <elementType name="InhibitorArc">
    <parent ref="Edge" />
    <propertyType name="Weight" type="integer" default="1" />
    <constraint>
      <check op="isOfKind" ref="from" kind="Place" />
      <check op="isOfKind" ref="to" kind="Transition" />
    </constraint>
  </elementType>
  <elementTypeRef ref="PN" />
</elementType>
</fdl>

```

As seen, FDL definitions are classes themselves. Two special abstract FDL definitions actually form the kernel of our formalism library: `GraphBased.fdl` and `Instantiable.fdl`. The first one defines (exploiting the FDL language itself) an abstract graph formalism to be extended by concrete formalisms. The second one defines a syntax to implement (exploiting the FDL language itself) instantiability of submodels within other submodels: it defines an *interface* and a *parameter* elements, as well as an *use* construct, which allows derived formalisms to export and import elements when instantiating models and to include submodels.

The RDL is used to define feasible results from the analysis of models of a certain formalism. RDL is a new feature of OsMoSys. RDL is similar to FDL: it allows to define elements, which have resultTypes that describe the quantities a result query can refer to. ResultTypes are typed and can be structured or contain other resultTypes. An RDL formalism should be designed with respect to the solution engine able to obtain the desired results from the evaluation of a model: special hints can be added to resultType definitions in order to modify the solver evaluation process evolution (e.g., to obtain optimisations). RDL is object oriented too: inheritance and aggregation are supported as well as in FDL. The following is a simple example of the RDL definition of Petri Nets (PN).

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE rdl SYSTEM "rdl.dtd">
<rdl main="PN">
  <elementType name="PN">
    <elementType name="Place">
      <resultType name="MeanTokens"/>
    </elementType>
    <elementType name="Transition">
      <resultType name="Throughput" defaultCompute="yes"/>
    </elementType>
  </elementType>
</rdl>

```

```

  </elementType>
  <elementType name="Arc">
    <resultType name="TokenFlow"/>
  </elementType>
  <aggregateType name="MeanValue">
    <propertyType name="Expression" type="string" default="" />
    <resultType name="Mean"/>
  </aggregateType>
</elementType>
</rdl>

```

The FRL is used to describe the visual representation of the elements of the formalism. FRL is a new feature of OsMoSys. It associates a set of graphical representations to the element of a formalism. The language used to describe the appearance of each element is a subset of SVG (Scalable Vector Graphic, a w3 standard for vector graphic representation (Eisenberg 2002)). Each element may have more than one graphical representation: the used one is chosen depending on the property of the associated element. In this way, for example, a PN place may have different representations depending on the property that holds the number of tokens: each representation adds to the circle commonly used to visualize a place, a number of small filled circles corresponding to the marking.

The fact of having the visualization appearance of a formalism in separate layer allows the possibility to have several different representation for a single formalism: a FRL document can be thought as a style sheet for the corresponding Model Representation Language (MRL) document. This can be useful for example for formalisms for which more than a single standard representation exists. For example for Fluid Stochastic Petri Nets two different representation exist: one in which the arcs used to transfer fluid are represented as bold arrows (Ciardo et al. 1999), and another where they are represented as double arrows (Horton et al. 1998).

## MODELLING MODELS AND RESULTS

As in the metadescription level, three languages are used in the description level of OsMoSys: they are the Model Description Language (MDL), the Model Query Language (MQL) and the Model Representation Language (MRL), each one used in a different layer of the modelling paradigm.

The MDL is used to represent the conceptual graph layer. MDL models are automatically generated by DrawNET Xe! from the graphical inputs of the modeller (Figure 1). MDL is used for the definition of both model classes and model instances. A MDL document refers to a FDL description that states the admitted model elements and eventually some additional characteristics, like the instantiability feature. In the following we will assume the general case of instantiable graph-based models. Although object oriented capability (inheritance, aggregation, information hiding) were already present in the previous version, a completely new approach has been adopted in OsMoSys for three reasons: to support class libraries, to enhance the instantiation mechanism and to improve the information

hiding features. An MDL model defines now the structure of a model class, e.g. the various elements present for each element type defined in the corresponding FDL, their connection and the included (sub)model objects, if any. In the last case, references to their description are included pointing to the same document or to external documents, e.g. from a MDL library. The instantiation of included model classes is pursued through the instantiation of their interface and parameters or a further export of them. A model class can also export some parameters and some interfaces: a completely instantiated model class is a model instance (the main model). Included classes are not necessarily based on the same FDL, thus implementing multi-formalism. The definition of inter-formalism connections should be defined in the container model class FDL.

The following is a simple example of the MDL definition of a PN model class including two nested classes, one internal and one external, and exports a parameter and an interface. Notice that PN.fdl should include Instantiable.fdl in order to include and instantiate model classes.

```
<mdl fdl="PN.fdl" main="Net1">
  <PN name="PNClass">
    <interface add="#T0"/>
      <parameter name="K" default="1">
        <assign obj="#P0" property="token"/>
      </parameter>
      <Transition name="T0" rate="1.0"/>
      <Place name="P0" token="2"/>
    </PN>
    <PN name="Net1">
      <Place name="P0" token="1"/>
      <use class="#PNClass" name="Inst1">
        <parameter name="K" value="7"/>
      </use>
      <arc name="arc0" from="P0" to="Inst1.T0"/>
      <!-- Includes an extern object -->
      <use class="library.mdl#MachineClass" name="Inst2">
        <parameter name="Speed" value="1.0">
      </use>
      <!-- Defers K and exports Inst1.T0 from the nested object -->
      <interface add="#Inst1.T0"/>
      <parameter name="K" default="1">
        <assign obj="#Inst1.K" property="value"/>
      </parameter>
    </PN>
  </mdl>
```

The MQL is used to define queries to be solved by the solution subsystem. An MQL document refers to a RDL document which defines the feasible results the MQL can ask for. Aggregate measures can be included, as well as derived measures through calculations. The following is a simple example of a MQL query for a PN model.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fdl SYSTEM "mql.dtd">
<mql rdlref="PN-GreatSPN.rdl">
  <element name="APetriNet" type="PN">
    <result name="HasPSemiFlow"/>
    <element name="P0" type="Place">
      <result name="MeanTokenNum"/>
    </element>
```

```
<element name="P1" type="Place">
  <result name="MeanTokenNum"/>
  <result name="MeanTokenDistrib"/>
</element>
<element name="T0" type="Transition">
  <result name="Trthroughput"/>
</element>
<aggregate name="M1" type="Mean">
  <property name="Expression" val="P0+P1" />
  <result name="Value"/>
</aggregate>
<aggregate name="M2" type="Mean">
  <property name="Expression" val="T0 * P0 / (P0+P1)" />
  <result name="Value"/>
</aggregate>
</element>
</mql>
```

The MRL is used to describe the visual representation of a specific model expressed in a given formalism. In order to make the tool more extensible and allow a deeper integration in a graphic environment, MRL files are *SVG compatible*. MRL are structured in a way that any SVG viewer (that are currently integrated in most web browser and available for most platforms) can show the model they describe. In this way the models drawn in DrawNET can be immediately imported and printed by most existing graphical editing packages. The MRL files do not only hold the information regarding the representation of the models, but also all the information required by the GUI to define the visual appearance of a model, such as layers, texts and the other features implemented in the interface.

MRL files are connected to their corresponding MDL file using the ID attribute of the various SVG primitives that composes the MRL description. Using a special syntax, each ID of a graphic primitive that represents a model element, can be used as a pointer to connect it with the corresponding element in the MDL file.

## LANGUAGES TO SUPPORT THE SOLVING ARCHITECTURE

The semantic level of OsMoSys also requires the support of proper languages in order to let the modeller take advantage of the multi-solving architecture. The semantic level is in charge of giving a meaning to the syntactic interactions between heterogeneous communicating submodels. The semantic level is implemented in the OsMoSys solving architecture, which is a system of software objects formed by many components. Figure 5 shows a general description of the software architecture of the solving subsystem. We will not investigate in depth this software architecture, but we will give a view on it in order to introduce the remaining languages of the OsMoSys family. Figure 5 shows a three-level architecture: a client interface, representing in this paper the DrawNET Xe! interface; a workflow management level, which implements the logic of the solution process; and a components level, in which several components can be found. The workflow management is under the control of a workflow engine: it uses a process repository to store the known solving processes, written in another XML-based language

which description is out of the aims of this paper. The available components (see Figure 3 for an UML description), coordinated by the workflow engine, are:

- adapters, used to interface OsMoSys with legacy solvers;
- external solvers, which are used to obtain partial evaluations on the model;
- pre-postprocessors, which implement the semantic layer of OsMoSys;
- the request manager, which is responsible of processing MQL documents and producing answers.

The client level supplies the substanding architecture with a MDL document and a MQL document and waits for results to show them back to the user. In order to compute a solution, more information should be supplied by the user: which is the preferred solution process, which documents are involved in the description of the model and which results must be computed for that model. A proper database structure has been developed to manage the set of XML-based languages.

The set of formalisms that the DrawNET GUI may have to cope with can grow quite rapidly. Since every formalism is described by several file (at least an FDL, FRL and RDL, but it may also have several different results definitions and graphical representations), it is clear that the number of files that must be organized can be quite large. The database structure helps with the formalism files organization. This database is also stored in a XML file and can be accessed by the GUI and by the various solution components to locate the required resources.

The GUI uses this database to present to the user a list of formalism from which he may choose which modelling language / solution component he may use to describe his models. The database is also accessed by the solution components to find the formalism definition files (which may be required to compute the solution of the model) and to find parameters that may be required to guide the solution process.

The database is composed by four tables: *formalisms*, *results*, *styles* and *solution processes*.

The *formalism* table holds the list of the available formalisms. For each formalisms it holds a textual description of its name, and the URL of the corresponding FDL file. The *results* table associates to each formalism a set of solvers. This is required since, as we have pointed out before, each formalism may have more than solver that can handle it, and each solver may be able to compute different results. Each row of this table contains the URL of the corresponding RDL file and a textual description of the solver. The *styles* table is parallel to the results table, in the sense that it holds the definitions of the visualization style that can be associated to a formalism. Each row of this table holds the textual description of the style and the URL of the corresponding FRL file. We must point out that in most of the cases, both the results and the styles table will have just one entry for each formalism. The

*solution processes* table holds several entries for each formalism-result pair. Each solver can be able to perform different solutions, using different techniques. For example a Petri Net steady state solution can be computed either using analytical techniques or by simulation. Each row of the solution process table describes a specific solution. It contains a textual description of the solution, an URL for the process that carries out the solution, and a set of parameters. Those parameters are passed to the solver process, together with the MDL and MQL files when that particular solution technique is requested. The GUI uses this table to present the user the list of the possible solutions that can be invoked for a specific formalism/solver pair. Figure 6 shows the ER diagram for the proposed database.

Once the desired solution process has been selected and sent to the workflow management level, documents are routed from one tool to another by the engine, as well as the components of the solving subsystem are activated in turn as needed. After the solution process computes the results, they have to be conveniently represented and sent back to the DrawNET Xe! interface. For this purpose another language has been designed, namely ReSult Language (RSL). The RSL is designed to be a bridge between the solving subsystem, the result definition mechanism and the result presentation logic of DrawNET Xe!: the organization of this document reflects the presentation needs. Results are described by model element name as well as by the measure type and by the graphical organization: they can be organized in frames, in order to show temporal series, as well as in single values to be showed near the corresponding element or in series, to obtain plottable curves.

The following is a simple example of a RSL document for a PN model.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fdl SYSTEM "mql.dtd">
<rsl rdlref="PN-GreatSPN.rdl">
  <frame label="Control" base="true">
    <element name="ReteDiPetriDiProva" type="PN">
      <result name="HasPSemiFlow" value="yes"/>
      <element name="P0" type="Place">
        <result name="MeanTokenNum">
          <value val="3.7777"/>
        </result>
      </element>
      <element name="P1" type="Place">
        <result name="MeanTokenNum" format="single">
          <value val="3.7777" />
        </result>
        <result name="MeanTokenDistrib" format="table">
          <value index="1.0" val="3.7777" />
          <value index="1.1" val="4.7777" />
          <value index="1.2" val="5.7777" />
          <value index="1.3" val="4.7777" />
        </result>
      </element>
      <element name="T0" type="Transition">
        <result name="Trthroughput">
          <value val="237.34"/>
        </result>
      </element>
      <aggregate name="M1" type="Mean">
        <result name="Value">
```

```

<value val="46"/>
</result>
</aggregate>
<aggregate name="M2" type="Mean">
  <result name="Value">
    <value val="0.000004534"/>
  </result>
</aggregate>
</element>
</frame>

```

```
</mri>
```

The RSL is thus the final brick of the languages system and close the circular data path from/to DrawNET Xe!. As a matter of fact, it can be considered another description level built on the metadescription level of the RDL, as well as MQL (see Figure 4).

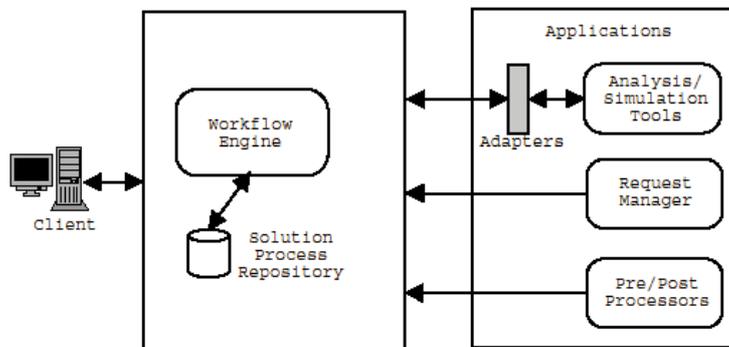


Figure 5: the structure of the OsMoSys solving architecture

## CONCLUSIONS

In this paper we analysed the language infrastructure used in the OsMoSys framework to specify the various different aspects of a model. We showed how a multi-formalism paradigm can be supported by exploiting object oriented developing techniques through a family of description languages enforced by a metadescription level. The metamodelling approach allows the framework to be flexible and easily extensible and gives it a canvas to glue together the conceptual modelling view, the graphical representation view and the solution process view. Moreover, the layer/level structure of the languages family gives a comprehensive organization to the framework. The intrinsic graph structure for the models is showed to be not a limitation for the expression power of the OsMoSys framework: on the other hand, it allows an easy hierarchical representation of models.

The modelling framework is supported as well by a GUI interface as by an extensible and open solution subsystem: the latter itself is parameterised and driven by proper languages in the framework and the solution process itself is described by a special purpose language.

XML is behind all the language systems: the modelling language implements advanced syntax constructs, like pointers or class parameterisation, thanks to the flexibility of XML. The language system is an infrastructure which connects the various levels and elements of the whole framework and constitutes its foundations.

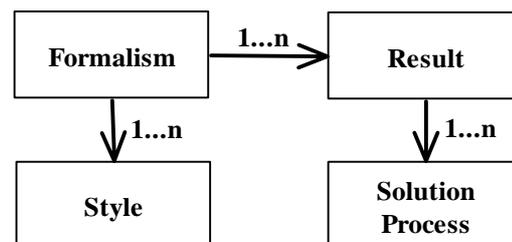


Figure 6: the ER diagram of the formalism database

## ACKNOWLEDGEMENTS

This paper is partially supported by Centro Regionale di Competenza sulle Tecnologie dell'Informazione e della Comunicazione of Regione Campania.

## REFERENCES

- Baravalle, A., Franceschinis G., Gribaudo, M., Lanfranchi, V., Iacono, M., Mazzocca, N., Vittorini, V. 2003. "DrawNET Xe: GUI and Formalism Definition Language". To be published in *Proceedings of Performance TOOLS 2003 Conference, part of the 2003 Illinois International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communication Systems* (Urbana, Illinois, USA Sep. 2-5).
- Ciarlo, G., Nicol, D. M., Trivedi, K. S. 1999. "Discrete-event Simulation of Fluid Stochastic Petri Nets" *IEEE Transactions on Software Engineering* 25, Vol.2, 207-217.
- Eisenberg, J. D. 2002. *SVG essentials*, O'Reilly.
- Franceschinis, G., Gribaudo, M., Iacono, M., Mazzocca, N., Vittorini, V. 2002. "DrawNET++: Model Objects to Support Performance Analysis and Simulation of Complex Systems". In *Lecture Notes in Computer Science 2324, Proceedings of the 12th International Conference on Modelling Tools and Techniques for Computer and*

*Communication System Performance Evaluation (TOOLS'02)*. Springer-Verlag, London, 233–238.

- Franceschinis, G., Gribaudo, M., Iacono, M., Mazzocca, N., Vittorini, V. 2002. "Towards an Object Based Multi-Formalism Multi-Solution Modeling Approach". In *Proceedings of the Second Workshop on Modelling of Objects, Components and Agents (MOCA'02)* (Aarhus, DK, Aug. 26-27). 47–65.
- Franceschinis, G., Marrone, S., Mazzocca, N., Vittorini, V. 2003. "SWN Client-Server Composition Operators in the OsMoSys framework". In *Proceedings of 10th Int. Workshop on Petri Nets and Performance Models (PNPM'03)* (IL, USA, Sep.). IEEE Soc. Press.
- Gilmore, S. and Gribaudo, M. 2003. "Graphical Modelling of Process Algebras with DrawNET". In *Tools presentation at the Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems* (IL, USA, Sep.).
- Gribaudo, M., Valente, A. 2000. "Framework for Graph-based Formalisms". In *Proceedings of the 1st International Conference on Software Engineering Applied to Networking and Parallel Distributed Computing (SNPD'00)* (Reims, France, May). 233–236.
- Gribaudo, M., Valente, A. 2000. "Two levels interchange format in XML for Petri Nets and other graph-based formalisms". In *Proceedings of the 21st International Conference on Application and Theory of Petri Nets* (June), 22-29.
- Gribaudo, M., Sessi, D. 2001. "A Multiparadigm Simulation Framework". In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'01)* (June), 1647-1653.
- Gribaudo, M. 2001. "FSPNEdit: a Fluid Stochastic Petri Net Modeling and Analysis Tool". In *Proceedings of Tools of 2001 International Conference on Measuring, Modeling and Evaluation of Computer and Communication Systems* (Aachen, Sep.). 24-28.
- Horton, G., Kulkarni, V. G., Nicol, D. M., Trivedi, K. S. 1998. "Fluid stochastic Petri Nets: Theory, Application, and Solution Techniques" *European Journal of Operations Research* 105, No.1 (Feb), 184-201.
- Vittorini, V., Franceschinis, G., Gribaudo, M., Iacono, M., Bertocello, C. 2002. "DrawNET++: a Flexible Framework for Building Dependability Models". In *Proceedings of Tools presentations, Proc. of the International Conference on Dependable Systems and Networks (DSN'02)* (June).
- Vittorini, V., Franceschinis, G., Iacono, M., Mazzocca, N.. "OsMoSys: a new approach to multi-formalism modeling

of systems". To be published in *SoSyM, Journal on Software and System Modeling*, Springer.

## AUTHOR BIOGRAPHIES

**MARCO GRIBAUDO** has currently a researcher position in the University of Torino, where he teaches Computer Graphics at the course of Arts, Music and Cinema. He obtained his PhD from the same university in 2002, with a thesis on Hybrid System for performance evaluation. His current research interests are: performance evaluation using hybrid formalisms, tool development and different formalism integration, communication networks and multimedia systems, computer graphic and virtual reality applied to learning.

**MAURO IACONO** is adjunct professor and research assistant at the Second University of Naples (SUN). He obtained his Laurea degree in Ingegneria Informatica in 1999 from the University of Naples "Federico II". After that, he joined the SUN where he received his PhD in Electronic Engineering in 2002. His current research interests are: performance evaluation by multi-formalism modelling techniques, complex systems and reactive systems engineering. He is a consultant of the Italian Ministry for Innovation and Technology (MIT) within the e-government national plan.

**NICOLA MAZZOCCA** is full professor of Calcolatori Elettronici at the Second University of Naples (SUN). He graduated in electronic engineering from the University of Naples, Italy, in 1987, and received his PhD from the same university. His scientific activity involves methodologies and tools for performance evaluation of computing systems, computer networks, communication protocols, general and special purpose parallel architectures and applications. Since 1998 he participated in various research projects as coordinator.

**VALERIA VITTORINI** is assistant professor at the Department of Computer Science and Systems of the University of Naples "Federico II", Italy. She graduated in Mathematics at the University of Naples in 1990 where she received her PhD degree in Computer Science in 1996. Her research interests include distributed systems, systems modelling and formal methods in system specification and design.

# MATCHMAKING IN THE ABELS SYSTEM FOR LINKING DISTRIBUTED SIMULATIONS

Joshua O. Peteet, John P. Murphy, and Linda F. Wilson  
Thayer School of Engineering  
Dartmouth College  
Hanover, NH 03755-8000 USA  
Email: [Firstname.Lastname@dartmouth.edu](mailto:Firstname.Lastname@dartmouth.edu)

## KEYWORDS

Distributed simulation, brokering systems, matchmaking, dynamic information exchange, software agents, simulation tools.

## ABSTRACT

Large-scale simulations often need dynamic access to heterogeneous data resources such as sensors, databases, or other simulations. The Agent-Based Environment for Linking Simulations (ABELS) is designed to facilitate the dynamic formation of a “cloud” of independent simulations and other data resources for the exchange of information. Participants in the data and simulation cloud join and exit the cloud as needed and have no prior knowledge of the other cloud participants. The formation of the cloud is achieved using a distributed brokering system that matches data consumers in the cloud with appropriate data producers, based on registration information submitted by the various participants in the cloud. This paper describes in detail the process used to match and rank prospective data producers for a given data consumer.

## 1. INTRODUCTION

Suppose you have a large simulation or similar application that needs access to data from multiple networked resources. One approach is to hardwire those connections so that your application knows exactly where to find the data, what formats are used, etc. One downside of this, of course, is that you must change your code if any of the resources change in location or format. Another approach is to write your application to a particular standard (e.g., HLA, Dahmann et al. 1998), so that you can communicate with other resources that conform to the standard. However, you may use only those resources that conform to the standard, and in many cases, you must still know information about the resources that are used by your application. Yet another approach is provided by the web services architecture, which uses the SOAP, WSDL, and UDDI protocols to provide interoperability between independent services (Curbera et al. 2002). The web services architecture, however, requires that you know in advance which specific resources will be used.

Networked resources come and go, so problems can occur if the resources you have specified become unavailable when you are running your simulation. It would be helpful if your simulation could use the best resources available and determine what those resources are at runtime. Furthermore, it would be even better if you could take advantage of all resources available, rather than only those written for a particular standard.

The Agent-Based Environment for Linking Simulations (ABELS) system is designed to facilitate the rapid formation of a distributed “cloud” of autonomous data resources (Mills-Tettey et al. 2002; Mills-Tettey and Wilson 2003a; Mills-Tettey and Wilson 2003b; Murphy et al. 2003; Wilson et al. 2001; Wilson et al. 2002; Wilson et al. 2003). Individual cloud participants join and exit the data and simulation cloud as needed and have no prior knowledge of the other cloud participants. A distributed brokering system is used to match data producers to consumers and initiate communication between cloud participants, but it does not control the independently-designed participants in any way. Each participant in an ABELS cloud is responsible for determining what resources it makes available to the cloud and for describing those resources accurately.

Cloud participants may be data producers, data consumers, or both. Each data producer is said to provide a service, while a data consumer is said to make requests or queries for information. A service definition includes the name, location, and description of the service along with detailed information about the functions it provides. A query is defined as the ideal function desired by the consumer. Note that a particular query may match functions from multiple producers, so the ABELS system must perform a detailed matching and ranking of the candidate services and functions.

This runtime matching of data producers and consumers is a key feature of ABELS. While a simulation or other application must be able to specify what resources it needs and what services it provides to the cloud, it does not need to know any specifics about the other participants (e.g., language, units, file formats, etc.). This feature allows the transparent replacement of one service provider with another that provides similar

functionality, without having the services be written to conform to a particular standard.

This paper describes in detail the matching and ranking system used to match consumers with producers. Section 2 provides an overview of the ABELS system and mentions related work. Section 3 discusses the goals of the matching and ranking system, and Section 4 describes the process of defining services and queries. Section 5 discusses the ranking process by which the individual service functions are evaluated in terms of fitness to a given query. Section 6 examines the query resolution process, while Section 7 presents conclusions and areas of future work.

## 2. BACKGROUND

### 2.1. The ABELS System

The Agent-Based Environment for Linking Simulations (ABELS) is a software framework that allows independent simulations and other data resources to exchange information with no prior knowledge of each other. An ABELS cloud is a federation of communicating participants, where each participant produces and/or consumes some type of data. As shown in Figure 1, the ABELS system architecture consists of three basic types of components: user entities, generic local agents (GLAs), and a distributed brokering system. An optional user interface is provided to permit human interaction with the system via the GLAs.

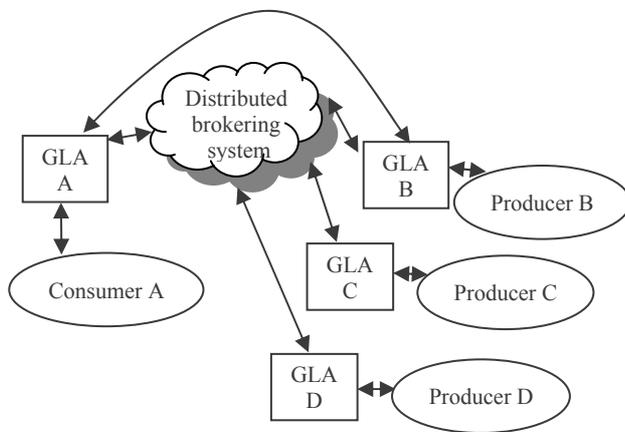


Figure 1: Basic Framework Connecting Elements in the Cloud

A user entity is any producer or consumer of data, and simulations often serve as both producers and consumers. A producer is considered to be a service that provides one or more service functions. A consumer makes requests or queries for the information it desires from the cloud. Figure 1 shows four user entities: Consumer A, Producer B, Producer C, and Producer D.

The ABELS system is designed for loosely-coupled interactions between cloud participants. That is, user entities are independent and are not written to a

particular standard, and consumers are not statically linked to particular producers. Furthermore, there are no tight interdependencies between cloud participants; ABELS is not appropriate for participants that are tightly-coupled to one another.

Each user entity communicates with the cloud via its general local agent (GLA). The user entity uses its GLA to join or exit the cloud, register its services and service functions, and make queries for data. A data producer is said to interface to the cloud via a producer GLA, while a data consumer communicates via a consumer GLA. Although a particular GLA may serve both producers and consumers for a single organization, it is still useful to discuss the GLA by separating its capabilities into producer actions and consumer actions. For example, a consumer GLA is responsible for handling any data format, unit, and file conversions that are necessary between the consumer and the producer that is serving it. A producer GLA is responsible for passing input data to a desired service, executing the desired service function, and returning the output data to the corresponding consumer GLA. In Figure 1, GLA A is a consumer GLA, while GLAs B, C, and D are producer GLAs. The GLAs are implemented in Java.

The distributed brokering system is responsible for managing all of the cloud participants and matching consumers with suitable data producers. Once the brokering system establishes links between two GLAs, the GLAs communicate directly without going again through the broker. In Figure 1, there is a link between GLA A and GLA B, indicating that a match has been made between a query of Consumer A and a service function of Producer B. Logically, the brokering system consists of the broker, the matching and ranking system, and the keyword and conversion databases. The broker is implemented using Java and Sun Microsystems' Jini technology (Kumaran 2002), while the other components are implemented using Java. (Additional information on Jini can be found at <http://www.sun.com/jini>).

The broker manages all of the resources in the cloud. It uses a system of leases to determine which participants are in the cloud and detect when someone has left the cloud unexpectedly. It also stores descriptions and remote references or proxies for all of the resources in the system. When a producer GLA registers a service with the brokering system, the GLA sends a proxy object that is used later by a corresponding consumer GLA to communicate with the producer GLA via Java Remote Method Invocation (RMI). When new services arrive or existing services become unavailable, the broker also notifies potential consumer GLAs of these changes, so that the best service function available can be used to resolve a consumer's query.

For better accuracy and efficiency, there are two levels of matching in the ABELS system. The broker stores the service information and performs the first-level matching according to high-level categories or groups such as “medical simulations” and “ocean simulations”. That is, the broker determines which services belong to the groups of interest. As indicated in Figure 1, there may be multiple producers eligible for matching with the query of a consumer. The matching and ranking system, which is the focus of this paper, performs the second-level matching by comparing the query with all of the service functions belonging to the services returned by the broker’s first-level matching. Note that each service may have multiple functions, only some of which may be relevant to the particular query. Thus, the matching and ranking system examines the query and all of the corresponding service functions in detail, and ranks the matching service functions according to their descriptions, data types, keywords, and measurement units. For efficiency and potential user interaction, the matching and ranking system is local to each GLA and is actually implemented as part of the GLA. The matching and ranking process will be described thoroughly in the remainder of this paper. Additional information on the ABELS framework can be found at <http://thayer.dartmouth.edu/~abels>.

The keyword and conversion databases provide users with keywords, units, and file types to accurately describe services and queries. The conversion database also contains conversions between related units or file types; thus, a consumer can receive data in meters even if its matching service function produces data in feet.

There may be multiple clouds running at a particular point in time, and each cloud has an administrator that sets the policies of the cloud. For example, one cloud may be set up for general use by anyone at a particular location, such as a college or university. Another cloud could be set up for use by researchers in a particular field, such as cancer research or oceanographic modeling. Although it is possible for consumers and producers to belong to multiple clouds, queries and service functions do not cross cloud boundaries. That is, a query defined in a particular cloud will match only those service functions belonging to the same cloud.

## 2.2. Related Work

The ABELS system is not the only one with the goal of enabling the interoperability and reuse of simulations. Other systems include the High Level Architecture (HLA) (Dahmann et al. 1998), the web services framework (Curbera et al. 2002), the First All Modes All Sizes (FAMAS) project (Boer et al. 2002; <http://www.famas.tudelft.nl>), and the Extensible Modeling and Simulation Framework (XMSF) (Brutzman et al. 2002).

In HLA, participants called federates participate in a federation of interacting simulations. Participants must be written to meet the federation object model (FOM) standard, and all interactions between federates occur through the runtime infrastructure (RTI), which acts as a distributed operating system for the federation. HLA is designed for tightly-coupled interactions, while ABELS is designed for loosely-coupled interactions.

The web services framework (Curbera et al. 2002) consists of a collection of protocols and standards for communication (SOAP), service description (WSDL), and service discovery (UDDI). Although it was not designed specifically for simulation, the web services framework can be applied to simulation interoperability. However, it does not directly support the runtime brokering or matching of data consumers with producers, and this runtime matching ability is a key feature of the ABELS system.

The Extensible Modeling and Simulation Framework (XMSF) applies web-based technologies, such as XML-based languages and the web services framework, to create standards which allow modeling and simulation interoperability between distributed systems (Brutzman et al. 2002). Like the application of the web services framework to simulation interoperability discussed above, the XMSF does not support runtime matching and brokering of resources.

The FAMAS Simulation Backbone (Boer et al. 2002; <http://www.famas.tudelft.nl>), like ABELS, is designed to support interoperability between different and distributed simulation models without requiring adherence to HLA standards. However, the FAMAS approach requires a predetermined scenario script to control simulation runs. In contrast, ABELS is adaptable to changing simulation resources at runtime, allowing flexibility to changing circumstances.

## 3. GOALS OF MATCHING AND RANKING

The goal of matching and ranking is to find the best available service to resolve a given query at runtime, with no prior knowledge of the data format that a particular service uses, or even where the resource is located. Data producers and consumers have no information about each other in advance of the matching and ranking process, and need not conform to any particular standards. Because several very similar services may suit a given query, finding the best service is often a difficult task. To optimize their performance, the matching and ranking processes are implemented in the generic local agent, although both are logically part of the brokering system.

Each service is defined in terms of what functionality it offers in the form of one or more service functions. Each service function is defined as a sequence of input and output variables, with information on data types, units, ranges, and subsets. A query,

seeking to find a single service function, also defines these parameters, and both query and service define a short description. These variables and descriptions are the sole basis for matching and ranking, and therefore consistency when registering services and queries is essential.

Two characteristics distinguish the ABELS matching and ranking system from similar approaches to linking simulations and other data services: its loosely-coupled nature and its capacity for runtime brokering. Unlike the High Level Architecture (HLA), ABELS is designed as a loosely-coupled system. A service is matched to a query solely on the basis of inputs, outputs, and user-defined descriptions, allowing ABELS to abstract both service and query from their implementation details. Unlike the web services framework, ABELS allows for runtime matching of services to queries, allowing the system to adapt to changes in the availability of networked resources.

#### 4. DESCRIBING SERVICES AND QUERIES

The user entities in the ABELS cloud can produce data, consume data, or both produce and consume data. Data is produced through registered services, and requests for data are specified through queries. Each service and query is defined with information specifically provided for the matching and ranking process.

The first step in the matching and ranking process is the definition of the service in question. Generally this will be done only once for any given service, as the definition persists and can be automatically re-registered if the service goes offline and later comes back online again. If the service itself changes, so must its definition, though it need not be wholly rewritten.

A service consists of one or more related functions that are offered from the same computer. For example, a service may be defined to offer access to a weather database, and the individual functions would offer particular information from that database, such as the temperature or wind speed on a certain day at a certain location. Each function has its own data flow, taking in the input information needed to retrieve or calculate its output information. These service functions are matched against the individual queries.

The service represents everything the user entity offers to the cloud, and its description should include details that are common to all of its functions. This includes group membership, keyword information, and a text description. This text description contains information about the origin of the service, the computations performed, any relevant equipment (such as a sensor setup), and anything else needed to describe the service. This description should ideally be as detailed as possible to facilitate the most informed matches.

Each function in a service has its own name and description, and has a precise description of the input data the function takes, the output data it gives, and details of variables, the individual data items. These details include variable order and grouping, data range, and unit or file type. Figure 2, below, shows an example set of variables, split into input (top) and output (bottom) for a sample function. A variable could be a number, a date, a string of text, or even a file. Each variable has a name and a description, both for matching and for human readability.

An example might better illustrate the distinction between a service and its service functions. If we were to have a database of tide measurements in a local bay going back a century, there would be certain functions we could offer based on that database. One function might calculate the average high tide mark over a given span of time. Another might simply give a table of tide values over a given month. Another might give the lowest tide reading for an entire year. Each function shares the database but retrieves different information from it and performs different calculations.

In a similar manner, the query is written as though specifying an ideal function, from the name of the ideal service providing the function to the specifics of the data flow. The user might specify a preferred system or sensor setup, for example, and the ranges on its variables would be based on custom factors.

The process of defining a service is done through the user interface to the GLA. It is done on a partly *a la carte* style, where the component pieces of a service are defined first, and then the service functions are built from them.

Description	Repeat	Type	Subset	Range	Units
Year	1	int		[1907,2003]	year
Month	1	string	{"jan", ...}		
Num Rows	1	int		(0, inf)	unitless
Tide Mark	Num Rows	float		(0.2, 40.3)	inches

Figure 2: Sample List of Variables

The process starts with the name and definition of the service. Individual clouds may have guidelines for how these are written, such as to correspond to a certain schema, but ABELS itself requires only that they be plain ASCII text. The information here is common to all the functions defined as part of this service, such as location, database specifications, or contact person.

In the tidal measurements example, the service description needs all of the general information about the database, regardless of the individual functions it offers. That description might consist of the following text: “This service offers tide information for [the fictional] Wheelock Bay, from a database maintained at Dartmouth College using information collected by the National Weather Service from March 1907 to present. All measurements were taken at Hanover, New Hampshire.”

The process continues by defining variable parts. Variable name/description pairs are written at this stage without associating them with an actual variable. There are three additional parts that can be defined here. The range is defined as a single numerical range in the style (min,max) or [min, max], allowing “inf” to indicate no defined upper bound beyond that of the type. The range is unitless as defined, and may be matched with different units. For example, the range “non-negative”, defined as [0,inf), could be useful for many units. Defining ranges separately reduces repetition in creating the variables themselves.

A subset represents a finite list of allowable values that a variable may take. This will most often be found with string variables, such as {“north”, “south”, “east”, “west”} or {“true”, “false”}. Numbers are also allowed as subsets, such as the set of odd integers between 1 and 20. Numerical subsets, like ranges, are unitless.

The next step in our example, then, would be to define the variable parts to use. Variable names would include “month”, “year”, and “tide mark” with short text definitions such as “Tide height at the Hanover measuring station”. Because the years are constrained by the measurement period, we could define a range “years period” as [1907,2003]; however, this would necessitate changing the service definition once 2004 data is available, so [1907,inf) is another possibility if the database interface can accept and properly deal with dates after the last entry. Similarly, we could define a “non-negative” [0,inf) range for use with the tide mark, or search the database to see what the actual global high and low values are to give a more exact output range. Using an actual range such as (0.2, 40.3) would be more likely to properly match query definitions, but would have to be updated if the database ever receives an entry higher or lower than that.

Month values could of course be integers, but if our particular database requires the three-letter month

abbreviations, that can be accomplished using a subset “three-letter months” of the strings {“jan”, “feb”, “mar”, ..., “dec”}.

A unit or file type can be either user-defined or selected from a list of pre-defined units and file types. User-defined units are allowed, but may not allow conversion to other units or file types; a match in this case would have to be identical, which is generally acceptable for files. The pre-defined units and file types, however, are defined in the conversion database maintained on a cloud-wide basis by the broker, and have conversion routines defined to translate data from one unit to another (e.g., inches to meters) or one file type to another (e.g., MS Word to LaTeX). These conversion routines may considerably expand the pool of possible matches to a given query.

When these variable parts are defined, the user may build variables from them and define the input and output variable lists for each function. A variable must have a name, description, and type such as integer, floating point value, date, file, or text string. Units, ranges, and subsets may be associated with a variable here but are not required. Each variable also has a repeat value, which can be either a number or a reference to the value of a previously defined variable. The single variable, then, becomes a column of data of either fixed length or of a varying length to be specified at runtime.

The variable order matters; when defining a service, the input variables should be in the order that the service expects the data, and the output variables should be in the order in which the service returns the data. In defining its ideal service, a query would assume that the service takes the input data in exactly the order that the consumer entity gives its data, and the output in the order that the consumer expects its results.

Defining variables in this way is perfectly acceptable, but may become repetitious if similar variables are to be defined for multiple functions in the service. To avoid this repetition, the user may save individual variables or groups of variables as patterns. The first use of patterns is simple reuse: once a pattern is defined it may be used multiple times in multiple functions. The pattern name and a repeat value are all that is necessary to add a variable or group of variables to the list. Because patterns can contain multiple variables, a pattern can be used to quickly and easily define multi-column tables of data.

Returning to the example, we know that the variable representing a tidal reading will be used multiple times, so we define a pattern named “tidal reading”. We use the name/definition pair “tide mark”, declare it of type floating-point number, with the “non-negative” range, and we select the pre-defined unit “inches”. We also give the pattern a repeat value of 1

because this is a single-variable pattern. Because year and month are also variables we will use repeatedly, we define appropriate patterns for them as well, “year” as an integer with range “years period” and pre-defined unit “years”, and “month” as a text string with subset “three-letter months” and unit “months”.

Defining these patterns will make it easier to define input and output lists. One service function offers the lowest tide reading for a year, so we define an input list with just the “year” pattern and an output list with just the “tidal reading” pattern, with one repetition of each. Another function offers a table of tidal values for a given month. The input list for that function would just be one instance of “year” and another of “month”. For the output list, the database gives a column of readings preceded by the number of rows. To define this, we can go back and add a name/definition pair “number of rows”/“The number of rows in the following table”, and add a variable to the output list with that name, of type integer, range “non-negative” (i.e., (0,inf)), and unitless. After that, we add the pattern “tidal reading” with a repeat value listing the variable we just defined. This is interpreted as repeating this pattern a number of times to be determined at runtime by the value of the variable “number of rows”.

The next step in registering a service is to define each service function in terms of its description and its input and output lists, and save both lists to the service definition. The function descriptions should describe the calculations or data retrieval performed and more general information not included in the service description, such as average response time. The service keywords are then selected, and the service can then be registered in any available groups the user desires.

Concluding the example, we assemble the functions from the input and output lists we just defined, together with a short description for each function, such as “This function returns the lowest tide mark over the entire given year.” Just like the variable parts and patterns, we can reuse the input and output lists. For instance, we defined an input containing the year and an output containing a single tide mark, which we could easily reuse for a function to offer the highest tide mark for the year. Once the functions are assembled, we select the keywords “oceanography”, “New Hampshire”, “tidal measurements”, etc., and register the whole service in the “oceanography” and “Dartmouth College” groups.

The service definition will persist as long as the producer GLA is connected to the cloud. When a subsequent query is registered in any of this service’s groups, the service description will be returned to the consumer GLA for matching and ranking. This group-based approach is the first-level lookup that is performed by the broker. Each service that is returned to the consumer GLA is examined and its functions

ranked in order to determine which service functions to use for resolving queries.

## 5. THE RANKING PROCESS

The ranking process provides a basis for quantitative comparison among services by assigning a numerical rank to every service function that might satisfy a particular user-defined query. This numerical rank, a number between 0 (a non-match) and 1 (a perfect match), is the weighted average of several factors, each of which reflects some aspect of the fitness of a particular function for a particular query. Each individual factor has a value between 0 and 1.

The ranking process begins when a query is registered with the cloud. The broker performs a first-level lookup based on the groups of interest defined in the query specification, and sends service information to the consumer GLA for every service that has joined any of the groups of interest. This first-level lookup is the first of the two steps in the matching and ranking process.

This collection of services will vary greatly in terms of the functionality actually being offered. Some of the services may contain functions that match the query, but many will not. The goal of the second-level matching and ranking process is two-fold; it must determine which service functions are appropriate matches for the query, and it must determine which of the appropriate matches is the best match.

In determining the rank for a service function, the matchmaker in theory achieves both goals. The rank, a number between 0 and 1, indicates the relative fitness of the function in satisfying the query. In order to best satisfy its goals, the ranking process will be carefully tested and adjusted so that the ranks of inappropriate services are all clustered near 0, and the ranks of the appropriate services near 1, with very few services in between. In that way, the rank distribution will be considered a general indicator of whether a given service function is suitable or unsuitable.

In second-level matching and ranking, we first consider the keywords defined for both query and service. We compute the percentage of key words in the query description that are also contained in the function description. This percentage is one of the weighted factors in our comprehensive rank.

Second, we consider the groups defined for both query and service. For one of the weighted factors in our comprehensive rank, we compute the percentage of groups in the query description that are also contained in the function description, which is also one of the weighted factors in our comprehensive rank.

Finally, we assess the mapping success between function and query, that is, the degree to which the

query specification is consistent with the function specification. In this stage of ranking, we consider the input and output variables for both service function and query. A function that is well-mapped to a particular query will contain the input and output variables specified in the query, measured in units compatible with the units specified in the query definition.

The mapping itself results from a variable-to-variable comparison where, in principle, each variable in the query is compared to each variable in the service in an effort to determine which service variable, if any, corresponds to it. On the left side of Figure 3, each directed edge represents such a comparison between variables for the query (Q) and service function (S). In practice, there is no reason to compare a number to a text string or to a file, or a file to a date, or any single variable to a table of variables. We can exploit this to reduce the number of comparisons we make by assigning to each variable a compound type, where all of the numbers (whether integer or floating point) are taken together, all the files, strings, dates, and arrays are each taken together, for a total of five compound types. Only variables of the same compound type are compared.

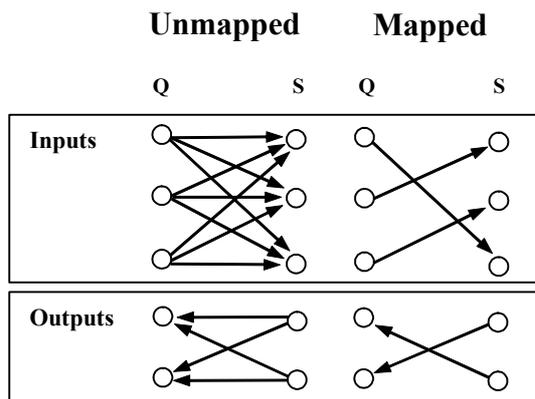


Figure 3: Unmapped and Mapped Variable Sets

The right side of Figure 3 represents the ideal outcome of the mapping process: Each variable in the query has been linked to a corresponding variable in the output. However, most of the functions being evaluated will not be perfect or even good matches; they are simply not matches suited to the query. These functions will often have different numbers of variables than the query, and so a one-to-one matching will not be possible or even desirable. Even if the query and service function have the same number of variables, if those variables are not describing the same data, they should be not be considered to correspond. It would be better to leave variables “orphaned” or unmapped than to map a query variable incorrectly as an indication of suitability.

When each variable pair is compared, we consider several factors. First, the name and description of the

variables are compared for similarity. Then, the units of the two variables are compared. If they are the same, that is a fair indication of a match, but if they are different they may still match; for example, one may be in centimeters and the other in inches. The conversion database comes into play here, determining whether a conversion path exists between the two units, and if so, how long it is. Because the conversion database is linked internally according to scale, a long conversion path would indicate a likely mismatch, such as from micrometers to nautical miles, which are both measures of length but several orders of magnitude away from each other.

The data flow indicates whether a match exists for the range or subset. The chief criterion here is whether one entity may be producing data that is out of range for the other. Data flows from consumer input to service input, and from service output to consumer output. The ranges for the consumer inputs, then, should match or fit inside the ranges of the service inputs, and the consumer output subsets should be a subset of those of the service output. For example, the consumer could offer {“north”, “northeast”, “east”, “southeast”, etc.} but if the service accepts only {“north”, “south”, etc.} then the consumer could be providing input data that the producer does not recognize. A service that does not recognize data provided by the consumer would be penalized in the ranking process.

Any function that is not well mapped to a particular query faces a two-tiered penalty to its rank. Functions that do not produce output variables specified in the query are penalized heavily, as are functions that require input variables not specified in the query. Similarly, functions that produce output variables not specified in the query are penalized to a lesser degree, as are functions that do not require input variables specified in the query, and functions that contain units not convertible to units specified by the query. All such penalties are assessed on a perfect rank of 1.0, and the result is the final, and most important, factor in our weighted average.

By design, any function that is ranked has been returned by the first-level matching and therefore must define at least one group that is also defined in the query description. Accordingly, any ranked function must have a positive rank. We reserve the negative and zero ranks for ranking errors.

In addition to the automatic ranking, the user has an opportunity to examine the service functions and their assigned ranks. The first level of control that a user has is to designate certain services as more or less desirable than others. By labeling a service as *preferred*, *deprecated*, or just *unsuitable*, the user can determine the order in which the GLA will select services during query resolution. The second level gives the user control over the ranking weights themselves, such as the

weight of the penalty given to missing variables or missing keywords.

## 6. THE QUERY RESOLUTION PROCESS

Once the query has been defined and registered, and the services returned from the first-level lookup have been ranked, the GLA is ready to resolve queries. The resolution process, which is shown in Figure 4, begins when the consumer entity connects to the GLA, provides the name of the query to be resolved, and sends the service input as a single data stream (Step 1).

The first task in the resolution process is to select the service to use. The GLA first looks at all of the services that the user has marked preferred, and selects from that list the service function with the highest rank. If that service is unavailable, it steps through the list of preferred services until it has exhausted them, and only then goes to the unmarked services, then to the list of deprecated services. If no services can be found in these three lists, the resolution fails rather than select an unsuitable service.

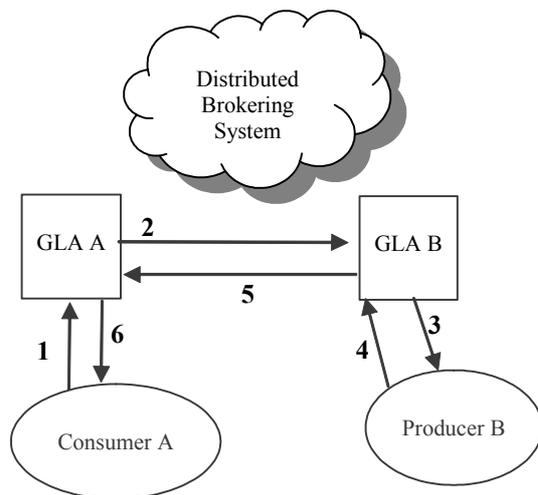


Figure 4: Data Flow Between Producer and Consumer During Query Resolution

When a service function is selected to satisfy the query resolution, the consumer GLA examines the mapping that was generated during the ranking process. It uses this information to determine how to rearrange variables and determine which variables to omit. The GLA extracts the data from the consumer's data stream, divides it into variables, rearranges as needed, and performs any necessary unit and file conversions. It formats this into a single data stream appropriate for the producer software, and sends it to the producer GLA (Step 2). The producer GLA contacts the producer entity (Step 3) with the data stream formed by the consumer GLA, and waits for the producer to return its output data stream (Step 4). This data is passed back to the consumer GLA (Step 5). Note that the work done by the producer GLA is kept to a minimum, to keep the burden on the producer side as small as possible.

The consumer GLA disassembles the output data stream according to the variable list in the service definition, and uses the mapping to reassemble the data for the consumer. It performs unit and file conversions, and reforms the data into a stream to return to the consumer entity, which may be waiting for the data (as a blocking query) or planning to contact the GLA at a later time to collect the data (as a non-blocking query).

## 7. CONCLUSIONS AND FUTURE WORK

The current implementation of the matching and ranking system is only partially complete; currently, it ranks on the basis of the service description, keywords, text descriptions, and number of variables. In the near future, the mapping process will be implemented, as will the conversion database. Further work will be required in several areas, such as finding more-sophisticated ways to match the text descriptions, adding user control, and determining the optimal weighting for different aspects of the ranking process.

Individual user groups may require a more customized system. Researchers whose simulations adhere to the SEDRIS format, for example, would require unit conversion to and from its Environmental Data Coding Specification (EDCS) standard for units. (Additional information on SEDRIS can be found at <http://www.sedris.org>.) Some unit schemes will be included in the standard ABELS conversion database, but new standards or ones that are not widely used may also be needed by certain users. To accomplish this, a flexible administrative interface for the conversion database will allow those groups managing an ABELS cloud to specify any number of units and file types and the conversion routines among them, without worrying about inconsistencies among service and query definitions.

## ACKNOWLEDGMENTS

This work is supported by National Science Foundation KDI Grant 9873138 and U.S. Army Corps of Engineers contract DACA42-01-P-0288.

## REFERENCES

- Boer, C.; Y. Saanen; H. Veeke; A. Verbraeck. 2002. "Final Report, Project 0.2 - Technical Design, Simulation Backbone FAMAS.MV2." TRAIL Research School, Delft.
- Brutzman, D.; M. Zyda; J. M. Pullen; K. Morse. 2002. "Extensible Modeling and Simulation Framework (XMSF), Challenges for Web-Based Modeling and Simulation", Findings and Recommendations Report: Technical Challenges Workshop, Strategic Opportunities Symposium (Fairfax, VA, 22 October, 2002), 1-52.
- Curbera, F.; M. Duftler; R. Khalaf; W. Nagy; N. Mukhi; S. Weerawarana. 2002. "Unraveling the web services web: an introduction to SOAP, WSDL, and UDDI." *IEEE Internet Computing* 6, No.2, (Mar./Apr.), 86-93.

- Dahmann, J.; F. Kuhl; and R. Weatherly. 1998. "Standards for Simulation: As Simple as Possible but Not Simpler, the High Level Architecture for Simulation." *Simulation*, 71, No.6 (Dec.), 378-387.
- Kumaran, S.I. 2002. *Jini Technology: An Overview*, Prentice-Hall, Upper Saddle River, N.J.
- Mills-Tettey, G.A.; G. Johnston; L.F. Wilson; J.M. Kimpel; and B. Xie. 2002. "The ABELS system: designing an adaptable interface for linking simulations". In *Proceedings of the 2002 Winter Simulation Conference*, Volume 1 (San Diego, CA, December 8-11), 832-840.
- Mills-Tettey G.A. and L.F. Wilson. 2003a. "Security issues in the ABELS system for linking distributed simulations". In *Proceedings of the 36<sup>th</sup> Annual Simulation Symposium*, (Orlando, FL, Mar. 30 – Apr. 2). IEEE, Piscataway, N.J., 135-144.
- Mills-Tettey, G.A. and L.F. Wilson. 2003b. "A Security Framework for the Agent-Based Environment for Linking Simulations (ABELS)". *Simulation*, to appear.
- Murphy, J.P.; G.A. Mills-Tettey; L.F. Wilson; G. Johnston; and B. Xie. 2003. "Demonstrating the ABELS system using real-world scenarios". In *Proceedings of the 2003 SAINT Conference*, (Orlando, FL, Jan. 27-31). IEEE, Piscataway, N.J., 74-83.
- Wilson, L.F.; D.J. Burroughs; A. Kumar; and J. Sucharitaves. 2001. "A framework for linking distributed simulations using software agents". In *Proceedings of the IEEE 89*, no. 2, (Feb.), 186-200.
- Wilson L.F.; B. Xie; J.M. Kimpel; G.A. Mills-Tettey; and G. Johnston. 2002. "The Design of the Distributed ABELS Brokering System". In *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT)* (Fort Worth, TX, Oct. 11-13). IEEE, Piscataway, N.J., 151-158.
- Wilson, L. F.; W. R. Lochridge; and G. A. Mills-Tettey. 2003. "The Secure ABELS Brokering System". In *Proceedings of the 15th European Simulation Symposium* (Delft, The Netherlands, Oct. 26 - 29), SCS, San Diego, CA, to appear.

## AUTHOR BIOGRAPHIES

**JOSHUA O. PETEET** is a master's student at Dartmouth's Thayer School of Engineering. He received his AB degree in computer science from Bowdoin College in 2002. His email address is [Joshua.O.Peteet@dartmouth.edu](mailto:Joshua.O.Peteet@dartmouth.edu).

**JOHN P. MURPHY** is a PhD student at Dartmouth's Thayer School of Engineering. He received his BS degrees in computer engineering and electrical engineering from West Virginia University in 2001. His email address is [John.P.Murphy@dartmouth.edu](mailto:John.P.Murphy@dartmouth.edu).

**LINDA F. WILSON** is an associate professor at Dartmouth's Thayer School of Engineering. She

received her BS degree in mathematics from Duke University in 1988 and her MSE and PhD degrees in electrical and computer engineering from the University of Texas at Austin in 1990 and 1994, respectively. Her email address is [Linda.F.Wilson@dartmouth.edu](mailto:Linda.F.Wilson@dartmouth.edu) and her web page can be found at <http://thayer.dartmouth.edu/~lwilson>.

# THE SECURE ABELS BROKERING SYSTEM

Linda F. Wilson, W. Riley Lochridge, and G. Ayorkor Mills-Tettey  
Thayer School of Engineering  
Dartmouth College  
Hanover, NH 03755-8000 USA  
E-mail: Firstname.Lastname@dartmouth.edu

## KEYWORDS

Distributed simulation, brokering systems, security, dynamic information exchange, software agents, simulation tools.

## ABSTRACT

The Agent-Based Environment for Linking Simulations (ABELS) system is a software framework whose goal is to enable independent simulations and other data resources to exchange information dynamically without prior knowledge of each other. Specifically, it enables the dynamic formation of a “cloud” of simulations and other networked resources such as sensors and databases. This data and simulation cloud uses a distributed brokering system to match data consumers in the cloud with appropriate data producers, based on registration information submitted by the various participants in the cloud.

In a system of interacting independent resources, there are several security concerns, including how to prevent undesirable entities from joining and participating in the cloud, how to protect sensitive information, and how to ensure the integrity of the cloud so that it functions reliably. This paper presents the redesign of the ABELS brokering system to incorporate security features that address these concerns.

## 1. INTRODUCTION

The Agent-Based Environment for Linking Simulations (ABELS) system is a software framework whose goal is to enable independent simulations and other data resources to exchange information dynamically without prior knowledge of each other (Kumar et al. 2002; Mills-Tettey and Wilson 2003a; Mills-Tettey and Wilson 2003b; Wilson et al. 2001). The ABELS system allows a collection of independent networked resources to associate with each other in what is referred to as a data and simulation *cloud*. The autonomous resources are producers and/or consumers of data. Individual cloud participants join and exit the data and simulation cloud as needed and have no prior knowledge of the other cloud participants. A distributed brokering system is used to match data producers to consumers and initiate communication between cloud participants, but it does not control the independently-designed participants in any way. Each organization

participating in an ABELS cloud is responsible for determining what resources it makes available to the cloud and which data consumers are eligible to access its services.

The ABELS system is designed for loosely-coupled interactions between participants. That is, participants in the cloud are not required to conform to a common stringent standard, and consumers are not statically linked to particular producers of information. In addition, there are no tight interdependencies among cloud participants.

There are many security concerns inherent in the ABELS system. These include concerns about how to protect undesirable entities from joining and participating in the data and simulation cloud, how to protect sensitive information produced by services in the cloud from being accessed by unauthorized entities, and how to ensure the integrity of the cloud as a whole so that it functions reliably. To mitigate the security threats in the system, the ABELS architecture must include mechanisms for access control, privacy and integrity, and logging. In addition, each cloud participant must be able to define its own security needs and capabilities, and the cloud must guarantee that each participant’s security requirements are met.

As discussed in Mills-Tettey and Wilson (2003a; 2003b), we have recently redesigned the ABELS system to incorporate appropriate security mechanisms. Security experts have noted that security must be designed into a system’s architecture rather than added at a later date. Thus, security features have been integrated throughout the ABELS system.

This paper describes the redesign and implementation of the secure ABELS brokering system, which is responsible for maintaining a database of cloud participants and matching data consumers with appropriate data producers. Section 2 describes the various components of the ABELS system, while Section 3 discusses related systems and Sun Microsystems’ Jini technology, which is used by ABELS. Section 4 presents the design of the secure brokering system while Section 5 discusses its implementation. Finally, Section 6 presents conclusions and areas for future work.

## 2. THE ABELS SYSTEM

### 2.1. Overview

The Agent-Based Environment for Linking Simulations (ABELS) is a software framework that enables the dynamic formation of a cloud of autonomous simulations and other data resources, which can then interact without prior knowledge of other cloud participants. As shown in Figure 1, the ABELS system architecture consists of three basic types of components: user entities, generic local agents (GLAs), and a distributed brokering system. An optional user interface is provided to permit human interaction with the system via the GLAs.

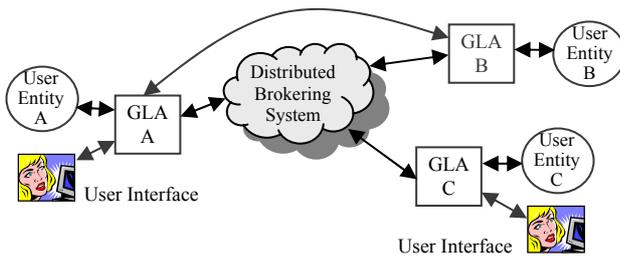


Figure 1: Basic Framework Connecting Elements in the Cloud

The cloud participants are the *user entities* which are producers and/or consumers of data. For example, a cloud might include simulations, databases, and sensors. A data producer provides a *service* with one or more *service functions*, while a data consumer makes requests or *queries* for information. User entities join and exit the data and simulation cloud as needed and have no prior knowledge of the other cloud participants. Each organization participating in an ABELS cloud is responsible for determining what resources it makes available to the cloud and describing those services accurately.

A user entity connects to the cloud via a software agent known as the *generic local agent* (GLA). Each GLA serves as the cloud interface for one or more producer and/or consumer entities, owned and operated by the same organization. The user entity uses its GLA to join or exit the cloud, register its services and service functions, and make queries for data. A data producer is said to interface with the cloud via a *producer GLA*, while a data consumer participates via a *consumer GLA*. The GLA is responsible for adapting information to the specific needs of its producer or consumer entities. For example, a consumer GLA is responsible for handling any data format, unit, and file conversions that are necessary between its consumer and the producer that is serving it. A producer GLA is responsible for passing input data to a desired service, executing the desired service function, and returning the output data to the corresponding consumer GLA.

The *distributed brokering system* forms the core of the cloud and is responsible for managing cloud resources and linking consumers to appropriate producers of information. In particular, it stores descriptions and remote references or *proxies* for all of the resources in the system, and it uses a system of leases to determine which participants are still in the cloud. Furthermore, it matches and ranks those service functions that may satisfy a given consumer's query. Once the brokering system links a consumer GLA with a corresponding producer GLA, communication occurs directly between the GLAs without going through the brokering system. In Figure 1, GLAs A and B have a direct connection, indicating that a match occurred between a producer on one end and a consumer on the other end. The brokering system is described in detail beginning in Section 4.

A key feature of ABELS is its runtime matching of consumers with suitable producers, and this matching is based on textual descriptions of the participants. A service definition includes the name, location, and description of the service along with detailed information about the functions it provides. A service definition also specifies one or more high-level categories or *groups* that characterize the service. For example, a service would belong to the "weather simulations" group if it provides information about the weather conditions in Hanover, New Hampshire. A query is defined as the ideal function desired by the consumer. For a given consumer query, ABELS must evaluate all service functions that may match the query and rank those functions so that the most suitable one is chosen to resolve the query. A query is resolved using the best-available service function, so a query resolved by one function today may be resolved by a different function next week.

For better accuracy and efficiency, there are two levels of matching in the ABELS system. The first-level matching is based on the groups of interest for the given query. That is, the brokering system generates a list of those services that belong to the groups specified in the query definition. In the second-level matching, the details of the query are compared with all of the service functions belonging to the list of services, and the service functions are ranked according to their suitability for the query. Details of the matching and ranking process can be found in Peteet et al. (2003). Additional information on the ABELS system can be found at <http://thayer.dartmouth.edu/~abels>.

### 2.2. Security Issues

As discussed in Mills-Tettey and Wilson (2003a; 2003b), we have recently redesigned the ABELS system to incorporate various security features. In particular, the participants may have varying security needs, and the cloud must guarantee that each participant's security requirements are met.

The specific security capabilities in ABELS can be classified as access control capabilities, privacy and integrity capabilities, and logging capabilities. Access control consists of the related processes of authentication and authorization. Authentication verifies the identity of an entity while authorization determines what permissions are granted to the authenticated entity. Privacy protects sensitive information from being viewed by unauthorized parties, while integrity provides the ability to detect and protect the information from tampering. Finally, logging is used to detect and audit a security breach.

Security features in ABELS appear at two levels: *centralized* aspects are handled by the brokering system, while *decentralized* aspects are managed by the individual GLAs in the cloud. The brokering system security focuses primarily on ensuring the integrity of the cloud, while the GLAs provide the primary means of protecting the entities participating in the cloud. In particular, a GLA may authenticate the users that wish to use it to communicate with the cloud, to prevent access by unauthorized users. In addition, a producer GLA may place limits on the consumer GLAs that wish to use its services. The brokering system may authenticate GLAs before allowing them to join the cloud, in order to prevent unauthorized participation by an illegal GLA. Encryption can be used to protect sensitive information passed between a GLA and its user entities, between two GLAs, or between a GLA and the brokering system. Finally, both the brokering system and the GLAs log all successful and unsuccessful attempts to add GLAs to the cloud or connect user entities to the GLAs.

The complete ABELS security framework is discussed in Mills-Tettey and Wilson (2003a; 2003b). This paper presents the architecture of the secure brokering system, and specific security details are given in Section 4.3.

### 3. RELATED WORK

#### 3.1. Other Approaches

ABELS is not the only system designed to create a dynamic, distributed community of heterogeneous entities, simulations, and services. Other approaches include the High Level Architecture (HLA) and the web services architecture.

The High Level Architecture (HLA) (Dahmann et al. 1998) is a software architecture for creating a federation of simulations communicating across a single runtime infrastructure (RTI). HLA is designed to support tightly-coupled interactions between simulations. A set of rules specifies the object models to which federations (communities of simulations) and federates (individual simulations) must conform in order to work with other entities in the system. To participate in an HLA federation, simulations must be written to conform to the

federation object model (FOM).

Unlike HLA, the web services architecture (Curbera et al. 2002) provides a mechanism for communication and data exchange between loosely-coupled entities. WSDL (Web Service Description Language), UDDI (Universal Description, Discovery and Integration), and SOAP (Simple Object Access Protocol) are three XML-based protocols that enable the description, registration, and invocation of web services. XML provides a language- and platform-independent communication mechanism that is an important feature of the web services architecture. Missing in the current architecture, however, is a mechanism for the runtime brokering between information consumers and producers that would allow the transparent replacement of one service provider with another one.

The ABELS system described in this paper targets loosely-coupled simulations and data resources, requiring little or no changes to existing simulations. Developed with Java and Jini, ABELS is designed to be platform-independent and currently is being developed on both UNIX and Windows systems. A brokering system is used to perform runtime matching of data producers and consumers, and software agents act as interfaces through which entities can participate in the ABELS cloud. These software agents enable the ABELS system to adapt to a wide range of simulations and resources without requiring these entities to be written in a specific language.

#### 3.2. Jini

Our distributed brokering system is developed in part using Sun Microsystems' Jini technology (Kumaran 2002). Jini is a protocol-independent, Java-based programming model that simplifies the development of a system of distributed services. In particular, the Jini application programming interface (API) handles tasks associated with the discovery and lookup of distributed services and the description of service attributes. Jini also provides event listeners and leases that are used to manage the networked resources. The ABELS brokering system uses the building blocks provided by the Jini architecture and its reference implementation to manage the cloud resources and match data consumers with suitable producers.

### 4. THE SECURE BROKERING SYSTEM

#### 4.1. Overview

The ABELS system links loosely-coupled data resources to form a distributed data and simulation cloud. The brokering system serves as the central framework through which the cloud is formed and maintained. To ensure efficiency and fault tolerance, the brokering system is designed using distributed database principles.

As discussed in previous work (Kumar et al. 2002; Wilson et al. 2002), the ABELS brokering system must serve as a database of participating services and match consumers with suitable producers. However, a better analogy compares the brokering system to a public or school library. Just as a library maintains and organizes its collection of books, the brokering system stores and categorizes each of the services registered within it. Just as books are organized into broad subject categories, services are organized into high-level categories called *groups*. Just as a librarian may take a reader's description of interests and recommend an appropriate book, the brokering system uses a detailed description of the desired service to match the consumer to the producer that best meets its needs. Furthermore, just as a borrower must obtain a borrowing card and present it to check out a book, so must a producer or consumer undergo identity-based authentication and authorization before joining or using services of the cloud.

The brokering system consists of the *broker*, the *matching and ranking system*, and the *keyword and conversion databases*. The broker manages the resources in the cloud, the matching and ranking system evaluates producers on behalf of consumers, and the keyword and conversion databases provide information needed for the matching process. The broker is implemented using Java and Sun's Jini technology, and the other components are implemented using Java.

The current implementation of the ABELS brokering system includes several Jini lookup services, each of which can run on a separate machine and supports several high-level categories called groups. The lookup service plays the role of registrar, or card catalog, for the cloud, managing a persistent database of all cloud participants by storing their service descriptions and proxies, which are described in Section 4.2.

The lookup service facilitates automated system startup and cloud formation by utilizing Jini's *discovery* and *join* protocols. It does this by providing its proxy to all GLAs as they join the cloud, allowing them to initiate contact at any later time. The lookup service also provides the means by which ABELS maintains a robust and self-healing network.

Due to the potential volatility in any distributed network, the ABELS cloud must be long-lived and resistant to sudden changes in the system that could initiate a system crash. The lookup services help ensure this by storing registration information for all GLAs in the cloud and discarding service proxies of lost GLAs. The lookup services support the join protocol, which assigns unique service IDs to all cloud entities. All GLAs in the cloud must conform to this protocol and maintain, for each service, the service ID, the list of groups it wishes to support, its Jini lookup entries, and the set of locations of the lookup services in which it has

registered (Li 2000). The lookup services also grant Jini *leases* for an entity's GLA once it has joined the cloud. A lease represents the GLA's proof of interest, allowing the entity to remain registered in the cloud as long as its lease has not expired. The GLA must renew its lease within the finite lease duration, according to the terms set by the lookup service. Note that the administrative overhead of handling leasing is distributed away from any one central component by placing it in the lookup services, thereby increasing fault-tolerance and reducing the potential for bottlenecks.

## 4.2. Services and Queries

Having joined the ABELS cloud, a producer GLA may register any of its services with multiple Jini lookup services, based on the groups the services wish to join. To locate the lookup services with which it wishes to register, the GLA contacts the broker's communication module, which finds and returns the locations of the lookup services that support the desired groups. Before a service can be registered, the GLA must provide the lookup service with the following information: the group(s) it wants to support, a service description, and a service proxy object that is used by a consumer GLA to access the service. Within each service description there may be multiple function descriptions. In each function description, the user defines a sequence of input and output variables, including information on data types, units, ranges, and subsets. A query is a description of the ideal service desired and is in the form of a function description.

Similarly, a consumer GLA that has joined the cloud can obtain the locations of lookup services that support its desired groups. After receiving the locations of the lookup services, the consumer GLA contacts them to obtain information on the services belonging to the groups of interest. This is the first level of the two-level process that matches producers to consumers. Specifically, the consumer GLA receives the service registration and service proxy information for each service that might meet the needs of the consumer's query. A standing request is also left with the broker; if a service that supports the desired group(s) later joins the cloud, the system informs the consumer GLA via remote event handlers. Thus, the consumer GLA knows at all times which services of interest are currently in the cloud. With this information, the consumer GLA interacts with the matching and ranking system for the second-level matching. In order to understand this process, a brief description of the matching and ranking system is in order.

Matching and ranking determines the service function that best resolves a given query (Petet et al. 2003). The matching and ranking process is designed as part of the brokering system but is implemented in the GLA in order to optimize performance and facilitate cloud scalability. The matching and ranking process is

based on the service description described above, making syntactic consistency critical. Accordingly, the human user can use the keyword database to determine which keywords are appropriate, and this will increase the likelihood of a suitable match.

The loosely-coupled nature and runtime brokering capability of ABELS differentiate it from related approaches like HLA and the web services framework. ABELS is a loosely-coupled system in which a service function is matched to a query exclusively on the basis of inputs, outputs, and a user-defined description. As a result, the matching process abstracts both services and queries from their implementation details, thus allowing a multitude of different implementation methods. The runtime brokering of services to queries allows the system to adapt to changes in network availability, providing a consumer with the best possible service available at the time of query resolution.

For each service function in the groups of interest, the matching and ranking system provides the consumer GLA with a numerical rank between 0.0 and 1.0, with 1.0 being a perfect match with the function. The rank is a weighted average of several factors that reflect various aspects of the fitness of a function for the particular query. While default weights are typically used, the human user may also customize the factor weights for his particular needs. To augment the ranking, a user may designate a service as *preferred*, *deprecated*, or *unsuitable*.

When resolving a query, the consumer GLA first attempts to use the highest-ranked preferred service. If that service is unavailable, it goes through the preferred services in descending order by rank. If no preferred services are available, the consumer GLA tries to connect to unmarked services in descending order of rank. If the query is still unresolved, the process continues through the list of deprecated services. If the query is still unresolved at this point, it will fail; unsuitable services are never used to resolve the query.

Once the consumer GLA has connected to a producer GLA using the provided service proxy, communication occurs directly between the two GLAs. If the service goes down during the process of information exchange, the consumer GLA will once again attempt to resolve the query, starting with the best-available service function.

### 4.3. Brokering System Security

As described in Mills-Tetty and Wilson (2003a; 2003b), security in ABELS is designed with a two-tier approach. The brokering system provides *centralized* security, restricting access to the cloud and maintaining cloud integrity. The brokering system is responsible for determining which GLAs may join the cloud and what permissions they have as cloud participants. This

determination is known as *access control* and is comprised of the related concepts of authentication and authorization. Brokering system authentication entails the verification of a GLA's identity, while authorization determines the specific permissions granted to the GLA. The GLA itself provides *decentralized* cloud security that prevents malicious use of the GLA as an entry point for attack.

An ABELS cloud is classified as *open*, *semi-open*, *semi-closed*, or *closed*, depending on the security protocols established when the cloud is initially created. The human user creating the cloud, known as the cloud administrator, is responsible for selecting the cloud's security protocols. The cloud administrator is assisted in this process by using a graphical user interface (GUI).

In the *open* cloud, all producers and consumers must conform to specified IP address restrictions. For example, an academic institution may wish to allow any user connecting from within its IP domain to connect to the cloud while all IP addresses outside this domain are denied. While restrictions are required, they can be set to allow all IP addresses.

All cloud classifications use IP restrictions, and additional forms of authentication are used in the semi-open, semi-closed, and closed clouds. In these three cloud classifications, the default authentication method is the use of X.509 digital certificates issued by a trusted authority (usually the brokering system, although this is customizable).

In a *semi-open* cloud, all producer GLAs must undergo identity-based authentication and authorization by presenting a digital certificate. The digital certificate is verified by the broker's access control module, which is described in Section 5.1. Once the producer GLA is authenticated, the access control module checks the IP address restrictions in place, and if the entity conforms to the restrictions, it joins the cloud. Consumer GLAs, however, are subject only to IP restrictions when joining the cloud. Within such a cloud, the brokering system effectively guarantees that data obtained from the cloud is from known producers while consumers may participate under less stringent requirements. This is analogous to the access control used for a professional organization's web site, such as IEEE Xplore (ieeexplore.ieee.org), in which a data producer is guaranteed by the organization, while any user logging in from a member institution's IP domain is granted access.

In the closed and semi-closed clouds, both producers and consumers must authenticate themselves using digital certificates. Both clouds require identity-based authorization, but it is in the requisite GLA security functionality that they differ. In a *semi-closed* cloud, a participant GLA authenticates all its human users, but does not necessarily authenticate all its

consumer entities or encrypt service input and output information, as described in Mills-Tetty and Wilson (2003a; 2003b). This provides an environment in which the cloud guarantees the legitimacy of information provided but not necessarily its privacy.

In a *closed* cloud, the GLA authenticates all entities, and all information passed between producers and consumers is encrypted. The GLA’s mandatory authentication of all entities adds legitimacy to the service registration and other information the GLA provides to the cloud. The encryption ensures that unauthorized users do not obtain sensitive data. Closed clouds are appropriate for environments where security and privacy of data is of great importance, such as at a military research agency.

When creating the cloud, the cloud administrator may select from a list of available authentication methods. As described in Mills-Tetty and Wilson (2003a; 2003b), ABELS currently supports digital certificates, username/password pairs, Windows, Unix, and Kerberos authentication methods. Additional authentication features may be added at a later date.

Once approved by the brokering system for admission to the cloud, the GLA receives a Kerberos-style ticket, called an entry ticket, which is presented to any ABELS component or entity in all subsequent interactions. This ticket proves that a user has been authenticated and authorized to join the cloud, and it functions in the same way as a library card, verifying the entity’s identification and authorization and enabling it to participate within the cloud and carry out its actions. This process is shown in Figure 2, in which GLA A and the access control module must authenticate with one another using digital certificates. Once authenticated, the access control module checks that GLA A conforms to the IP restrictions and, if authorized, returns an entry ticket to the GLA. This figure also mentions the lookup service references (proxies), which would be returned to the GLA in response to a given query, as described in Section 5.3. Finally, with the lookup service proxies in hand, the GLA contacts Lookup Service 1 to either register a service or perform a first-level lookup after first presenting its entry ticket to the lookup service.

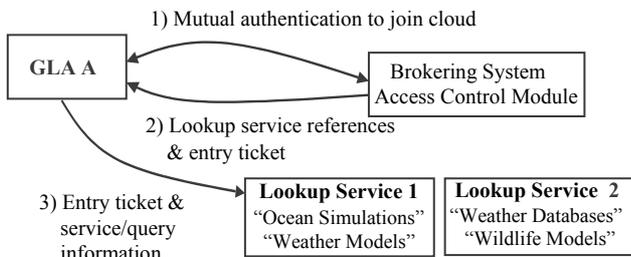


Figure 2: GLA Authentication, Authorization, and Cloud Participation

As mentioned previously, the brokering system has been redesigned to incorporate various security features.

This redesign includes a shift from multicast functionality to an exclusively unicast system. Unless otherwise specified, Jini uses multicasting in its join and discovery protocols. Multicasting is a broadcast-based form of communication in which any computer listening on a prespecified port (e.g., 4160 is used by Jini) will receive all packets sent. While multicast allows for easy bootstrapping and discovery of available resources by new entities, it represents a major security hazard in that any entity listening to port 4160 would receive Jini *announcement* packets describing the current cloud composition in terms of lookup service locations and available groups. Because of the information leak this represents, we determined that only unicast communication should be used in ABELS. The security benefit in switching from multicast to unicast is analogous to a person announcing his whereabouts and activities on a telephone versus a megaphone. While a telephone (and unicast) are not invariably secure, using a single channel approach requires that someone proactively tap the phone line (or intercept packets) in order to hear what is being shared. The unicast functionality is one aspect of the comprehensive, two-tier security framework for the cloud and its participants.

## 5. IMPLEMENTATION AND DESIGN

### 5.1. Components of the Broker

The broker is designed in a modular manner, where each component has a distinct and specific functionality. These components interact with one another, the lookup services, and the rest of the ABELS system on an as-needed basis. With the modular design of the broker, future changes in the functionality may be easily implemented by adding a component without changing the entire structure of the system. As shown in Figure 3, the core implementation of the broker consists of six components: communication module, access control module, load balancing module, data recovery module, resource manager module, and meta lookup service. Figure 3 also shows the Jini lookup services (LUS) used by the broker.

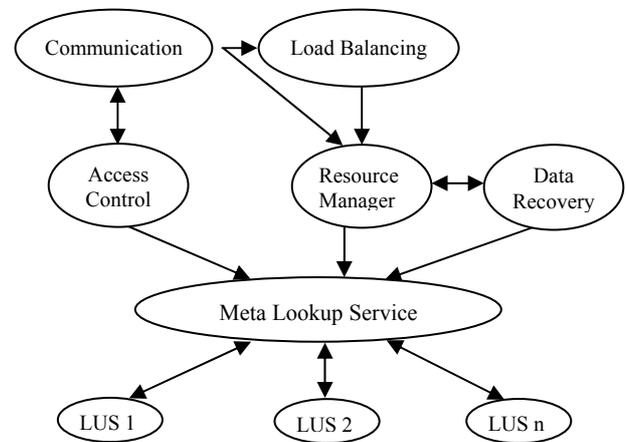


Figure 3: Components of the Broker

The *communication module* of the broker handles all communication coming into or leaving the broker. The communication module provides three main functions:

- Process join and exit requests sent by a user entity's GLA
- Process service registration requests from data producer GLAs
- Process consumer GLA requests for lookup services that support the group(s) of interest.

Upon receiving a join request, the communication module delegates authentication of the connecting GLA to the access control module. Once an entity has successfully joined the cloud, it can send other messages to the communication module. On receiving a message, the communication module verifies the sender's entry ticket to confirm that the sender is a valid cloud participant. If the sender is approved, the communication module examines the tag appended to the entry ticket, since the tag specifies the type of communication contained within the message. Based on the tag, the communication module routes the message to the appropriate broker module and, if necessary, later returns a reply to the sending GLA.

The *access control module* is used to authenticate the identity of a GLA and control access to resources in the cloud. This module is called when an entity initially wants to join the cloud or when the broker's access control list is either referenced or modified by an administrator. As mentioned previously in Section 4.3, there are four security classifications for an ABELS cloud, and the access control module implements the necessary functions of the various security protocols. In addition to the list of IP address restrictions, there is a second list containing the IP addresses of those who are granted administrative privileges for the brokering system. This module is used in two situations:

- When a user entity joins the ABELS cloud for the first time, this module will be called by the broker's communication module to authenticate the entity's GLA, using its digital certificate.
- When a request is made for an administrative modification to a lookup service or group, the module must check its access control list to determine whether the requester is authorized for administrative privileges.

Digital certificates and the secure socket layer (SSL) protocol are used to guarantee secure communication within the brokering system. Once access is approved, the broker issues the entity's GLA an entry ticket that must be presented before any subsequent communication with ABELS components or entities. After receiving an entry ticket, the GLA can communicate directly with the lookup services, for example.

The *load balancing module* is responsible for distributing the registration load among lookup services, and also ensuring that lookup services are distributed evenly across the network. Load balancing is carried out automatically to maintain a robust system that is not bogged down by excessive load concentration. Manual data distribution by a human administrator is also allowed. Automatic load balancing occurs under two scenarios: too many groups are supported by a single lookup service, or too many lookup services are running on a single computer. Currently, a lookup service exceeding the group threshold will have some of its supported groups (and the serviced proxies of the services that support these groups) moved to a lighter-loaded lookup service. To accomplish these modifications, the load balancing module sends the required changes to the resource manager module, which actually moves the groups. In the future, we will consider additional load balancing factors, including the load of the Java Virtual Machine, the number of network connections to any one machine, and the hardware capabilities of the machine.

The *resource manager module* manages the cloud data, keeping track of the lookup services and the groups they support. Thus, the resource manager maintains a list of available resources in the cloud. When a consumer GLA submits a query, the resource manager uses this list to perform a first-level lookup on the query's groups, and it returns to the consumer GLA the lookup service(s) of interest.

The resource manager also performs certain administrative tasks. As appropriate, the module takes care of adding, removing, changing, and editing lookup services. Some of these changes will occur automatically at the request of other broker modules, while others will be performed at the request of the human administrator who oversees the cloud operations.

The actual modification of lookup services requires communication between the resource manager and the *meta lookup service*, which is a lookup service of lookup services. As shown in Figure 3, the meta lookup service provides the link between the broker modules and the various lookup services in the cloud. It is required to replace multicast communication with the safer unicast communication.

As discussed in Section 4.3, by default Jini uses multicast communication to implement the discovery protocol; in this case, there are frequent announcement messages broadcast to the network describing the location and function of the message's sender. By monitoring these broadcasts, the broker could determine the locations of the available lookup services. However, the decision to use unicast communication for security purposes means that no such announcements are made, yet the broker still must know the locations of all of the

lookup services. In fact, unicast means that the broker can communicate only with lookup services whose addresses are known. The solution is to create the meta lookup service at a known location once the broker has been initialized. All lookup services in the cloud must then register themselves with the meta lookup service. Once the lookup services are registered with the meta lookup service, the broker can use the meta lookup service to access the regular lookup services.

Finally, the *data recovery module* is called as soon as a lookup service becomes unavailable. For fault tolerance, at least two copies of each lookup service are maintained, and the duplicate copies are used to recover from a lookup service failure. The required recovery data is first transferred from the backup lookup service to the broker. Next, the resource manager is called to rebuild the crashed lookup service while it concurrently brings the backup online as the primary lookup service. As the data is recovered and a backup lookup service rebuilt, the broker informs the consumer GLAs of the new lookup service. Alternatively, if available system memory is limited, data recovery can occur through the use of maintained log files and Jini lookup service *snapshots* (Li 2000). Snapshots are occasional recordings of the entire system state that are saved to disk. In the case of a system crash, the snapshots are reloaded by the system and the state of the system at the time of the last snapshot is restored. Because only a single set of lookup services is maintained in this method of data recovery, the cloud requires less memory space on the network. However, two main shortcomings exist to this method: the cloud is unavailable while the system reloads the snapshots, and any changes in the system between the last snapshot and the crash are permanently lost. With either method of data recovery, new service registrations and queries may be temporarily blocked to ensure data consistency, but all the services can still be read.

## 5.2 Join Process

The join process is the means by which an entity initially joins the cloud. It can be divided into the following steps:

- An entity's GLA sends a join request to the communication module of the broker.
- The communication module calls the access control module to perform authentication and authorization of the GLA, depending on the cloud's security level.
- The access control module returns the result to the communication module.
- If the registration is accepted, the broker's communication module returns an entry ticket to the entity's GLA. Otherwise, an error message is returned to the GLA.

Once a GLA has received its entry ticket, it is said to have joined the cloud. It is then able to participate as a cloud member by presenting its entry ticket in all subsequent cloud communications.

## 5.3. Lookup Process

Once a GLA has joined the cloud, it can register services (if a producer GLA) or submit queries (if a consumer GLA) after first presenting its entry ticket which verifies it is an approved cloud participant. From the perspective of the broker, the procedures for registering services and submitting queries look very similar. As discussed in Section 4.2, a producer GLA must know the group(s) it wishes to support and a consumer GLA must determine the group(s) it would like to search for potential matches. This determination occurs before the GLA contacts the broker. Once determined, the specified set of groups is sent to the broker in order to receive all lookup services supporting these groups. This process can be divided into the following steps:

- The entity's GLA requests from the broker's communication module, a list of lookup services that support its groups of interest.
- The communication module verifies the entry ticket supplied by the GLA with its request.
- The communication module calls the resource manager module, passing the desired group(s).
- The resource manager gives the list of groups to the meta lookup service to receive the desired lookup services' addresses and ports.
- The broker returns the IP addresses and ports to the entity's GLA.

With these addresses in hand, a producer GLA can register its service with the lookup services returned. A consumer GLA will then perform a first-level lookup with the lookup services to find producers registered in its particular group(s) of interest. It will then require a second-level lookup to determine the best service for its needs. This process, described in Section 4.2, will rank all the services returned to the GLA. Following the second-level lookup, the consumer GLA can connect to the highest-ranked service's GLA to resolve the query.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the design of the secure ABELS brokering system. The brokering system manages the resources in the cloud, storing and maintaining a dynamic set of descriptions and references to the entities currently participating. The brokering system matches data producers to consumers while restricting cloud users to those that are authorized. A flexible security framework has been designed so that the system can be adapted to multiple deployment scenarios. In conjunction with the GLA, the broker provides an effective, two-level security framework for the ABELS system.

Currently, we are continuing the implementation of the design presented in this paper. We are also in the process of developing a graphical user interface (GUI) to aid the cloud administrator in creating and managing the broker. Future work will include the implementation of ABELS-specific lookup services and the development of improved load balancing algorithms. We also plan to add mechanisms to gather and utilize various system statistics, and provide email notifications for user-specified events of interest.

## ACKNOWLEDGMENTS

This work is supported by National Science Foundation KDI Grant 9873138 and U.S. Army Corps of Engineers contract DACA42-01-P-0288.

## REFERENCES

- Curbera, F.; M. Duftler; R. Khalaf; W. Nagy; N. Mukhi; S. Weerawarana. 2002. "Unraveling the Web Services Web: an Introduction to SOAP, WSDL, and UDDI." *IEEE Internet Computing* 6, No.2 (Mar./Apr.), 86-93.
- Dahmann, J.; F. Kuhl; and R. Weatherly. 1998. "Standards for Simulation: As Simple as Possible but Not Simpler, the High Level Architecture for Simulation." *Simulation* 71, No. 6 (Dec.), 378-387.
- Kumar, A.; L. F. Wilson; T. B. Stephens; and J. T. Sucharitaves. 2002. "The ABELS Brokering System". In *Proceedings of the 35<sup>th</sup> Annual Simulation Symposium* (San Diego, CA, April 14-18). IEEE, Picataway, N.J., 63-71.
- Kumaran, S. I. 2002. *Jini Technology: An Overview*. Prentice-Hall, Upper Saddle River, N.J.
- Li, S. 2000. *Professional Jini*. Wrox Press Ltd, Birmingham, U.K.
- Mills-Tettey, G. A. and L. F. Wilson. 2003a. "Security Issues in the ABELS System for Linking Distributed Simulations". *Proceedings of the 36<sup>th</sup> Annual Simulation Symposium* (Orlando, FL, Mar. 30 – Apr. 2). IEEE, Picataway, N.J., 135-144.
- Mills-Tettey, G. A. and L. F. Wilson. 2003b. "A Security Framework for the Agent-Based Environment for Linking Simulations (ABELS)." *Simulation*, to appear.
- Peteet, J. O.; J. P. Murphy; and L. F. Wilson. 2003. "Matchmaking in the ABELS System for Linking Distributed Simulations". *Proceedings of the 15th European Simulation Symposium* (Delft, The Netherlands, Oct. 26 – 29), SCS, San Diego, CA, to appear.
- Wilson, L. F.; D. J. Burroughs; A. Kumar; and J. Sucharitaves. 2001. "A Framework for Linking Distributed Simulations Using Software Agents." *Proceedings of the IEEE* 89, no. 2 (Feb.), 186-200.
- Wilson, L. F.; B. Xie; J. M. Kimpel; G. A. Mills-Tettey; and G. Johnston. 2002. "The Design of the Distributed ABELS Brokering System". *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT)* (Fort Worth, TX, Oct. 11-13). IEEE, Picataway, N.J., 151-158.

## AUTHOR BIOGRAPHIES

**LINDA F. WILSON** is an associate professor at Dartmouth's Thayer School of Engineering. She received her BS degree in mathematics from Duke University in 1988 and her MSE and PhD degrees in electrical and computer engineering from the University of Texas at Austin in 1990 and 1994, respectively. Her email address is [Linda.F.Wilson@dartmouth.edu](mailto:Linda.F.Wilson@dartmouth.edu) and her web page can be found at <http://thayer.dartmouth.edu/~lwilson>.

**W. RILEY LOCHRIDGE** is a master's student at Dartmouth's Thayer School of Engineering. He received his AB degree in history and engineering modified with computer science from Dartmouth College in 2002. His email address is [Riley.Lochridge@dartmouth.edu](mailto:Riley.Lochridge@dartmouth.edu).

**G. AYORKOR MILLS-TETTEY** is currently working in her native country of Ghana. She received her AB degree in computer science in 2001 and her BE and MS degrees in engineering in 2003, all from Dartmouth College. Her email address is [Ayorkor.Mills-Tettey@alum.dartmouth.org](mailto:Ayorkor.Mills-Tettey@alum.dartmouth.org).

# AN ANALYSIS OF INTERNAL/EXTERNAL EVENT ORDERING STRATEGIES FOR COTS DISTRIBUTED SIMULATION

Simon J. E. Taylor  
Navonil Mustafee  
Centre for Applied Simulation Modelling  
Department of Information Systems and Computing  
Brunel University  
UB8 3PH, Uxbridge, England  
E-mail: simon.taylor@brunel.ac.uk

## KEYWORDS

COTS Simulation Packages, Distributed Simulation, Discrete Event Simulation.

## ABSTRACT

Distributed simulation is a technique that is used to link together several models so that they can work together (or interoperate) as a single model. The High Level Architecture (HLA) (IEEE 1516.2000) is the *de facto* standard that defines the technology for this interoperation. The creation of a distributed simulation of models developed in COTS Simulation Packages (CSPs) is of interest. The motivation is to attempt to reduce lead times of simulation projects by reusing models that have already been developed. This paper discusses one of the issues involved in distributed simulation with CSPs. This is the issue of synchronising data sent between models with the simulation of a model by a CSP, the so-called *external/internal event ordering* problem. The motivation is that the particular algorithm employed can represent a significant overhead on performance.

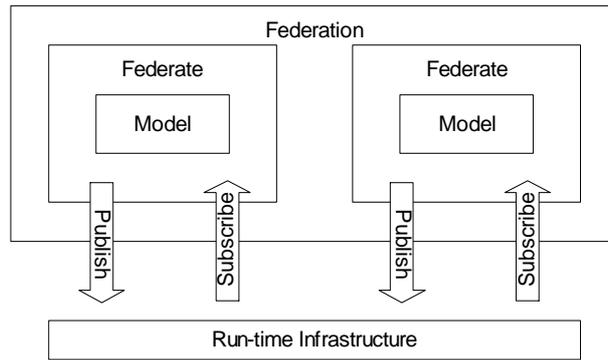
## INTRODUCTION

Distributed simulation is a technique that is used to link together several models so that they can work together (or interoperate) as a single model. The High Level Architecture (HLA) (IEEE 1516.2000) is the *de facto* standard that defines the technology for this interoperation. Models, or federates, interoperate together to form a federation. Interoperation takes the form of organised communication of data specified in a Federate Object Model, using tables derived from the Object Model Template (IEEE 2000b), via supporting communication technology called a Runtime Infrastructure (RTI) (as defined by IEEE 2000a). Currently the HLA is most widely used in real-time simulation of defence related training problems.

In many areas of industry, COTS Simulation Packages (CSPs) are used to model systems in diverse domains such as commerce, defence, health and manufacturing. A CSP is a generic term that refers to a computer simulation package that is a visual interactive modelling environment that helps simulationists to build models,

perform experiments, visualise and report during simulation projects. Although not exclusively, they are typically based on some variant of the discrete event simulation paradigm, i.e. models change state at discrete points in time by scheduled or conditional events and typically represent entities or objects (documents, patients, parts, trains, etc.) in some form that pass through networks of queues and workstations (work queuing at a desk in an office, patients waiting to see a doctor, parts buffered for machining, trains waiting at a station, etc.) Generally, each package has a range of basic model elements (queue, workstation, resource, source, sink, etc.) and advanced model element (conveyor, shift worker, warehouse, etc.) that are used to build a model via a drag and drop visual interface. Each model element can be modified as is required, either by a menu system or by a package programming language, to better represent the system being studied (for example the queuing logic of a queue or the behaviour of a resource). Entities or objects can be represented and differentiated by attributes. Terminology between packages differs as there is no generally recognized naming convention.

The creation of a distributed simulation of models developed in CSPs is of interest. The motivation is to attempt to reduce lead times of simulation projects by reusing models that have already been developed. For example Boer, et al. (2002) discuss the use of distributed simulation to simulate container handling at a port, while Sudra, et al. (2000) and Taylor, et al. (2002) discuss how distributed simulation can facilitate the modelling of supply chains and problems in the automotive industry. A factor that distinguishes this work from other research in distributed simulation is that interoperation must not only take place between models but also the CSPs in which the models reside. This paper discusses one of the issues involved in distributed simulation with CSPs. This is the issue of synchronising data sent between models with the simulation of a model by a CSP, the so-called *external/internal event ordering* problem. This is of interest as the particular algorithm employed can represent a significant overhead on performance. The paper is structured as follows in section 2 we describe the external/internal event ordering problem in more



**Figure 1: A Distributed Simulation Federation**

detail. Section 3 introduces four algorithms that can be used for this problem. Section 4 presents some results. Section 5 ends the paper with some conclusions and further work.

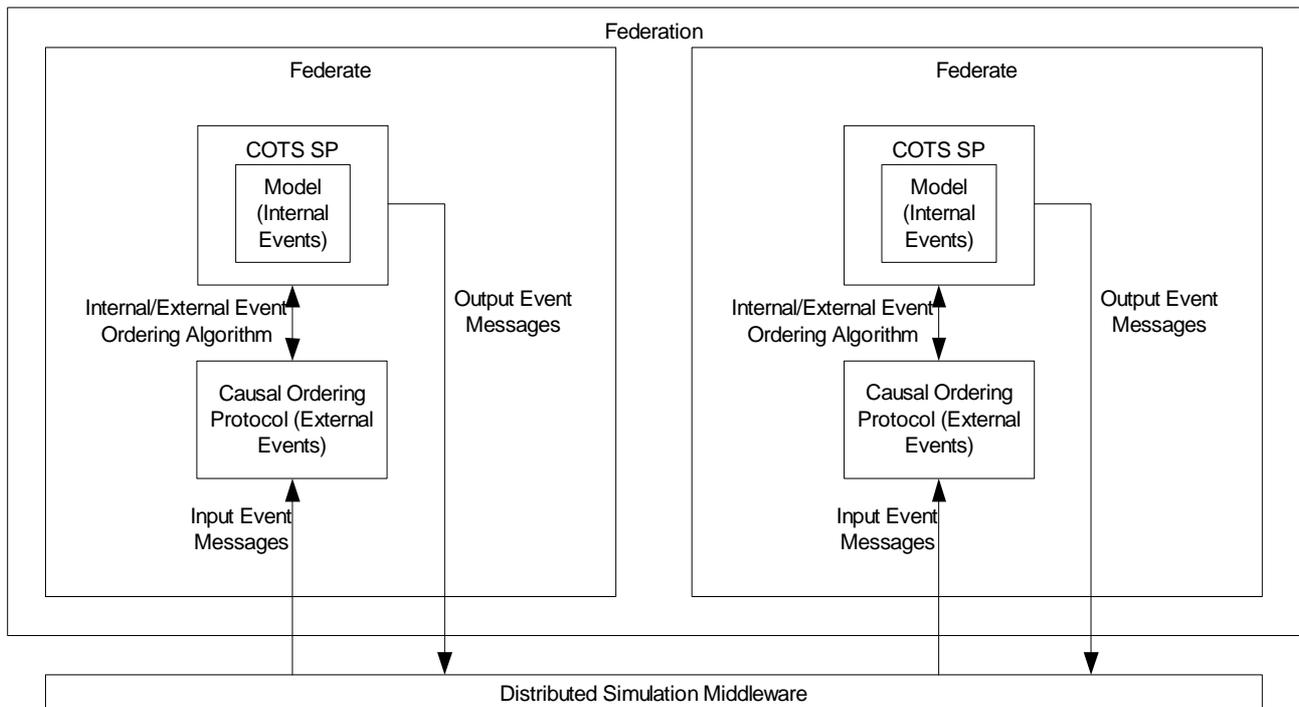
**THE EXTERNAL/INTERNAL EVENT ORDERING PROBLEM**

The general problem of external/internal event ordering can be described as follows. Generally speaking, in a federates exchange information to perform a simulation of a particular system. Initially this is done on the basis of publication of information of interest and subscription to information of interest (publish-subscribe). A run-time infrastructure performs the actual exchange of information. Each federate performs the simulation of a particular model. The information

exchanged between federates is therefore dependent on the models being simulated. Figure 1 shows a general distributed simulation federate.

In distributed simulation with CSPs, a federate contains a CSP which in turn contains a model. Data is generally on the basis of timestamped event messages. Event messages arriving at a federate from another are notionally organised by some causal ordering protocol and are introduced to the CSP and then the model being simulated by the CSP in timestamp order. These *external* events are ordered with the CSP/model's *internal* events according to some algorithm. Figure 2 shows these relationships.

To investigate the implications of this problem in a CSP



**Figure 2: Distributed Discrete Event Simulation**

federate, consider the following. A CSP typically possesses a simulation executive, an event list, a clock, a model state and a number of event routines (this is a gross simplification as these packages have many variants of this). The model state and the event routines represent the state of the model at a particular time and the logic by which it changes and are derived from the model that is implemented in the package (and therefore represent the model). Initialising the simulation, events are placed on the event list (typically modelling entities arriving in the model, e.g. raw materials arriving in a factory). If we assume that the simulation executive uses some form of the three phase approach (TPA), the simulation first advances clock time to the time of the next event (the A Phase) and then executes that event (the B Phase) according to its event routine. This may result in a change in the simulation state, the scheduling of new events on the event list and the sending of new timestamped event messages. The simulation executive then determines if the changed state has enabled any conditional events (the C Phase). If any have been enabled, these events are executed in some priority order and again may result in a change in the simulation state, the scheduling of new events on the event list and the sending of new timestamped event messages. The simulation executive then makes a new cycle of the three phases. Algorithm 1 describes this (note that we assume that *Update event list*, *Update simulation state*, and *Send event messages* are conditional on the results of the B/C Phase).

```

while not terminated do
    Advance to time of next event (A Phase)
    Execute event (B Phase)
        Update event list
        Update simulation state
        Send event messages
    Test conditional events (C Phase)
        Update event list
        Update simulation state
        Send event messages
endwhile

```

#### Algorithm 1 The Three Phase Approach

The problem of external/internal event ordering is therefore as follows. If a federate consisting of a CSP and its model has an event list that contains internal events, and a causality ordering protocol has ordered the external events messages arriving from other CSPs, how can the simulation executive of the CSP determine the next event to process? Is the next event an internal one taken from the event list or an external one represented by the timestamped event message offered by the protocol? In the next section we review several algorithms that could be used to perform this ordering.

### EXTERNAL/INTERNAL EVENT ORDERING ALGORITHMS

There are several possible algorithms that can be used to order external events with internal events. Each algorithm is defined on the basis of a relationship between a modified form of the CSP TPA and an external body that orders external events via a causality ordering protocol (the *external event manager* EEM). Each of these with their assumptions are now discussed.

#### Event List Externalisation

A simple solution to this is to remove the event list from the CSP and treat all events as external events. Events scheduled within the simulation package are externalised and ordered with the external events, i.e. any *internal* event becomes an event message. *Get next external event* represents the action of taking the next external event that has been identified by the EEM and introducing it to the CSP. Algorithm 2 describes this.

```

while not terminated do
    Get next external event
    Advance to time of next event (A Phase)
    Execute event (B Phase)
        Update simulation state
        Send event messages
    Test conditional events (C Phase)
        Update simulation state
        Send event messages
endwhile

```

#### Algorithm 2: Event List Externalisation

#### Permission request

In this approach, the CSP's TPA is modified to request permission from the EEM. Prior to the A phase, time advance, the modified form of the TPA asks permission from the EEM to advance to the time of the next event on its event list by performing *Request (permission (Next\_Event\_Time))*. This sends the time of the next event *Next\_Event\_Time* to the EEM. The TPA would then wait until the EEM responds with a *Reply Message* where *Message* can be *advance(Time)*, *event(Time)* or *wait*. The actions dictated by the reply from the EEM are either (a) to grant permission to advance to a given time *Time* by message *advance(Time)*, (b) to pass a timestamped external event *event* with timestamp *Time* by message *event(Time)*, or (c) to request the simulation executive to wait by message *wait*. In the case of (a), the timestamp of the next external event is greater than the scheduled time of the next (internal) event; the TPA would therefore execute phase A by advancing to the time of the next event and then perform phases B and C as normal before making a new cycle of the modified TPA. If the timestamp of the next external event is less than the scheduled time of the next (internal) event (b), the external event would be passed to the simulation executive. The TPA would then carry on by executing phase A, i.e. advancing to the time of the newly scheduled event. Phases B and C would be executed as

normal. If the EEM could not determine the earliest safe timestamped message (as is possible with causal ordering protocols), when the TPA next asked permission it would be requested to wait (as in case (c)). The TPA would then be suspended until the EEM indicated a change of circumstances. Algorithm 3 describes this.

```

while not terminated do
  Request (permission(Next_Event_Time))
  if Reply (advance(Time)) then
    Advance to time of next event
      (A Phase)
    Execute event (B Phase)
      Update event list
      Update simulation state
      Send event messages
    Test conditional events (C Phase)
      Update event list
      Update simulation state
      Send event messages
  else if Reply (event(Time)) then
    Advance to Time (Modified A Phase)
    Execute event (B Phase)
      Update event list
      Update simulation state
      Send event messages
    Test conditional events (C Phase)
      Update event list
      Update simulation state
      Send event messages
  else if Reply (wait)
    wait until notified
  endif
endwhile

```

### Algorithm 3 Permission Request

#### Incremental advance

Rather than controlling the advancement of time in the CSP through the TPA, this algorithm assumes that it is not possible to obtain access to the “next event time.” Here we must advance time by the smallest possible time unit of the CSP. Before each time advance the TPA performs *Request permission(Time\_Increment)*. This sends the time of the next time increment *Time\_Increment* to the EEM. The TPA must then wait until the EEM responds with a *Reply Message*, where *Message* can be *granted*, *event* or *wait*. The actions dictated by the reply from the EEM are either (a) to grant permission to advance by a single time increment by message *granted*, (b) to pass a timestamped external event *event* and to grant the time increment advance by message *event*, or (c) to request the simulation executive to wait by message *wait*.

The consequence of these messages are that if the EEM is aware of the next safe external event, and the timestamp of this greater than the next incremented

time, the CSP is allowed to make another incremented advance (a). If the timestamp of the next external event is equal to the next incremented time, the external event will be introduced for execution at the next incremented time and the TPA is allowed to make another incremented advance (b). Finally, if the EEM cannot identify the next safe external event the incremental time advance will be halted until a new message arrives (c).

```

while not terminated do
  Request (permission(Time_Increment))
  if Reply (granted) then
    Advance by Time Increment
      (A Phase)
    if next event then
      Execute event (B Phase)
      Update event list
      Update simulation state
      Send event messages
      Test conditional events
        (C Phase)
      Update event list
      Update simulation state
      Send event messages
    endif
  else if Reply (event) then
    Execute event (B Phase)
      Update event list
      Update simulation state
      Send event messages
    Test conditional events (C Phase)
      Update event list
      Update simulation state
      Send event messages
  else if Reply (wait)
    wait until notified
  endif
endwhile

```

### Algorithm 4 Incremental Advance

#### External control

An alternative to making the TPA request permission is to effectively make the CSP a slave of the EEM. The EEM first determines the course of action and then externally controls the behaviour of the CSP’s time advancement. Depending on the status of the external events, the EEM may make the CSP wait on *Wait (instruction)*. The possible values of *instruction* are *advance(Time)* and *event(Time)*. When the value of *instruction* is instantiated, the modified TPA may, if *instruction* equals *advance(Time)* execute as normal until *Next\_Event\_Time* is greater than *Time* (a), or if *instruction* is *event(Time)* execute as normal until *Next\_Event\_Time* is greater than *Time* and then execute the new event *event* (b). In the case of (a) the EEM has determined that it is safe for the CSP to advance to a given time. The CSP cycles through the TPA,

advancing time until this “safe” time. If the EEM has identified a new safe external event, it orders the CSP to advance until the timestamp of the event message and then introduces the new (external) event to the CSP to be processed (as in (b)). If neither is the case then the CSP waits until an instruction is sent by the EEM, i.e. the EEM cannot identify a safe course of action.

```

while not terminated do
  Wait (instruction)
  if instruction = advance(Time) then
    while Next_Event_Time < Time do
      Advance to time of next event
        (A Phase)
      Execute event (B Phase)
      Update event list
      Update simulation state
      Send event messages
      Test conditional events (C Phase)
      Update event list
      Update simulation state
      Send event messages
    endwhile
  else if instruction = event(Time) then
    insert event(Time) in event list
    while Next_Event_Time <=
      event(Time)
      Advance to time of next event
        (A Phase)
      Execute event (B Phase)
      Update event list
      Update simulation state
      Send event messages
      Test conditional events (C Phase)
      Update event list
      Update simulation state
      Send event messages
    endwhile
  endif
endwhile

```

### Algorithm 5 External Control

#### Summary

This section has introduced four algorithms to solve the external/internal event ordering problem. The next section reports on some results obtained from experimentation with programs based on the algorithms.

#### EXPERIMENTS

As the purpose of the experiments are to investigate the external/internal event ordering problem, not external event ordering, we shall assume that there is no wait time, i.e. all external events have been produced and the

EEM has ordered them. This is an artificial but valid assumption as the results of these experiments will give us a base line on performance which will degrade under conditions where algorithms are forced into their wait state caused by the EEM being under populated with event messages (or other). Timestamps of internal and external events were arbitrarily selected to suit the experiments and are deterministic. Experiments were performed on the basis of event density,  $D$ , which is the ratio of the number of external events  $X$  to the number of internal events  $I$ , i.e. defined as  $D = X/I$ . The number of internal events were held constant at 1000. The processing time for an event executed by the artificial CSP was held at 5ms. The data points on the graph are for an average based on 10 runs. The data points on the graph are for values of  $D$  against average time to process 1000 internal event messages. 0.001, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 3, 4, 5. The program was implemented as a client-server system in Java under Microsoft Windows 2000 using sockets. The computer was an Intel Pentium III processor 744 MhZ with 256Mb RAM running Windows 2000. Figure 3 shows these results. Note that the results for the incremental advance algorithm are excluded as they are a magnitude greater than results for the other algorithms.

#### CONCLUSIONS

Of our four algorithms, *external control* appears to be the “winner.” *Event list externalisation* and *permission request* give similar results, while *incremental advance* gives a magnitude worse performance. This is unsurprising as *external control* allows the CSP to proceed with the least interaction. However, the selection of the “best” solution to our problem cannot be made just on performance. The problem faced by interoperating CSPs is that many of the features that are required for the ordering of external and internal events are sometimes not obtainable. For example, some CSPs have COM controls that make all data structures (including the event list) easily accessible. Others have little in the way of accessibility – even the time of the next event is hidden. For example, even though *external control* may appear to be the best ordering algorithm, only *incremental advance* may be possible as there is no method of advancing the simulation clock to a given timestamp. Further work will investigate this problem of compatibility.

It is hoped that the work presented in this paper will stimulate other external/internal event ordering algorithms. This ordering represents a major performance overhead and attempts to reduce this overhead can only make distributed simulation a more attractive possibility.

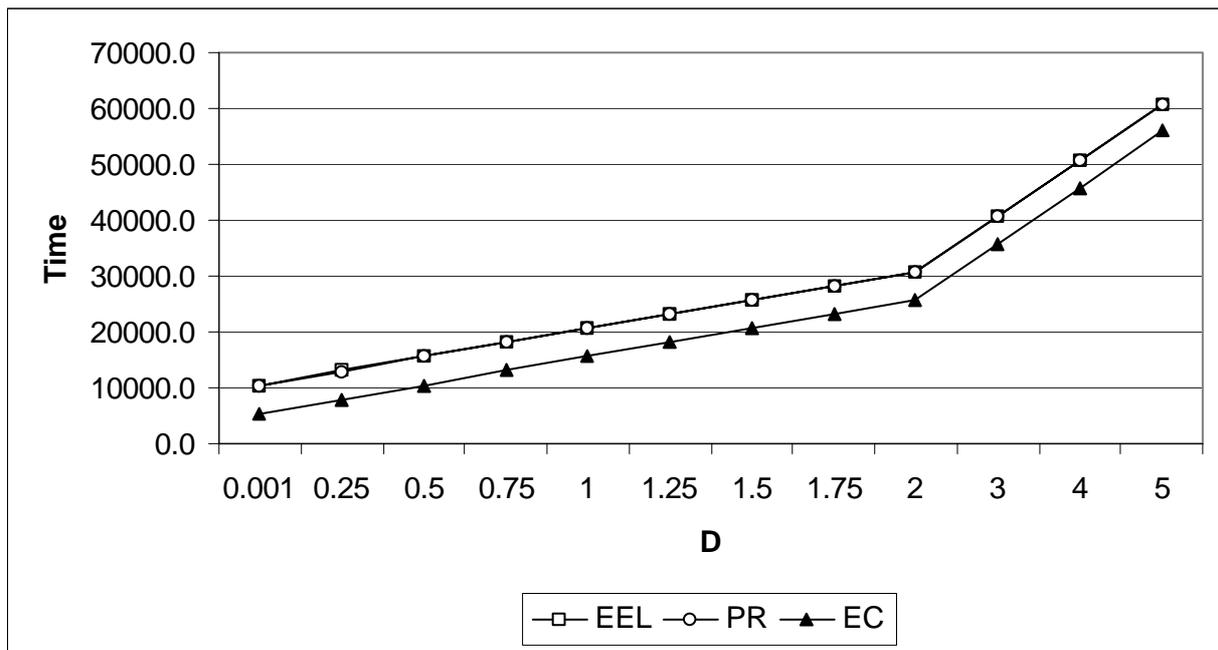


Figure 3: Comparison of External/Internal Event Ordering Strategies

## REFERENCES

- Boer, C.A., A. Verbraeck and H.P.M. Veeke. 2002. Distributed Simulation of Complex Systems: Application in Container Handling. In Proceedings of SISO European Simulation Interoperability Workshop. Simulation Interoperability Standards Organisation, Orlando, Florida.
- IEEE 2000a. IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification. IEEE Std 1516.1-2000. IEEE Computer Society, New York, NY.
- IEEE 2000b. IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification. IEEE Std 1516.2-2000. IEEE Computer Society, New York, NY.
- Sudra R., S.J.E Taylor and T. Janahan. 2000. Distributed Supply Chain Management in GRIDS. In Proceedings of the 2000 Winter Simulation Conference. 356-361. Association for Computing Machinery Press, New York, NY.
- Taylor, S.J.E., R. Sudra, T. Janahan, G. Tan and J. Ladbrook. 2001. Towards COTS Distributed Simulation Using GRIDS. In Proceedings of the 2001 Winter Simulation Conference. 1372-1379. Association for Computing Machinery Press, New York, NY.
- Taylor, S.J.E., A. Bruzzone, R. Fujimoto, B.P. Gan, S. Strassburger and R.J. Paul. 2002a. Distributed Simulation and Industry: Potentials and Pitfalls. In Proceedings of the 2002 Winter Simulation Conference, San Diego, CA. 688-694. Association for Computing Machinery Press, New York, NY.
- Taylor, S.J.E., R. Sudra, T. Janahan, G. Tan and J. Ladbrook 2002b. GRIDS-SCS: An Infrastructure for

Distributed Supply Chain Simulation. SIMULATION. 78(5): 312-320.

# A TOP-DOWN APPROACH TO MODEL INTEROPERATION PROVISION IN COTS SIMULATION PACKAGES

Michael D. Ryde

Simon J. E. Taylor

Centre for Applied Simulation Modelling

Department of Information Systems and Computing

Brunel University, Uxbridge, UB8 3PH. UNITED KINGDOM

## ABSTRACT

This paper examines current methods for model interoperation when using COTS (Commercial Off-The-Shelf) simulation packages. The viewpoint taken for this work is from that of the simulation engineer. By applying distributed simulation theory an attempt is made to suggest how an example COTS simulation package could be modified to provide the necessary functions and interoperability required. Further, by studying current methods employed, which enable COTS simulation packages to interoperate, this paper will discuss the tools currently used, examine their appropriateness and suggest further areas of research.

## INTRODUCTION

The interoperation of simulation models through distributed simulation has provided many areas for academic research. Much of this research has been focused on the technological challenges faced by software engineers who have strived to determine the most efficient and accurate way of enabling simulation models to communicate. The technological problems have included (amongst others) Web and Network based simulations, see Miller et al. 2001, model and object reuse, see Pidd 2002, model synchronization, see Fujimoto 1990 and 1999 and the technical implications of using distributed simulation in a specific application area, for two examples from many, see Zeigler et al. 1999, Carothers et al. 1994.

A major contribution made by the distributed simulation field of research is the High Level Architecture (HLA). The standard (IEEE 1516) provides a framework for distributed simulation. Each model, or federate, interacts with each other (interoperates) to accomplish the simulation exercise and the combined set of interoperating federates is referred to as a federation. The HLA gives standards for data representation (needed so that the communicating federates can “talk” the same language – the format of data exchanged between models) and middleware (to allow communicating parties to “talk” – this is the federate interface specification, the implementation of which is called a run time infrastructure, RTI).

Distributed simulation enabled by the HLA has been used extensively in military systems (see previous

Winter Simulation Conferences and SISO’s Simulation Interoperability Workshops for many examples). There have been relatively few examples of this in industry. This is not for the lack of opportunity. See Strassburger 2001 for an in-depth discussion on how the HLA could be used outside of the defence arena. Another use of the HLA outside the defence arena was put forward as part of the Intelligent Manufacturing Systems (IMS) mission project. See McLean and Riddick 2000. Interestingly, an observation made during this research was that the current RTIs (developed by different sources) did not interoperate with each other thus all models in a distributed simulation would need to use the same RTI.

Another RTI based development includes GRIDS, which provides a generic run-time infrastructure for the execution of distributed simulations. GRIDS provides basic simulation services to connect simulation models (federates) cooperating to perform a distributed simulation (federation), and extensible simulation services to provide performance enhancement, time-management, mobile entities, as required. Sudra et al. 2000.

This paper however, is primarily concerned with Commercial-Off-The-Shelf simulation applications and takes a top-down approach to the same issues mentioned above. We attempt to show how the nature of a COTS Simulation package can be changed through the provision of interoperational features. This approach is taken with the firm belief that simulationists will use the tools at their disposal and currently, in many COTS simulation applications, model interoperation functionality either does not exist or requires software development knowledge to use. Arguably, this could be given as a significant reason for the low up-take and use of distributed simulation in commercial sectors.

This paper is divided into 5 sections. Following the introduction in section 1, Section 2 is concerned with the current definition of the COTS simulation package and highlights the typical attributes found in these packages. Section 3 examines how interoperation may be achieved using current packages and suggests potential pitfalls. Section 4 suggests possible enhancements that could be adopted by the software development houses to enable model interoperation. Finally, section 5 concludes and suggests further areas for research in this field.

## COTS SIMULATION PACKAGES

A typical COTS simulation package, for the purposes of this paper, is considered an application in which simulation models can be constructed, saved and reused. The model would normally be constructed from objects, some of which would be standardized between models. Further, it is also expected that the package would have some form of representation for entities (items of work) that would be used within the model. Typically, these packages would include definitions for entity distributions and methods by which various objects within the model could be linked or ordered. COTS simulation packages can be, and often are, used by various sized organizations but are easily accessible to even the smallest of businesses because of their low cost. Thus the diversity of model that the packages are expected to deal with is fairly broad.

Many COTS simulation packages have a Visual Interactive Modeling (VIM) interface, use event lists and have defined entities. In addition, these packages are accessible to many organizations due to their costing structures and as with many 'volume' packages are available on the Microsoft Windows platform. VIM interfaces provide a method of control to functions available to the user. Also, these types of interface often support a 'drag and drop' style of interaction making simulation model building a rapid process.

A brief review carried out during March 2003 revealed the following (although not exhaustive) list of COTS simulation applications:

1. ARENA (Rockwell Software)
2. AUTOMOD (Brooks Automation AutoSimulations Division)
3. Awe Sim (Frontstep, Inc.)
4. EXTEND (Imagine That, Inc.)
5. GPSS for Windows (Minuteman Software)
6. GPSS/H/Proof Animation/SLX (Wolverine Software Corporation)
7. iGraphx Process 2000 (Micrografx, Inc.)
8. microGPSS/webGPSS (Ingolf Stahl)
9. ProModel (Production Modelling Corporation)
10. QUEST (DELMIA Corporation)
11. SIGMA (Custom Simulation)
12. SIMPROCESS/SIMSCRIPT II.5 (CACI Products Company)
13. SIMUL8 (SIMUL8 Corporation)
14. Taylor Enterprise Dynamics (F & H Simulations)
15. Visual Simulation Environment (Orca Computer, Inc.)
16. WITNESS (Lanner Group, Inc.)

## THE INTEROPERATION OF COTS SIMULATION PACKAGES

Currently, there are no known products that have the ability to support and natively allow, multiple models to interoperate without at-least the use of some basic middleware component. See figure 1. However, there are methods used to emulate the interoperation of models.

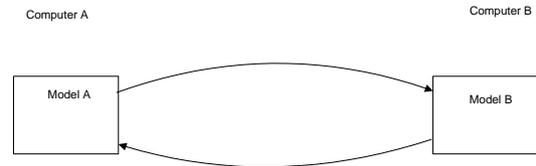


Figure 1: Native Model Interoperation

Usually simulation models require, as a minimum, input in the form of a distribution of entities. The entity distribution for a model could be taken from existing models by executing a number of experimental runs to determine the required spread and frequency. This information could then be passed directly into a model via a spreadsheet, see figure 2. Many COTS simulation packages provide functionality to write out to and read variables from a spreadsheet package in order to provide a way of passing information between models. In many cases this provides little more than the passing of information sequentially from one model to another.

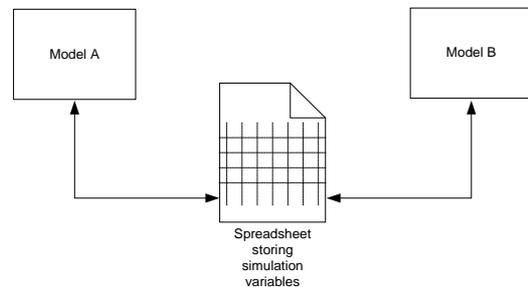


Figure 2: Simple Model Interoperation using a Spreadsheet package

To apply the same method to many models passing information (entities) to one another, one must consider the synchronization if causality issues are to be avoided. It is likely that if multiple models were running and passing information to each other, then these models could be running at different speeds; i.e. the simulation clocks could be different. Thus in figure 3, Model A, when receiving an event from Model B and Model C, would need to determine which event to process first. Using a spreadsheet package to facilitate the passing of entities may provide some limited mechanism for reading/writing time-stamped information, event list information and even synchronization logic (time-management). However it is suggested that a spreadsheet, using basic functions would be grossly

inadequate and such a mechanism would require some further middleware logic (program instructions) to give the required functionality. It can then be argued that the spreadsheet package is no longer acting as a simple data passing mechanism, more as a fully-fledged time-management component. Is a spreadsheet package really the best tool for the job in this case?

It has long been suggested that the distribution and interoperation of simulation models can be achieved through the use of a 'Spreadsheet' some evidence of this can be found in Clarke 1993. This we term as the '*Spreadsheet Approach*', which, it is postulated, is inappropriate for all but the simplest interoperations.

As suggested earlier, it can be seen that using this method for distributed simulation cannot work without some layer of intermediate code to deal with the time-management functionality. It could therefore be assumed that programming skills would also be required by the simulation engineer in order to create this middleware.

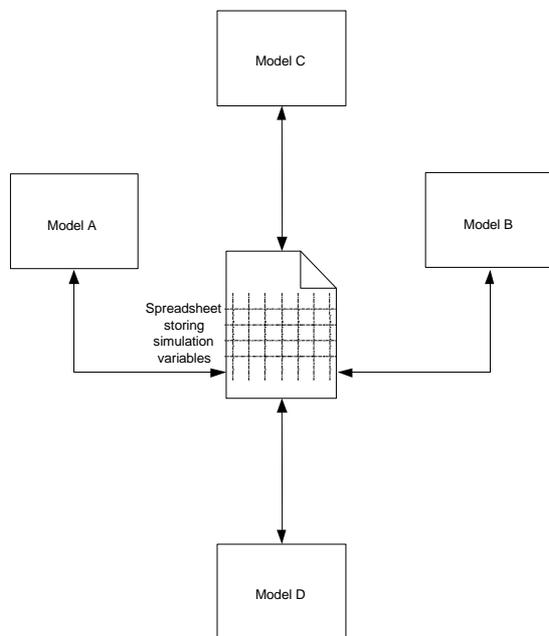


Figure 3: Complex Model Interoperation

The following is an attempt to build an initial set of requirements for the interoperation of simulation models using COTS packages:

- To be able to link objects in different models and use their output as entity distributions or actual 'parcels' of data where required. This provides the core functionality for the linking of models by connecting defined simulation objects, which generate simulation events and output entities.
- The facility to pass entity data between objects in disparate models. This will define the mechanisms to pass actual simulation data between the models

and would involve the standardisation of entities throughout the entire distributed simulation model.

- The provision of access to control the starting and stopping of a model externally. This is essential so that models can be started and terminated in a synchronized manner.
- The implementation of time-management algorithms for model synchronization. Possibly the provisions of different synchronization algorithms could be provided so that the appropriate type of synchronization could be used for particular simulation problems.
- The ability to interrogate the event list in order to examine the next event before it is executed. This is required to facilitate the implementation of time management functionality, specifically look-ahead as used in the Conservative synchronization algorithms.
- Separate control for re-running C-Phase of operation as specified in the three-phase simulation methodology, see Brooks and Robinson 2001. Again, this would be required specifically for synchronisation algorithms.

### SUGGESTED ENHANCEMENTS TO A COTS SIMULATION PACKAGE

It is believed that due to the way many COTS Simulation packages are designed adding interoperability could be relatively straightforward. For the purposes of this paper we restrict ourselves to one package, SIMUL8 (SIMUL8 Corporation). This package has a VIM interface and uses event lists with defined entities. SIMUL8 is an accessible package for many organizations due to its costing structure and is available on the Microsoft Windows platform. The VIM provides a high level of control to many of the technical features and functions available to the simulation engineer and the package is believed to be an appropriate candidate for our suggested enhancements. An attempt has been made to suggest new or modified functions and even a possible user interface, using SIMUL8 as an example.

We have also decided for the purposes of this case study not to address heterogeneous COTS simulation package interoperability.

### Functions

Table 1 gives examples of functions that could be made available in COTS simulation packages such as SIMUL8. The authors of this paper have no knowledge of the internal mechanisms or software design that SIMUL8 uses and so these functions serve merely as general software design suggestions.

At the current time the main body of work has focused on run control and entity exchange. The functions suggested would allow a model to use external objects and variables and also enable the model to share its

own objects and variables. Further, the distribution of an ‘input’ could be defined as an external function, providing an alternate method of distribution. A final function is provided to enable a selected model to become ‘the master’ for ease of control and synchronization of the ‘Global Model’ or ‘Federation’.

The functions in Table 1 serve merely as example functions, which could exist within an API (Application Program Interface), but are not intended to represent a complete list. However, they do serve to highlight some important mechanisms, which are required to provide external control and entity exchange within the COTS simulation package.

Table 1: Run control and entity exchange functions

Function	Description
Handle ExternObj(Object)	Externalisation of objects for external access. Returns handle to object.
Handle Extern(Variable)	Externalisation of variables for external access. Returns handle to variable.
SetMaster(Boolean)	Set Master Model - Allows a specific model to be set as a master to stop and start the entire simulation.
Entity GetExternDist(Model, FromObject, ToObject)	Get external distribution - Modify existing routine to interrogate objects within separate SIMUL8 models for distribution patterns. Returns Entity.
Boolean LinkExternal(Model A, Object, Model B, Object)	Links object in model A to an external object in model B. Returns True if successful.

SIMUL8 supports the notion of Plugins, which enable specific software modules to be integrated in to the package. A possible use for this could be for time-management algorithms. This could allow different synchronization protocols to be used when models have been distributed. The Plugins could include Conservative (lookahead, lookback and null message protocols) and Optimistic (Time Warp) algorithms. The integration detail is expected to be more complex for these software components; however, the mechanism could provide a neat and elegant solution to the problem.

## SIMUL8 Application Programming Interfaces

Although strictly not relevant to the simulation engineers (due to the requirement of software development skill), the application programming interfaces (API’s) provide the first steps towards interoperability. Once the necessary native functions have been introduced to the application, it is not unreasonable to expect separate organisations and even users with software development experience, to develop standardized middleware to be used by general simulation engineers (to allow model interoperability). Currently SIMUL8 supports API’s at a number of different levels, i.e. OLE Automation, COM and through the ActiveX interface. There are also some direct linking facilities, using the user interface, which can enable the user to link to Microsoft Excel or Visual Basic (although these probably use the facilities provided in the API).

## SIMUL8 Interface Suggestions

Modifications to the SIMUL8 interface will be required to enable the Simulation Engineer to design interoperating models. Below are suggested interface enhancements to provide access to the interoperability functionality, primarily focusing on model selection, object linking and setting the master control.

## Selecting External Models

The current object linking box in SIMUL8 version 9 provides a mechanism to link various objects within the same simulation, see figure 4. Figure 5 suggests a modification to this dialog box to allow links to be made to external objects by first selecting the model in which the object resides.

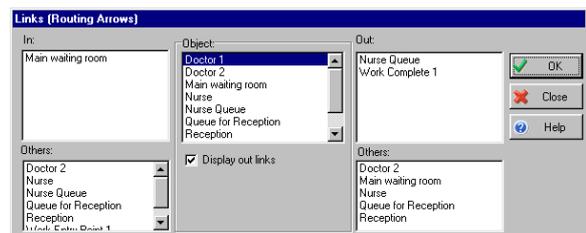


Figure 4: Current Object Linking Dialog

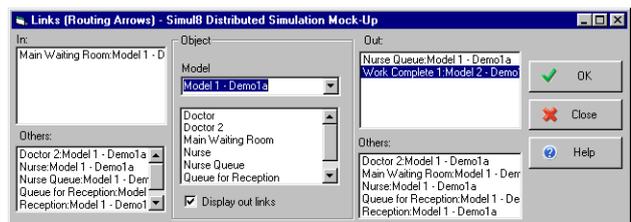


Figure 5: Modified Object Linking Dialog

## Linking external objects

Once a model has been selected, external objects could then be used for specific distributions. Alternatively, an externalised variable from the model such as a published 'results' variable could be used to provide the input. Figure 6 shows an example of the dialog boxes to enable external distribution selection.

The main purpose of creating external distributions is to replace the commonly used stochastic distributions and provide 'real' input in the form of entity occurrences (as opposed to a statistically derived distribution). The input captured from interoperating models could then be used to define, after a number of experimentations, a distribution, which could be used within a single model. Further implementation could be considered to integrate the process with the 'optimisers', which are often provided in COTS simulation packages. This could provide a mechanism by which experiments could be automated from which a set of distributions could be derived from interoperating models.

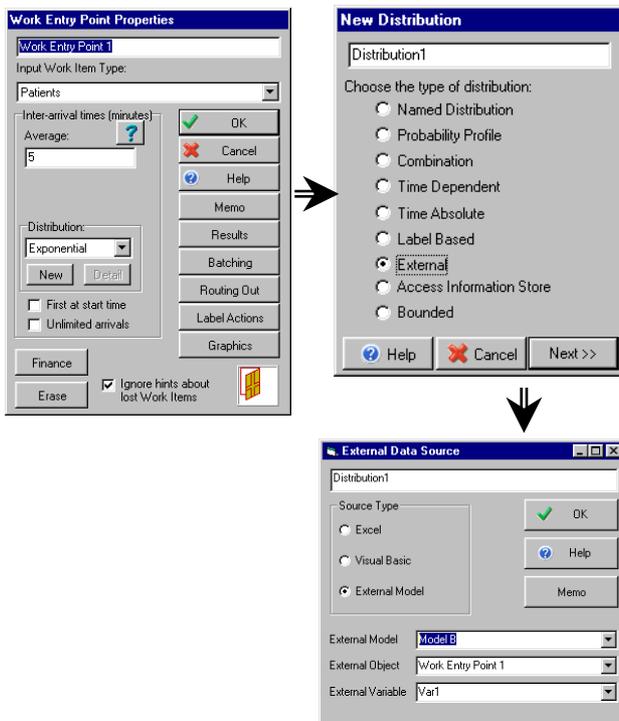


Figure 6: Modified External Distribution Dialog

## Setting Model to be the Master

The modified user interface shown in figure 7 reveals an additional menu option to set the current model to be the master controller for all linked models. This functionality could provide 'central' control for all interoperating models, such as synchronized start and stop.

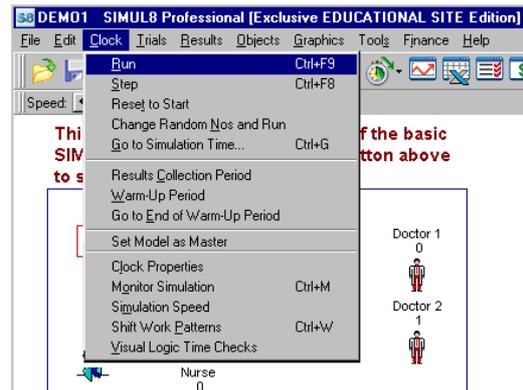


Figure 7: Modified Clock Menu

## CONCLUSIONS AND SUGGESTED AREAS FOR FURTHER RESEARCH

COTS simulation packages tend to be designed for use on a single disparate simulation model. This paper has pointed to some areas, which could be addressed to provide further integration of interoperability (i.e. distributed simulation) functionality.

The two key areas discussed are functionality and interface design. Of-course many packages include functionality for the reading and writing of variable data to a specified application, such as a spreadsheet package. Some packages like SIMUL8 also include the facility to pass data directly into a program developed by the simulation engineer. Functionality is being addressed already, in packages like SIMUL8, where an API is currently in development to facilitate model interoperation. However, the level of integration required to enable any reasonable amount of distributed theory integration, such as synchronisation, still requires much work on the behalf of the software companies.

Although some development is underway on the functionality of individual COTS simulation packages, further research is required to determine how model interoperation can be standardised between heterogeneous packages. Even once this standardisation work has taken place a body of research, in parallel, could well be required to investigate the tools and methodologies required by the simulation engineer to aid the development of large models within a team. This has been eluded to in more recent years, see Hibino et al. 2002. Concurrent development of a simulation model would require a tool set and methodologies similar to that used by software engineers. i.e. source code control (or model control) and version control. Further, the paradigm could be extended to include specific development tools for the simulation modeller, for example, determining the best partition points within a simulation – this could be calculated through experimentation, possibly an extension to the simulation optimising tools currently available. It also believed that

the paradigm could be extended to include specific methodologies and practices for use in large model development, in much the same way that project management and systems management methodologies are used in large IT developments (such as PRINCE or SSADM). Extensions to existing software development tools could also be investigated, such as UML (the Unified Modelling Language), to include a standardised set of development stages and model definition.

## AUTHOR BIOGRAPHIES

**MICHAEL D. RYDE** is a Ph.D. student at the Department of Information Systems and Computing, Brunel University in the United Kingdom. He also received his M.Sc. at Brunel University in 2000 and is a member of the university's Centre for Applied Simulation Modelling (CASM).

**SIMON J.E. TAYLOR** is the Chair of the Simulation Study Group of the UK Operational Research Society and the collaborative simulation-modelling forum, the GROUPSIM Network ([www.groupsim.com](http://www.groupsim.com)). He is a Senior Lecturer in the Department of Information Systems and Computing and is a member of the Centre for Applied Simulation Modelling, both at Brunel University, UK. With Dr Gary Tan of the School of Computing, National University of Singapore he is joint leader of the UK (EPSRC)/Singapore (DSTR)-funded BRUNUSIM distributed simulation research programme. He has an undergraduate degree in Industrial Studies (Sheffield Hallam), a M.Sc. in Computing Studies (Sheffield Hallam) and a Ph.D. in Parallel and Distributed Simulation (Leeds Metropolitan). His main research interest is collaborative simulation modelling. He is also a member of the London-based Purple Theatre Company.

## REFERENCES

- Boer, C.A. and Verbraeck, A. 2002. Connecting High level Distributed Simulation Architectures: An Approach for a FAMAS-HLA Bridge. In *Proceedings of the 14<sup>th</sup> European Simulation Symposium*. Society for Computer Simulation Publishing House, Erlangen, Germany. 398-405.
- Brooks, R.J. and Robinson, S. 2001. *Simulation*. Palgrave, Hampshire, UK 32-35.
- Carothers, C.D., Fujimoto R. M., Lin, Y., and England P. 1994. Distributed simulation of large-scale PCS networks. MASCOTS 1994.
- Clarke, R. 1993. Module interconnection frameworks for a real-time spreadsheet. Computer Abstracts International Database, reference: 39\_1890.
- Fujimoto R.M. 1990. Discreet event simulation; Communications of the ACM. Vol. 33, No 10.
- Fujimoto R.M. 1999. Parallel and distributed simulation; In *Proceedings of the 1999 Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds: 122-131.
- McLean, C. and Riddick, F. 2000. The IMS Mission Architecture for Distributed Manufacturing Simulation. In *Proceedings of the 2000 Winter Simulation Conference*, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds. Association for Computing Machinery Press, New York, NY. 1539-1548.
- Miller J, Fishwick P.A., Taylor S.J.E., Benjamin, B. and Szymanski, B. 2001. Research and Commercial Opportunities in Web-Based Simulation. *Simulation: Practice and Theory*. 9(1-2), pp. 55-72.
- Pidd M., 2002. Simulation Software and Model Reuse: A Polemic. *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. Association for Computing Machinery Press, New York, NY. 772-775.
- Strassburger, S. 2001. Distributed Simulation Based on the High Level Architecture in Civilian Application Domains. Society for Computer Simulation Publishing House, Erlangen, Germany.
- Sudra, R., Taylor, S. J. E. and Tharumasegaram, J. 2000. Distributed Supply Chain Simulation in GRIDS. In *Proceedings of the 2000 Winter Simulation Conference*, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds. Association for Computing Machinery Press, New York, NY. 356-361.
- Taylor, S. J. E., Bruzzone, A., Fujimoto, R., Boon Ping Gan, Straßburger, S. and Paul, R.J. 2002. Distributed Simulation and Industry: Potentials and Pitfalls. *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. Association for Computing Machinery Press, New York, NY. 688-694.
- Zeigler B.P., Kim, D. and Buckley, S. J. 1999. Distributed supply chain simulation in a DEVS/CORBA execution environment; In *Proceedings of the 1999 Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds: 1333-1340.

# METHODOLOGICAL REFLECTIONS ON COLLABORATIVE WORK PRACTICES IN SIMULATION MODELLING: A SHORT JOURNEY TO ELSEWHERE

Gabriella Spinelli and Simon J.E. Taylor  
Centre for Applied Simulation Modelling  
Department of Information Systems and Computing,  
Brunel University,  
Uxbridge, Middlesex, UK

## ABSTRACT

Effective collaboration is at the heart of simulation modelling. By looking at the investigations undertaken within other discipline domains, e.g. Computer Supported Cooperative Work (CSCW), we highlight

**Keywords:** *Simulation Modelling, Collaborative Work, Groupware, Distributed Cognition and Common Information Space.*

## INTRODUCTION

“Simulation modelling is impossible without collaboration” is possibly one of a very select set of truisms that one might make of this decision support technique. However, to confirm this we cite Robinson and Pidd’s (1998) study of end user expectations. They identify “effective collaboration” as being one of several critical factors of success. Poor collaboration between simulationists and stakeholders may not always guarantee poor results from a simulation study but it will certainly contribute towards it.

Of the work in this area there are interesting and invaluable lessons in how we might work together in a simulation study. However, if a criticism might be made, these are written by simulationists for simulationists. As part of a wider series of events, the GROUPSIM Network (a UK Government funding program to investigate strategies in infrastructures for collaboration simulation modelling [www.groupsim.com](http://www.groupsim.com)) has begun to foster promising cross-disciplinary studies. One theme that has emerged is on the importance of research method used to study collaboration. This paper reports on one important aspect of this work by presenting an overview of methodologies used in an analogous and complementary field – a short journey to elsewhere if you will.

The paper is structured as follows. First we present some observations on the study of collaborative work practices. We then present an emerging theoretic framework used to study collaboration. The *Common Information Space* for collaboration is then presented with some methodological concerns. The paper then

methodologies issues that could be applied in the area of simulation modelling to support a more effective project timeline and to stimulate a discussion about the next generation of tools for collaborative simulation.

finishes with some brief observations from a study and some conclusions.

## STUDYING COLLABORATIVE WORK PRACTICES

In general, being a social and organisational phenomenon, collaboration incurs some costs. Additional work is required in order to achieve cooperative activity, beside the effort spent toward the task (Schmidt, 1994). The cost is both visible at individual as well as at an organisational level since supplementary resources need to be strategically planned and marshalled in its support. Collaboration costs but its benefits balance the outlay invested in its establishment and maintenance. In particular, the advantages of collaboration lie in the overcoming of individual’s limited capabilities, providing constructive opportunity for mutual critical assessment, confrontation of perspectives, combination of differences and the enhancement of individual capacity.

*Space* and *tools* constitute the technical resources of any human activity and defines the context where activities take place. Kirsh (2001) defines with *activity context*, a structured amalgam of informational, physical and conceptual resources whose interactions are not yet clear. Understanding the network of relations between the several components that make up a collaborative system represents the intent of many researchers in the area of Computer Supported Collaborative Work (CSCW) in order to plan a proactive approach to the design of technological and organisational solutions for collaboration. (Bannon, 1992; Bannon and Hughes, 1993; Grudin and Poltrock, 1997).

While Human Computer Interaction focuses on the study of the partnership between individuals and technological equipment and on the interaction

engaging single users with considerably small equipment (Bannon and Huges, 1993), CSCW, on the other hand, embraces a broader spectrum of human activities. The intention of supporting collaboration as it occurs in the real world and through real practices involves a redirection of the analytical tools employed in the research and a new range of methodologies for the collection and analysis of data and evaluation of users' experience. Extending the territory of observation for studying co-operative activities results in the identification of a new unit of analysis as a complex arrangement embedding artefacts, emergent behaviours and mediation. In order to consider such a unit of analysis, the required paradigm of cognitive science to frame the study needs to be enforced by approaches that allow a broader view on the phenomena under analysis, involving disciplines and new methodologies able to enrich the insights about collaboration (Bannon, 1992; Suchman, 1983). A cognitive theoretical framework such as Extended Cognition, in conjunction with disciplines as Anthropology and Ethnography can support the understanding of medium-to-long-term user study, involving participants in *real settings and under natural circumstances*.

A further tenet within CSCW community is that designing technology is not just about designing artefacts but also social practices and possibilities that are realised through their employment (Flores *et al.*, 1988). In this perspective, space is the setting that surrounds us. It is not just through the physical properties and the interactions between space, artefacts and human body that we construct a meaningful environment to our activities. We perceive and understand the workspace not just by looking at the locations of artefacts and three-dimensional arrangements, but also by making sense of the resources and of the way we can use them (Harrison and Dourish, 1996). Space seems to be the structural precondition for socio-cultural reality: the *place*, a collection of people, beliefs, rules, artefacts and interpretations. It is through ethnographically oriented studies that we might regard different activity contexts in order to understand how artefacts and space affect the settings where collaboration occurs.

## A THEORETICAL FRAMEWORK

With the intent of simplifying the analysis of complex phenomena, naturalistic approaches have confined the study of cognition to lab experiments and observations. This selective attention to some aspects of human activity has neglected the complexity that shapes our cognitive capabilities, which are intrinsically linked to the social and historical context where they occur. Several schools of thought, motivated by the intent of re-contextualising human intelligence, identified a larger unit of analysis able to account for the role of external resources in the moulding of human plans,

actions and collaboration. A composite theoretical framework labelled as Extended Cognition has brought forward the concept of mediation where physical and cognitive tools are considered as catalysts and products of the higher human psychological functions. Tools are embedded into the relation we establish with the outside world changing the nature of the interaction with it. Once they are embedded in activities, artefacts are mediating links between individuals and the world. Vygotsky (1978) and the Soviet School of Cultural Psychology have identified a complex unit of analysis, the *activity*, as a triad of *subject-tool-object*, that was further developed by the *Activity Theory (AT)*. AT regards collective activities and expands the basic ideas of artefacts and their mediation in everyday life.

The basic triadic relation *subject-tool-object* identified by the Soviet School is stretched with the intention of embracing a broader context that provides the configuration of resources involved in human performance. Within the network established by the components of the *activity system*, artefacts represent the media supporting our cognition and the *loci* where it is externally distributed. The augmentation operated by the AT creates a more comprehensive understanding of the artefacts' mediation between people and context, and therefore generates a more predictive framework for informing the design of artefacts. Tools become not mere filters through which we perceive reality, but actors that help define our objectives and ultimately our identity.

A more radical perspective on human activity is taken by *Situated Action*, a theoretical framework that finds its origin in ethnomethodology and branches from traditional cognitive science, rejecting the tautology for which "cognition is *just* computation" (Suchman, 1987). Situated Action is a radical account of human behaviour that is not based on plans or on cultural universals but on the *situatedness* that characterises human acting (Salomon, 1993). The emphasis of the approach resides on the interaction between the individual and the environment (Nardi 1996), resulting in a new unit of analysis: the *person-acting-in-setting* (Suchman, 1987). The contribution of Situated Cognition to the overall theoretical framework resides in its interpretation of human activity as contingent re-orientations of resources performed *in situ*. This is to achieve the most suitable arrangement that allows us to undertake a potentially successful next step in the course of action.

Suchman (1987) suggests that activity is an emergent phenomenon whose values are developed at the same time that the activity unfolds. Activities are not driven or structured according to preconceived plans. Once we are engaged in an event, we try to direct its course in an opportunistic way, in a step-by-step computational process performed within the immediacy of the situation we are experiencing. The specificity of the circumstances where the activity occurs can not be

transcended. Situated Action accounts of actions as if they are always determined by material and social circumstances. Thus our activities can not be fully understood if their study transcends the context where they occur. This makes human actions unpredictable to determine, while consistency can be found in the set of transformations aiming to structure the resources for the activity. For Situated Action, the achievement of intelligent strategies is based on the use of circumstances and this provides a correction to the simplified view that cognitive science has held. The elegant theoretical structure of traditional cognitive science is rejected and reveals its fallacies when human action is studied as a phenomenon not solely centred on human mind.

The contingent nature of human cognition has been further investigated by a cognitive approach, *Distributed Cognition* (DCog). The framework emphasises the distributed nature of cognitive processes and the transformation that information undergoes in order to get into a specific format that is the most appropriate for the performance of a task. DCog incorporates external and internal resources into a larger cognitive system, the *socio-technical system*, where human and technological components are both regarded as media for information representation and transformation, despite their intrinsic differences. The *socio-technical system* is based on the principle that components, being they humans or technologies, hold information representations that are manipulated, co-ordinated and propagated, changing the state of the overall system which, by a set of transformations accomplishes its cognitive task. People and artefacts are media that carry fragments of information that are necessary for the ultimate goal of the *socio-technical system*.

It is through observational studies that DCog promotes the understanding of complex cognitive systems with the intent to discover strategies that a distributed cognitive entity opportunistically chooses to take in order to achieve the desirable state, given its environmental circumstances. Artefacts change the nature of the task making them less 'cognitively expensive' by engaging human skills that are not limited. In ideal situations we delegate to the environment and to the artefacts the load of information we cannot mentally deal with, and the processes we cannot compute internally, yet being able to achieve an effective performance.

The theoretical approach of *Extended Cognition* configures a new landscape for the study of intelligence as a property that is manifested as *people in action*. With the intent of establishing the realignment of mental and physical nature of human intelligent behaviour, the Soviet School of Psychology, Activity Theory, Situated Action and Distributed Cognition shorten the distance between theoretical

*apparatus* and the realm of design of new information technologies. The aim is to stimulate a theoretically informed design (Hollan *et al.*, 2000), which accounts for the social and environmental embedded nature of human cognition.

## A COMMON INFORMATION SPACE FOR COLLABORATION

The need for a larger unit of analysis in order to analyse how we work together finds application in the concept of Common Information Space, CIS. Bannon and Bødker (1997) identify CIS as the shared informational environment required for grounding the communication and co-ordination of cooperative activities, i.e. how we collaborate. CIS is meant to refer to both the artefacts that carry information, the representation of information, and the meaning attributed by the user to these representations in a shared space. The value of utilising the notion of CIS in understanding collaborative work *in situ* is its focus on the seamless interweaving of people, artefacts, information and activities.

Sharing information for collaborative activities can lead to problematic situations in either co-located or distributed settings (Reddy *et al.* 2001). When actors are physically and temporally separated, expensive strategies need to be employed in order to package the relevant context of information that needs to be communicated. Interpretation and negotiation problems can also arise if participants, sharing the same space and timeframe, do not work toward a common interpretation of the information at hand.

In general terms, we hope to use the general concepts of CIS to tease out from field research a greater understanding of the difference between a physically supported collaborative space, with its rich resources and a virtual or distributed collaborative space, which all tend to function in a more impoverished form. In doing so we highlight why the current design of information technology that users employ in such spaces do not truly support their current collaborative needs.

## METHODOLOGICAL CONCERNS

In a study of collaborative activities in order to capture the complexities of the various CIS under investigation, ethnographically inspired fieldwork observations and interviews were undertaken to document users' activities, their context of work and the artefacts they employed (Spinelli and Brodie, 2003a; 2003b). The first study in the research focused on three design teams co-located and distributed; while the research on mobile work turned its attention to collaboration in a variety of remote and mobile settings, such as at airports and on trains.

The field data collection spanned approximately eight months. Observational work was supported by methods such as digital video recording of events; digital photography, contextualised interviewing and participatory user data reviews which helped to capture the richness of interaction that was occurring in the various CIS under review. Furthermore, participants took part in collaborative sessions where they reviewed some of the observational data and offered valuable insights and understanding of the critical collaborative scenarios observed. These served to highlight implicit work practices and workarounds elaborated in the attempt to avoid the disruptions that the use of technology in collaboration can cause.

## CO-LOCATED AND DISTRIBUTED COLLABORATION

In order to select the study sites, it was important to take into account some considerations that have methodological and content relevance to the research. Firstly we aimed to select organisations that could provide the opportunity to follow an entire project or at least a well-identified phase of it in order to see the establishment, evolution and maintenance of the co-ordination patterns. Also it was worthwhile pursuing the opportunity to observe more than one team in order to compare and contrast the different way of organising collaborative work.

The study was framed within a consistent domain of observation. Three organisations were selected on the basis of the activities they performed. The overall choice was made considering the nature of the collaborative activity as the most important aspect to emphasise for the selection of the work context(s) to observe. This was in order to avoid too many differences that would not allow the comparison of the observations. The three teams that were shadowed were all involved in design activities of different types, as listed below:

- the conceptual design of an information appliance;
- the engineering design of an innovative public building; and
- the design of a new set of national standards in construction procedures.

## INSTANCES OF COLLABORATION

Three diverse instances of collaborative work emerged from the observations:

- a *physically-centred* collaborative space (the project space), a dedicated environment where a group of professional designers collected and manipulated information in order to support their activities;

- a *virtually maintained* space, resulting from the combination of web application and tele-video conferencing technologies for the collection, retrieval and storage of organisational knowledge to support problem solving activities;
- a *locally distributed* space arising from the collected use of several digital devices (mobile phones, faxes etc.) and protocols of communication (circulation of the people, email, snail mail etc.);

All the instances of collaborative space observed in this study do not find counterparts just in the physical world. They resemble more a collection of established organisational practices and technologies used to achieve collaborative tasks. This observation led us to postulate that we cannot rigidly define collaborative space by simply considering its physical boundaries. This consideration thus directed our research towards the identification of those tasks that make up the dimensions of collaborative work and of those features that seemed to be crucial across the field observations in supporting collaboration. Further results from these studies can be found in Spinelli and Brodie (2003a, 2003b).

## CONCLUSIONS

This paper has given an overview of methodological observations from disciplines that study the nature of collaboration. The purpose of this paper is to raise awareness of methodological issues in the study of collaboration in simulation modelling. It is hoped that this “short journey to elsewhere” will provoke thought and debate in this area that will lead to better collaboration and reduced simulation project costs.

## REFERENCES

- Bannon, L.J. 1992. Discovering CSCW. Proceedings of the 15<sup>th</sup> Information Systems Research in Scandinavia (IRIS), Larkollen, Norway, Aug 9-12, 1992, 507-520.
- Bannon, L. and Bødker, S. 1997. Constructing Common Information Spaces. In *Proceedings of the 5th European CSCW Conference*. Dordrecht: Kluwer Academic Publishers.
- Bannon, L. and Hughes, J. 1993. The Context of CSCW. In Schmidt K. (ed) *Report of COST14 “CoTech” Working Group4*, 9-36.
- Flores, F., Graves, M., Hartfield, B. and Winograd T. 1988. Computer Systems and the design of organisational interaction. *ACM Transactions on Office Information System* 6, 2, 1988, 153-172.
- Grudin, J. and Poltrock, S. E. 1997. Computer supported Cooperative work and Groupware. In Zelkowitz (ed), *Advances in Computers*. Orlando, FL: Academic Press.
- Harrison, S. and Dourish, P. 1996. Re-place-ing space: the roles of place and space in collaborative

- systems. In *Proceedings of CSCW 1996*. ACM Press New York, NY, USA.
- Hollan J., Hutchins, E., and Kirsh, D. 2000. Distributed Cognition: toward a new foundation for human computer interaction research. *ACM transactions on Human Computer Interaction*, 7(2): 174-196.
- Holtzblatt, K. and Beyer, H. 1998. *Contextual Design. Defining Customer-Centered Systems*. San Francisco, Morgan Kaufmann Publishers, Inc.
- Nardi B. A. 1996. *Context and Consciousness*. The MIT press, Cambridge, Massachusetts, London, England.
- Reddy, M. C., Dourish, P. and Pratt, W. (2001) Coordinating Heterogeneous Work: Information and Representation in Medical Care. In *Proceedings of the Seventh European Conference on CSCW*. Netherlands, Kluwer, 239-58.
- Robinson, S. and Pidd, M. (1998). Provider and Customer Expectations of Successful Simulation Projects. *Journal of the Operational Research Society*, 49 (3): 200-209.
- Salomon G. (Ed) 1993. *Distributed Cognition: psychological and educational considerations*. New York: Cambridge University Press.
- Suchman L. A. (1987). *Plans and situated actions: the problem of human-machine communication*. Cambridge: Cambridge University Press.
- Spinelli, G. and Brodie, J. (2003a). Fieldwork implications for the design of Common Information Space (CIS) in collaborative work. In *Proceedings of HAAMAHA 2003*, 8<sup>th</sup> International Conference on Human Aspects for Advanced Manufacturing: Agility and Hybrid Automation. Rome 27-30<sup>th</sup> May 2003, 397-404.
- Spinelli, G. and Brodie, J. (2003b). Towards an understanding of Common Information Spaces in Distributed and Mobile Work. In *Proceedings of HCI international 2003*, Crete 22-27 June, 859-863.
- Vygotsky L. S. 1978. *Mind in society: the development of higher psychological processes*. Ed. Cole M. Cambridge, MA: Harvard University press

# DISTRIBUTED, OPEN SIMULATION MODEL DEVELOPMENT WITH DSOL SERVICES

Niels A. Lang  
Erasmus University Rotterdam  
Rotterdam School of Management  
Dep. of Decision & Information Sciences  
Burg. Oudlaan 50, 3000 DR Rotterdam, NL  
nlang@fbk.eur.nl

Peter H.M. Jacobs and Alexander Verbraeck  
Delft University of Technology  
Faculty of Technology, Policy and Management  
Systems Engineering Section  
Jaffalaan 5, 2628 BX, Delft, the Netherlands  
p.h.m.jacobs@tbm.tudelft.nl; a.verbraeck@tbm.tudelft.nl

## KEYWORDS

Distributed simulation, object-orientation, simulation services, simulation environment.

## ABSTRACT

Information technology innovations, like web-services and object-oriented design, are fast finding their way into Business Information Systems (BIS). At the same time similar developments in the field of business system simulation do not gain momentum. This observation, combined with research needs, has been the starting point for the development of an open, object-oriented, distributed and extendible research test-bed for business simulation, called DSOL. DSOL, which stands for 'Distributed Simulation Object Library' was introduced on Wintersim '02 (see Jacobs, Lang and Verbraeck 2002), and has now evolved into a firm distributed simulation core, extended with several services for visualization, event notification and BIS integration.

This paper illustrates in detail the way in which concepts and principles on discrete-event, multi-formalism simulation have been translated into the object-oriented, distributed DSOL environment. It introduces the simulation concepts underlying the DSOL design, the DSOL implementation itself, example models and finally discusses the outcomes.

## SIMULATION FOR BUSINESS DESIGN

Recently, several internet web service based architectures have been introduced that promise to streamline inter- and intra-organizational communications. These developments allow organizations to integrate their distributed business systems fast and effective and, as a result, improve performance of and control over their business processes.

A corresponding movement towards open, distributed interaction standards has not yet gained momentum in the field of simulation. Most simulation packages have limited and incomplete capabilities for interaction with other systems, effectively limiting the potential of linking simulations to other types of information systems, such as ERP systems and databases. Moreover, standards for interoperable simulation systems are not yet widely accepted, limiting the development of multi-model simulation systems.

The Java-based Distributed Simulation Object Library (DSOL, see Jacobs, Lang and Verbraeck 2002) was developed to fulfill the need for such an open integrated platform for business simulation. Section 1 describes the simulation concepts underlying the platform. Section 2 presents its current implementation. Section 3 presents the *M/M/1* queue example. Section 4 finally discusses the results achieved thus far.

We regard simulation to be a model-based approach for ill-structured problem solving (Sol 1982), a model being *a representation of a part of reality, constructed for a particular purpose*. Simulation has been defined by Shannon (1975) to be:

*'The process of designing a model of a concrete system and conducting experiments with this model in order to understand the behaviour of a concrete system and / or to evaluate various strategies for the operation of the system.'*

We furthermore distinguish between *conceptual models*, that provide a problem-contingent language for system model design and *system models*, instantiations of the conceptual model describing a part of the system of interest. The system models allow experimentation. Describing a system model in terms of a conceptual model eases its description (many system model parts can be related to a single concept) and allows for the definition of consistency rules. Such rules constrain the model designs allowed and therewith ease model

understanding.

While our definition of simulation speaks of the design of 'a model', we follow Simon 1973 in that ill-structured problems by their very nature require the development of *multiple* models. The overall ill-structured problem solving process does not proceed in a straightforward manner, but proceeds by alternating continuously between model development for (chosen) sub-problems and sub-model integration. Changes may not only occur in system models but also in the underlying conceptual model, yielding the requirement for flexible modeling of both.

The simulation approach described thus far could be applied in many fields. We will in this paper, however, focus on its appliance in the field of business systems. Current business systems have three characteristics that we deem relevant for the simulation approach.

First of all, there is little doubt on the *complexity* of many current business systems. Business processes easily involve hundreds of steps, complicated planning and scheduling decision procedures and thousands of distributed resources.

Moreover, businesses are globalizing since the dawn of the 20th century (Acs, Morck and Yeung 2001), under the drivers of market liberalization and cheap communication infrastructure. This trend implies that business design problems essentially has become *distributed* in two ways: not only the object of design (e.g. the business system) has become an inherent distributed system, the process of design, with typically involves many stakeholders worldwide, is moving towards distributed collaboration as well.

Finally, the trend of automation and information drives the *virtualization* of business systems. Business process organization and management is more and more realized in complex ICT systems (such as Enterprise Resource Planning (ERP), Supply Chain Management (SCM) and many others), which allow for (computationally) complex business processes and offer global business connectivity.

As a result, a simulation environment supporting simulation for the current business environment should pay explicit attention towards the aspects of system complexity, distributed systems and the presence of virtual business information.

The remainder of the section presents a conceptual framework for simulation environments which fits the described peculiarities of business simulation.

### Simulation as experimenting

Figure 1 introduces an experimentation oriented framework of the simulation environment, in line with our definition of simulation. The framework is based on

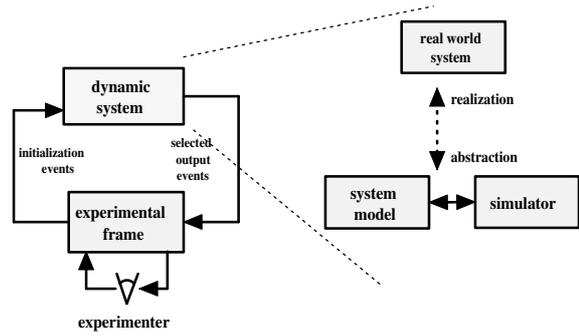


Figure 1: Experiment oriented framework for simulation

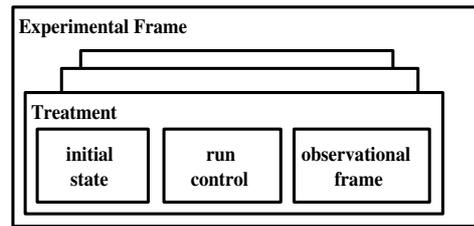


Figure 2: The experimental frame defined

Zeigler, Praehofer and Kim (2000) and introduces two main concepts, that of 'experimental frame' and 'dynamic system'. According to Zeigler, the experimental frame is a '*specification of the conditions under which the system is observed or experimented with*'. The system being experimented with is basically any dynamic system, with the experimental frame defining and limiting the scope of the observations with regard to time, region and aspect. In the context of simulation the dynamic system is either a concrete system, or a corresponding model. We call an experimenter developing and experimenting with several models (in order to improve the concrete system's behaviour) a *designer*. In [Verbraeck / Sol] the concept of experimental frame is defined more precise, by regarding it as a collection of *treatments*, where each treatment comprises an initial state, run control parameters and an observational frame, see figure 2. Using this definition, it is possible to define roles other than experimenter or designer. We define an *observer* as a participant interacting only via an observational frame. We finally define a *player* as a participant interacting using an observational frame and an *action frame*, which allows the player to provide input to the dynamic system during an experiment. The last role essentially turns the simulation model into a game model.

With respect to business simulation, we observe that the framework allows concurrent, distributed collaboration, since no restriction is imposed on the communication used or the number of participants concurrently

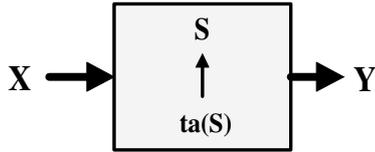


Figure 3: The DEVS model formalism

involved. We finally recall the 'virtualization' of business systems. This basically implies that real world realizations of business ICT systems do already contain models and data of the business system they support. In potential such models and data could considerably speed up the development of a system model of the business system, if services to convert them into a simulation model are made available. A typical example of this would be to retrieve workflow processes and data from a company's ERP system and use this as the basis for a workflow simulation model.

### DEVS: a base formalism

We distinguish between simulation model and simulator (Zeigler, Praehofer and Kim 2000), illustrated in figure 1. Whereas the model defines structure and behaviour, the simulator actually brings the model behaviour about by *executing* the model in a time controlled manner. The simulator allows the experimenter to control speed, start and end times, by discretizing the overall behaviour into events, e.g. atomic state-changes. The separation between model and simulator raises a need for a *contract* defining the interaction between model and simulator. We'll define a formally defined contract to be a model formalism. It follows that a simulator for a given formalism is able to execute all models specified in that formalism.

In business simulation, many types of models are used, ranging from complex control systems, to simple workflow models and to distributed gaming models. For integrated business analysis, it is often desirable (but not always feasible!) to integrate such models of different but related business system aspects. To ensure the applicability of the DSOL simulation environment for these situations, a choice was made for a base formalism of high expressive power (Zeigler, Praehofer and Kim 2000; Vangheluwe and de Lara 2002): Discrete Event System Specification (DEVS).

Figure 3 schematically introduces the DEVS concept, which is presented here with an emphasis on illustration, not on rigor.  $\mathbf{X}$  defines the set of input values,  $\mathbf{Y}$  defines the set of output values. The model has a state  $\mathbf{S}$ , which is a function of an incoming value and of the internal transition function. The latest is triggered by the function  $\mathbf{ta}(\mathbf{S})$  which defines the time for the next internal state transition. The model output ( $\mathbf{Y}$ )

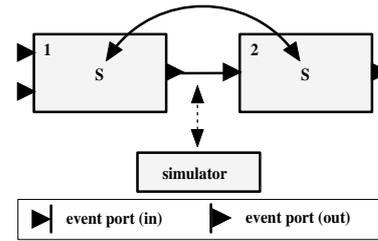


Figure 4: The extended DEVS formalism

is defined to be a function of the model's state only.

The formalism introduced thus far imposes no constraints on the semantics of input, output or state. It can be extended to encompass specific worldviews like transaction-flow or actor oriented ones.

### Modularity and hierarchy

The DEVS formalism as introduced thus suggests the model to be a monolithic entity. As such, it is still of little use to us, since a monolithic model will be as complex (or worse) as the part of reality being modeled. The DEVS formalism has, however, been extended with general systems concepts enabling the description of multi-component, modular and hierarchical DEVS models. This extended DEVS formalism is illustrated in figure 4.

It introduces the concept of a *multi-component* model connected by *event ports*. Between event ports, *connections* can be attached, allowing components to interact by the exchange of events. Actual event propagation is facilitated by the DEVS simulator, which is in this way able to control the speed of inter-component interaction. The formalism does not impose any restrictions on how event interaction is actually realized thus allowing for distributed implementations.

The possibility to break down the model in several components eases model development and enhances understanding. The concept of event ports furthermore allows for *modularity*, e.g. components managing a self-contained, not externally visible state. By restricting component interaction only to take place via event connections, components in effect become modules, with event ports defining their interfaces. In such a setting the direct interaction (bypassing event connections) shown in figure 4 is thus forbidden. The well-known advantages of modular models is that they allow for separation of concerns and freedom of implementation: essential qualities for complex business system modeling.

Finally, figure 5 illustrates the suitability of the DEVS formalism to construct models containing several levels of abstraction, e.g. hierarchical models. Modularity is maintained by restricting internal mod-

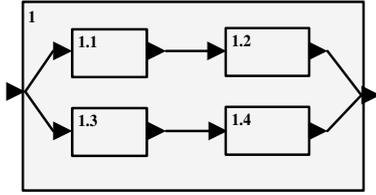


Figure 5: Hierarchical DEVS

els to connect only to event ports of or internal to the parent model. The relevance for business system modeling is that business system design occurs at different levels of abstraction, from individual workplace design to inter-organizational coordination (Sol 1982).

### Towards multi-formalism simulation

The extendability of the DEVS formalism has already often been demonstrated (Zeigler, Praehofer and Kim 2000; Vangheluwe and de Lara 2002). However, many simulation packages, although often based on DEVS, only provide access to a higher level set of concepts, such as the transaction-flow worldview. Given the already mentioned business problem diversity the challenge is to realize an environment which is open to new formalisms (realizing conceptual freedom). A sound basic formalism and operators for its extension are therefore essential conditions. We hold DEVS to be such a formalism.

### SIMULATION WITH DSOL

This paper introduces DSOL: a Java based research test-bed for simulation in object oriented, distributed environment. Before we introduce DSOL's implementation of the fundamental DEVS concepts, we start here with an overview of the services provided by the framework. Currently DSOL consists of:

- A core DEVS simulator introduced throughout the rest of this section. From a perspective of information system design, this simulator is set up as a remotely accessible service. It fully supports all current enterprise information system standards.
- A basic statistics library consisting of a number of pseudo random number generators, tallies, charts, counters, etc (see Law & Kelton 2000). Planned extensions include dynamic input analyzers, monte carlo analysis and the integration with data mining suites.
- A 2D visualization and representations described in Jacobs, Lang and Verbraeck (2002). 3D animation is currently tested based on Sun's 3D API.

### Remote method invocation scheduling

In the previous section we focused in great detail on the DEVS formalism and pointed out that this formalism is based on scheduled interaction. The challenge of the DSOL framework is to see to what extent an object oriented programming language like Java supports this principle of interaction and how it is able to schedule it in a profound and natural way.

The synergy between an object oriented language and the DEVS paradigm becomes clear when we consider that both are based on a principle of interaction. In an object oriented programming language, objects interact by the invocation of methods. An object oriented programming language furthermore distinguishes *public*, *private* and *protected* methods. Where private methods are only accessible to the object itself, protected methods are also accessible to instances of subclasses (in the Java programming language protected methods are also accessible to all classes in the same package) and public methods are accessible to everyone.

For us it became clear that the most essential part of designing a DEVS based simulator was to support this notion of object oriented interaction; instead of the direct method invocation which occurs in normal Java programming, DSOL schedules method invocation based on the following requirements:

- The scheduled method invocation must map on the scheduled state change illustrated in figure 3. A *SimEvent* illustrated in figure 6 maps the event introduced by the DEVS framework.
- All public methods must be able to be scheduled and invoked by the DSOL simulator. There may be no constraint on the method name, return type or any argument.
- Overloading and polymorphism must be supported. This results in a simulation framework fully supporting all potential distinctions between methods.
- Though both the simulator and the object on which interaction is scheduled may reside within the same Java virtual machine, DSOL is designed for distributed interaction. All simulation events must therefore be serializable.

The modular approach introduced in the first section encouraged us to accept that potentially better implementations of the scheduled method invocation may be developed. DSOL nevertheless provides a reference implementation based on all above requirements (figure 6).

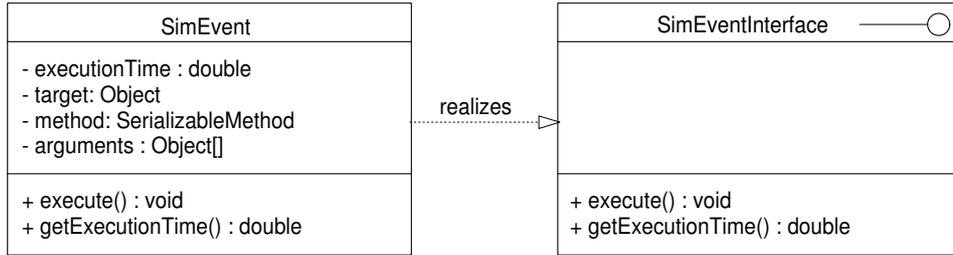


Figure 6: Reference implementation of a DSOL simulation event

### Process modeling in an object oriented simulation framework

In section 1.2 we elaborated on the complexity of conceptualizing an ill structured real-world business system under investigation. We furthermore emphasize on Banks' (Banks, 1996) recommendation of involving the problem owner in this activity. Finally we conclude that this often leads to a resource based conceptualization of the real-world business system. This section focuses on ways to support the translation of these resource based concepts, e.g. resources, queues and flows, into a DEVS based specification framework like DSOL. There are several options to accomplish this translation:

The first option discussed here is the NULL option. We just do not provide any assistance in the translation of resource based conceptual models in DEVS based specification framework. Though Zeigler (2000) proved that all resource based concepts can fundamentally be explicated in DEVS concepts, this option leaves it is up to the designer to make this translation while coding the specification model.

A second option is to provide a library which enables the model builder to code the process in the entities of the specified simulation model. This is an approach followed by most Java based simulation frameworks such as Silk, SSJ, SimJava, etc. The following pseudo code reflects this approach.

```

public class Customer extends entity
{
    public void process()
    {
        server.request(1.0);
        //request a capacity of 1.0
        this.delay(1500); //suspends for t=1500;
        server.release(1.0); // resume and release
    }
}
  
```

Key consequences of this approach in an object oriented programming language are:

- The conceptual model is specified in long procedural code. For large models, this is in clear contrast with the required modularity and hierarchy supported both by the DEVS formalism and the Java programming language.
- The customer in the above pseudo code must be able to suspend himself for a particular time. In order to accomplish this the entity must break out a method and afterwards resume to the next line within this method. In an object oriented programming language this can only be accomplished by extending a thread (Healy 1998; Garrido 2001). This multi-threaded approach is in the Java programming no longer supported and considered deprecated (Sun 2002). The approach is inherently unsafe; arbitrary behavior can result, which may be subtle and difficult to detect, or it may be pronounced. We therefore state here that the results of simulation models based on this approach are fundamentally un-trustworthy.

The third approach presented in this paper is to design and develop a library for resource based conceptual models. This approach is implemented by most non object oriented simulation frameworks such as Arena, EM-Plant, Automod and a few object oriented frameworks such as SimKit and DSOL. In this approach one designs the model as a chain of stations. The following pseudo code reflects this approach:

```

public class SimulationModel
{
    public static void main()
    {
        //create a generator with interarrivaltime of
        //1.2 and a batchSize of 1
        StationInterface generator =
            new Generator(Customer.class,1.2,1);

        //create a delay of 1500 time units
        StationInterface delay = new Delay(1500);

        //create an exit station
  
```

```

StationInterface exit = new Station();

//now we create the flow
generator.setDestination(delay);
delay.setDestination(exit);
}
}

```

Key consequences of this approach in an object oriented programming language are:

- Since a simulation model is specified by the creation of a chain of stations , this approach keeps as close as possible to resource based or process oriented, e.g. IDEF-0, conceptual modeling languages.
- Since the model is programmed in the stations it is easier to replace a station by an extension. Specifying a model is done in a service oriented approach supporting both Java's component based and DEVS modular and hierarchical design patterns .
- Since incoming entities are scheduled from station to station there is no need for the suspension of threads. An entire model may be single threaded and entities can easily be serialized and streamed from computer to computer.

DSOL has followed both the first NULL approach and the third approach. The reason why this paper mentions the first NULL approach is that though DSOL provides a library of resource based concepts, they are considered an add-on. The underlying DEVS framework remains directly accessible.

### A-synchronous event model

In order to emphasize the advantages of implementing a DEVS formalism in an object oriented language this section illustrates the benefits of using an asynchronous event mechanism.

The asynchronous event mechanism consists of two sides: a listening client subscribes to a topic and an event producing publisher notifies all subscribed listeners on a particular change. The interfaces and reference implementation are illustrated in figure 7.

In figure 7 a *ListenerInterface* provides a method on which a callback is made possible and the *EventProducer* consists of a private *List* containing all subscriptions. A subscription consist of a listener and a topic. Whenever a producer invokes a *fireEvent* method, all listeners are one by one matched on the topic and if required notified.

Though the above elaboration on an asynchronous event mechanism might sound all too familiar to experienced Java programmers, it is good to understand and emphasize on its consequences:

- An asynchronous event notification mechanism supports point-to-multipoint interaction between model components and distributed (web-based) representation components.
- An asynchronous event notification mechanism supports dynamic soft coded relations between deployed (web) services.

### EXAMPLE: AN $M/M/1$ QUEUE

This section describes how to implement the traditional  $M/M/1$  queue: a convenient example to illustrate DSOL. It is furthermore used as an example by L'Ecuyer [L'Ecuyer, 2002] to illustrate SSJ. Since the SSJ framework is a multi threaded process oriented simulation framework, the combination of papers illustrates the difference of specifying the resource based conceptual model in entities versus stations.

In the  $M/M/1$  queue, the service time follows a normal distribution with  $\mu=0.8$  and  $\sigma=0.1$ . The system starts empty and has a runlength of one million time units. Entities are created with a batchsize=1 and an interarrival time following an exponential distribution with  $\beta=1.0$ .

Figure 8 illustrates the code for this system. First of all we see the creation of a simulator. The next step is to set the runlength of this particular simulator. Then we create a random stream to be used within the application. For this particular example we have used the Mersenne Twister random stream developed by Makoto Matsumoto [Matsumoto, 1998].

The first resource based building block used within figure 8 is the generator. The generator creates instances of a class by the scheduled invocation of its appropriate constructor. In order to deal with constructor overloading we submit besides the name of the class, the array of arguments for the constructor and the array of classes of which these arguments are instances. We also provide distributions describing a start time, an interarrival time and a batchsize. The fact that we generate instances of the *java.lang.Object* class emphasizes that there are no restrictions within the DSOL framework to the postponed invocation of methods or constructors.

The next steps are very straitforward. First we create a resource with a capacity of 1.0. Then we create a seize block which claims 1.0 unit of the resource. After the resource is successfully claimed, the entity will flow to the server which will delay the entity for around 0.8

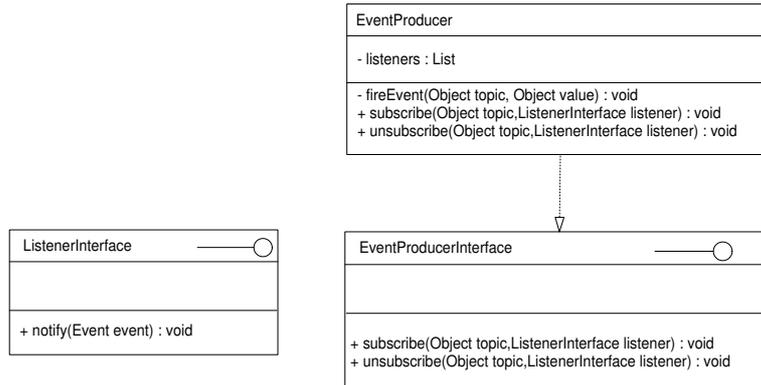


Figure 7: Reference implementation of DSOL's asynchronous messaging

time units and release the resource. Finally we initialize and start the simulator.

## DISCUSSION AND CONCLUSIONS

This paper started with the need for an interaction between simulations and business systems. The problem with this integration is of course that these two worlds differ very much. Where simulations abstract and reduce reality, business systems try to fully capture that reality. The simulation clock runs faster than reality, but the business system has no choice but to operate in real time. On the other hand, there *is* a clear need for tight integration, because the most time consuming tasks in simulation studies are the mapping of the business system on the simulation and the process of gathering and preparing data for the simulation experiments. By using the asynchronous interfacing mechanisms that was introduced in section 2.3, data collection from real-life data sources is much easier than trying to create a tight interface between business systems and simulations. In addition, there does not need to be such a big difference between *delayed* method invocation and *real* method invocation. Architectures for creating the business systems can therefore also be used for simulating these systems, where the most important differences consist of the reduction of certain processes and the scheduling against artificial time. A welcome add-on is the possibility to create hybrid systems where simulation models are a component in a larger system, e.g. providing decision support functionality. Several projects have already been carried out where DSOL simulations easily exchange information with databases, spreadsheets, and external systems. Two interesting challenges will be researched in the near future: directly mapping the architecture of a business system in reduced form on a simulation architecture, and including DSOL simulation models as modules in decision support systems.

The DSOL framework as introduced offers more than a possibility for integration, it is also a basic DEVS simulator with many extensions such as statistics, animation and visualization, and process modelling. In contrast to many other approaches, though, these extensions are not tightly coupled to the DSOL core, but loosely coupled, enabling other implementations of these additions as well. The heavy use of Java interfaces in the implementation ensures that users extending or creating their own implementation of a certain functionality, do not have to change pieces of code that make use of the newly created software. It is, for instance, very easy to change the `SimEvent` or `EventList` in the core of DSOL for another implementation without making any changes to projects that have already been built on top of the DSOL core. The fact that DSOL is an open source project will hopefully stimulate specialists to indeed create better or faster implementations of some of the already implemented functionality of this Java-based simulation framework.

## OBTAINING THE SOFTWARE

DSOL is published under the General Public Licence. More information on the license can be found at <http://www.gnu.org/copyleft/gpl.html>. The DSOL project description can be found at <http://www.simulation.tudelft.nl> and the software can be downloaded from <http://sourceforge.net/projects/dsol/>.

```

public class MM1Queue
{
    public static void main(String[] args)
    {
        if(args.length!=0) System.out.println("Usage: java MM1Queue");

        SimulatorInterface simulator = new Simulator();
        simulator.setRunLength(1000000);

        StreamInterface randomStream = new MersenneTwister(555);

        //The generator
        DistContinuous generatorStartTime = new DistConstant(randomStream,0.0);
        DistContinuous arrivalTime = new DistExponential(randomStream,1.0);
        DistDiscrete batchSize = new DistDiscreteConstant(randomStream,1);
        GeneratorInterface generator = new Generator(simulator,java.lang.Object.class,null,null,
            generatorStartTime,arrivalTime,batchSize);

        //The queue and server
        Resource resource = new Resource(simulator,1.0);
        StationInterface queue = new Seize(simulator,resource,1.0);

        //The server
        DistContinuous serviceTime = new DistNormal(randomStream,0.8,0.1);
        StationInterface server = new Delay(simulator,serviceTime);

        //The flow
        generator.setDestination(queue);
        queue.setDestination(server);

        //Starting the model
        simulator.initialize();
        simulator.start();
    }
}

```

Figure 8: M/M/1 Queue

## REFERENCES

- Acs J., R.K. Morck and B.Yeung. 2002 *Entrepreneurship, globalization and public policy*, in *Entrepreneurship, globalization and public policy*, vol. 7 (2001), pages 235 - 251.
- Banks J. , J.S. Carson II, B.L. Nelson. 1996 *Discrete-Event System Simulation*. 2nd ed., Prentice Hall, Upper Saddle River, N.J.
- L'Ecuyer P., L. Meliani, J. Vaucher. 2002 *SSJ: A framework for stochastic simulation in Java*. Conference paper Winter Sim'02 conference.
- Garrido J.M., 2001 *Object-Oriented Discrete-Event Simulation with Java*, Kluwer Academic/Plenum Publishers, New York.
- Healy K.J. and Kilgore R.A.,1998 *Introduction to silk and java-based simulation*.
- Jacobs P.H.M., N.A. Lang, A. Verbraeck. 2002. *DSOL; A distributed Java based discrete event simulation architecture*. Conference paper Winter Sim'02 conference.
- Law A.M., W.D. Kelton. 2000. *Simulation modeling and analysis*, 3th ed., McGraw-Hill, New York
- Matsumoto M., T. Nishimura, *Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator*, ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, January pp.3-30 1998
- Shannon R.E., 1975. *Systems simulation: the art and science*, Prentice-Hall
- Simon H.A., 1973. *The structure of ill-structured problems* in *Artificial Intelligence*, vol. 4 , pages 181-202
- Sol H.G., . 1982. *Simulation in information system development*.
- Sun Microsystems Inc., 2002 *Why Are Thread.stop, Thread.suspend, Thread.resume and Runtime.runFinalizersOnExit Deprecated?*, <http://java.sun.com/j2se/1.4.1/docs/guide/misc/threadPrimitiveDeprecation.html>
- Vangheluwe H. and J. de Lara, 2002. *Meta-models are models too* in *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J.L. Snowdon and J.M. Charnes, ed., pages 597 - 605.
- Zeigler B.P. , H. Praehofer and T.G. Kim. 2000. *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2d ed. San Diego: Academic Press.

## AUTHOR BIOGRAPHIES

**NIELS A. LANG** is a Phd. student at Erasmus University Rotterdam. He researches simulation in logistic system design, with an emphasis on economic analysis. His e-mail address is <nlang@fbk.eur.nl>.

**PETER H.M. JACOBS** is a PhD. student at Delft University of Technology. His research focuses on the design of decision support services for the design and management of a business alliance portfolio. His e-mail address is <p.h.m.jacobs@tbm.tudelft.nl>.

**ALEXANDER VERBRAECK** is an associate professor in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology, and a part-time full professor in supply chain management at the R.H. Smith School of Business of the University of Maryland. He is a specialist in discrete event simulation for real-time control of complex transportation systems and for modeling business systems. His current research focus is on development of open and generic libraries of object oriented simulation building blocks in Java. Contact information: <a.verbraeck@tbm.tudelft.nl>.

**SIMULATION IN  
BUSINESS, ECONOMY,  
FINANCE AND  
COMMERCE**



# **BUSINESS MODELING FOR NON-MODELING EXPERTS SIMULATION AND VISUALIZATION AT THE AMSTERDAM MUNICIPAL POLICE FORCE**

Mariëlle den Hengst and Hermen Geerts  
Delft University of Technology, Faculty of Technology, Policy and Management  
PO Box 5015; 2600 GA Delft; the Netherlands  
phone: +31 15 278 8542; fax: +31 15 278 3429  
email: [M.den.Hengst@tbm.tudelft.nl](mailto:M.den.Hengst@tbm.tudelft.nl)

## **KEYWORDS**

Simulation, visualization, decision making, non-modeling experts

## **ABSTRACT**

Business modeling is increasingly being used as supporting tool in taking important decisions, sometimes even at a frequent base. This requires that non-modeling experts are able to work with the model. Most static and dynamic modeling methods, however, are too complex for non-modeling experts to work with. The combination of visualization and simulation offers a promising means for making business models accessible to non-modeling-experts for their decision making. A simulation model of the Amsterdam Municipal Police Force was built to support managers in making decisions. Three visual modules were developed to allow non-modeling experts to work with the model without going into much detail.

## **1. INTRODUCTION**

Decision Support Systems (DSS) as a field had appeared in the 70s. Traditional DSS were based on fundamental technologies, allowed limited communication and followed a rational approach. Most early DSS focused on presenting financial numbers to decision makers. Today's tools and technologies develop at a high rate and allow for a sophisticated support environment including simulation tools, information visualization technologies, and collaborative technologies. These developments allow decision makers not to just extract numbers and do useful calculations with them as in the traditional DSS, but also to use models to do 'what-if' analyses.

Business modeling is increasingly being used as supporting tool in taking important decisions, sometimes even at a frequent base. This requires that non-modeling experts are able to work with the model (Sterman 2000, Vreede 1997). One prerequisite for this is that the models are easy to communicate. A model, therefore, must closely resemble the mental model of the persons involved (Checkland 1981). Vreede and Verbraeck (1996) show that traditional –static– diagramming techniques, such as Entity Relationship Diagrams, Data Flow Diagrams, or SADT models do not meet this requirement. Dynamic modeling methods

offer wider opportunities for understanding business processes and to analyze the process dynamics (Paul et al. 1998). Simulation is especially valuable for the evaluation of different alternatives as well as for providing statistical evidence to convince actors of the efficiency and effectiveness of a particular organizational system. However, a simulation model does not automatically resemble the mental models of the non-modeling experts. Using visualization on top of the simulation models has the potential of overcoming this problem (Vreede and Verbraeck 1996). Visualization is an essential supporting component for gaining insights and relaying knowledge (Wenzel and Jessen 2001).

Visualization offers one of the most promising means to convey information from a simulation model to decision makers in a meaningful way (Macal 2001). The goal of visualization is dependent on the phases of a simulation study and the respective target groups, for example simulation experts and decision makers (Wenzel and Jessen 2001). In this paper we look at the added value of visualization for decision makers in the different phases of a simulation study.

In the remainder of this paper, section 2 presents more background on decision making, simulation, and visualization. We present a framework grounded in literature that enables us to analyze the added value of visualization for decision makers. Then, in section 3 we use this framework in a case study. A case study was carried out at the Amsterdam Municipal Police Force to show the opportunities of visualization and simulation to allow non-modeling experts to work with business models for their decision making. In section 4, the results of the case study are presented and discussed. The paper concludes with a discussion of the findings of the study and an identification of some issues for further research.

## **2. BACKGROUND**

Decision making is closely related to problem solving. Ackoff's (1981) definition of solving problems requires decisions to be made: "By a problem we mean a situation that satisfies three conditions. First, a decision making individual or group has alternative courses of action available. Second the choice made can have a significant effect. And, third, the decision maker has some doubt as to which alternative should be selected."

Simon et al. (1987) described the work of making decisions and solving problems as work of choosing issues that require attention, setting goals, finding or designing suitable courses of action, and evaluating and choosing among alternative actions.

Simulation is a problem solving approach and shows many similarities with the notions on problem solving and decision making presented above. It can be used to support decision making on complex systems. Various approaches exist to conduct a simulation study. A well known approach is described by Banks et al. (2000). They distinguish the following steps: (1) problem formulation, (2) setting objectives and overall project plan, (3) model conceptualization, (4) data collection, (5) model translation, (6) verification, (7) validation, (8) experimental design, (9) runs and analysis, (10) more runs?, (11) reporting, and (12) implementation. These steps are depicted in figure 1 and described in more detail below. In contrast to most diagrams, the diagram presented in figure 1 does not put the steps on the forefront. The diagram puts main emphasis on the products resulting from the steps, since, when we talk about visualization, we talk about visualization of the products and not of the steps.

- A. After the problem has been formulated (1), and the objectives are set (2), the problem situation in an organization is conceptualized (3) in order to structure the problem situation in such a way that the efforts for detailed, low-level data gathering for creating the empirical model can be focused and minimized.
- B. Next, a descriptive empirical model is built that can be used to analyze and diagnose the problem situation. For this purpose, data about the problem situation is collected (4), and the model is implemented in a simulation language (5). Before the model can be used, it must be checked whether it is a good representation of the problem situation.

First, the model is verified to ensure that it behaves as intended (6). Then, the model is validated to test the (statistical) correspondence between the model and the problem situation (7). If this check is passed, the empirical model can be used to identify causes and effects of the problem.

- C. Based upon the results from the problem diagnosis, several alternative solutions may be generated. The alternative solutions are worked out in detail in a number of prescriptive empirical models (8). These models can be experimented with in order to study the effects of the alternatives in more detail (9). When the solutions are not satisfying enough, new alternatives can be constructed to run experiments with (10).
- D. The actual choice is made based on the results of the experiments among others. This may involve a combination of possible solutions, or leaving the situation as it is. The results of the analyses are reported about (11).
- E. Finally, in order to actually solve the problem situation, the solution must be implemented (12).

In a decision making process that is based on simulation, we can distinguish between several roles that communicate different knowledge about the problem simulation in different steps of a simulation study. Kuljis et al. (2001) distinguish between two roles: the analyst and the user. The analyst builds the model and works with it, the user uses the model to experiment with 'what-if' scenarios. The use of the model, however, is restricted to the use of the outcome, which is collected and presented with the assistance of a simulation expert (Kuljis et al. 2001). Vreede and Verbraeck (1996) further elaborate the role of user into the role of problem owner and the role of decision maker. The problem owner is confronted with a problem situation, but often does not have the authority

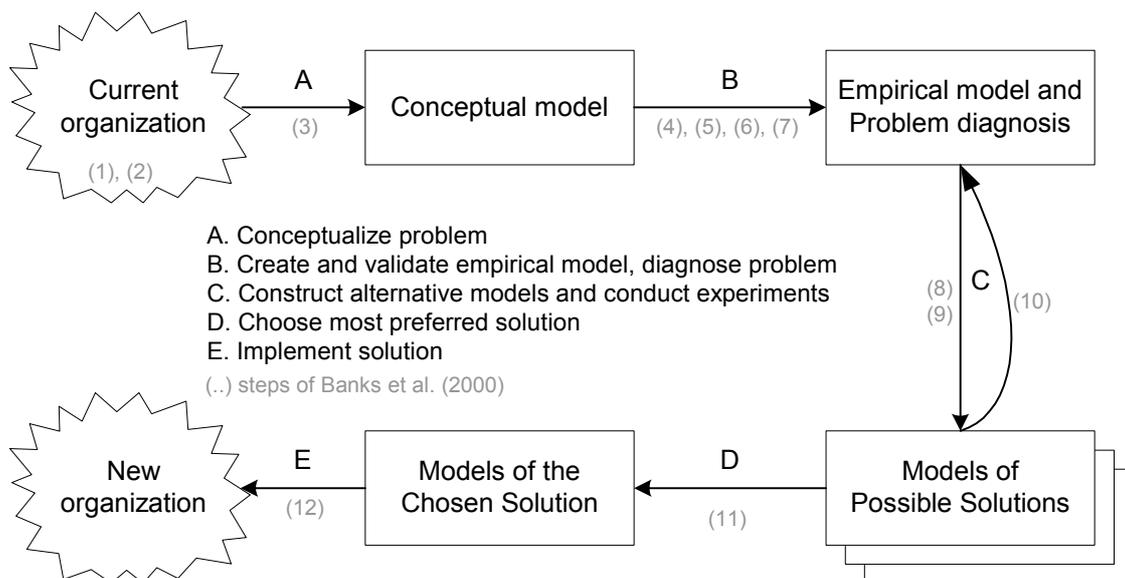


Figure 1: Approach for Decision Making in which Simulation Models are Used

to decide on changes with respect to the problem situation. The decision maker does have this authority. We distinguish between three roles: analysts, problem owners, and decision makers.

The three roles exchange information in each of the steps of a simulation study. Problem owners and decision makers communicate with each other to formulate the problem. Then the analysts enter the picture and they start to communicate with the problem owners to set the objectives, to conceptualize the problem, and to collect the data. The analysts then build the model. The problem owners and decision makers together with the analysts discuss the results of the simulation model and design alternatives. The analysts build the models for the experiments and the results are presented to the problem owners and decision makers. Finally, the decision makers decide on which alternative to implement.

In this paper we focus on the communication between the analyst on the one hand and the decision maker and the problem owner on the other hand. The analyst functions as a human interface between the simulation model and the problem owners and the decision makers. This interface would not have been necessary if the modeling knowledge of the analyst is incorporated in either the simulation tools or the problem owner and the decision maker. The latter is not an alternative, since problem owners and decision makers usually lack the modeling knowledge to complete a simulation study. Visualization has been seen as a way to support and simplify communication (Macal 2001, Vreede and Verbraeck 1996, Wenzel and Jessen 2001). Visualization as such enables to decrease the role of analyst as interface between the model and the problem owner and the decision maker. Modeling knowledge is increasingly incorporated in the simulation tool through visualization.

The use of visualization in simulation has been discussed for more than ten years. A simulation advances the state of a modeled system through time, and a visualization provides an abstract visual rendering of the state of that system at any point in time (Macal 2001). Pegden et al. (1990) state that a visualization, or animation as they call it, consists of a static part and a dynamic part. The dynamic part depicts status-changes in the simulation model and the static background represents the environment in which the simulated system exists. We follow Vreede and Verbraeck (1996) in stating that the static part represents more than the problem situation alone. It may also contain elements not present in the problem situation, like text, or static-artifacts of the simulation model, like the values of the exogenous variables. Visualization, however, encompasses more than the animation as described above. Animation depicts the results of a simulation run and puts main emphasis on the dynamic character. Visualization can also be used in the phases preceding and following the running of the model. During model construction, graphical interfaces can be used to build the model. And during the analysis of the results,

statistical output of the model runs can be used for evaluation purposes.

Visualization has added value, but also limitations in enhancing communication between analysts, problem owners, and decision makers in many different ways. The enumeration below shows the limitations and advantages of visualization for each step in the simulation study with regard to communication between the different roles. These notions are summarized in the framework presented in table 1.

A. A conceptual model is a static description of the elements in a problem situation and the relationships between these elements. Two important activities must take place to build a conceptual model. First, the problem should be identified and demarcated. Second, the problem situation should be translated into a conceptual model. The problem owner and decision maker provide the analyst with information, mostly in the form of oral or written text, to carry out these activities. The analyst usually reports back with a conceptual model.

B. The analyst collects data to quantify the elements and relationships between these elements in the conceptual model. The problem owner provides this data, mostly in the form of text. The analyst then builds the simulation model. Graphical interfaces for building simulations have been developed. These visual tools are expressive enough to be used to assemble complete simulation models and to specify alternative simulation runs (Ozden 1991). Most of these tools, however, still require the user to have simulation expertise (Kuljis et al. 2001).

Before the model can be used, it must be verified and validated. Animations can be used as a verification aid for creating models. Animation models may decrease the development time of the simulation model, because of the enhanced debugging possibilities. With animation it might be quicker to find and localize mistakes than by going through output of traces or using a debugger (Vreede and Verbraeck 1996). However, a correctly functioning animation does not imply a completely debugged model, much less a verified model (Johnson and Poorte 1988).

Animations can be helpful in the validation of the simulation models as well. Because of the 'cartoonlike' behavior, animations have the potential to resemble closely the mental models of the problem owners involved. Hence, it is easier to communicate to problem owners. As a result, animations offer unique possibilities for face-validity tests. Structural mistakes in the model or deviant model behavior can be pointed out by problem owners. Animations, however, can only be used to show that a simulation model is not valid (Law and Kelton 1991). Furthermore, animations do not support the 'statistical' validation that is required besides the face-validity.

If the model is verified and validated, it can be used to identify causes and effects of the problem.

Animations can enhance the problem diagnosis. Animations provide more insight into deadlock situations, system bottlenecks, queue lengths, and so on. Furthermore, an animation can illustrate the statistical results of a process analysis in an accessible way. There is no need for decision makers to go through large amounts of numerical data afterwards (Vreede and Verbraeck 1996). Animations can even be considered a better way to illustrate results to decision makers: seeing it is believing it (McHaney 1991). Some reserve, however, must be taken into consideration, since snapshots of a running visual simulation are a dangerous yardstick to determine what is going on in the system over time (Grant and Weiner 1986, Paul 1991). The statistical output after running several replications with the model should be used as well to come to firmly grounded and statistically sound conclusions.

- C. Based upon the results from the problem diagnosis, several alternative solutions may be generated. The problem owner and decision maker discuss these solutions with the analyst. The analyst translates these into alternative simulation models. The changes to the simulation model can take place at two levels: changing the structure (the elements and the relationships between these elements) and changing the data (the parameters in the model). The analyst can use a graphical interface to make those changes.

Once the alternative models are finished, they should be run to analyze the results. The same visualization means can be used as with the problem diagnosis. An extra dimension, however, can be added for conducting experiments. In diagnosing the problem, one situation must be analyzed, but in conducting experiments, different situations should

be compared with each other. An animation allows to show the results of one situation only. The statistical output of the different alternatives should be presented next to each other to enable comparison between the alternatives.

- D. The results of the simulation study are reported about. This usually is a document prepared by the analyst and used by the problem owner and the decision maker. The decision maker decides based on the results reported.
- E. Finally, the solution chosen is implemented. This step is often outside the scope of the simulation study.

Concluding from this, it can be stated that visualization is used in many different ways to enhance the communication between the analyst and the problem owner and the decision maker. It is also noted that, despite the enhanced communication, the analyst still functions as an interface between the model and the decision maker and the problem owner. To enable non-modeling experts to really work with simulation models in all steps of a simulation study requires more advanced visualization means than currently used. We distinguish three visualization modules: a visual input module, a visual run module, and a visual output module. The visual input module should enable the non-modeling expert to enter information on the structure and data of the problem situation to build the simulation model. Graphical interfaces to enter this information already exist. Interfaces to enter information on the structure, however, often require simulation expertise. Interfaces to enter the data, on the other hand, are open to non-modeling experts, especially when the choices are prestructured. The visual run module already exists in the form of animations. As already noted, animations make the simulation model more accessible to non-

Step	Visualization means	Visualization topic	From	To
(1) (2) problem demarcation	text text	different aspects of the problem	po/dm an	an po/dm
(3) model conceptualization	text conceptual model	elements and relationships between the elements	po an	an po
(4) data collection	text	numerical data on the elements and relationships	po	an
(5) model translation	simulation code graphical interface	structure + data	an	
(6) verification	animation	structure + data	an	
(7) validation	animation	structure + data	an	po
problem diagnosis	animation	structure + data	an	po/dm
(8) experimental design	text graphical interface	structure + data	po/dm	an
(9) runs and analysis	animation statistical output	structure + data + output	an	po/dm
(11) report	text	structure + data + output	an	po/dm
(12) implementation	-	-	-	-

po = problem owner; dm = decision maker; an = analyst

Table 1: Visualization to enhance communication in simulation studies

modeling experts, but they present snapshots only. To get a complete picture of the problem situation modeled, the visual output module is required. The visual output module presents the statistically sound results of the simulation model after several replications. Although each of the three modules already exists in some way, it is not clear whether they are sufficient to enable the non-modeling expert to work with the simulation model without the support of the analyst. Bell et al. (1990) state that great challenges exist to build visual simulation tools to support decision makers. In the next section we take up this challenge in a case study. In the case study we developed a simulation model to be used by non-modeling experts without the support of analysts.

### 3. THE RCCT STUDY APPROACH

The Regional Collecting Controlling and Tracing (RCCT) department of the Amsterdam Municipal Police Force processes the charges of civilians, for example for speeding, ignoring red light, and parking wrongly. About 5600 police officers, divided over eight different districts in the region of Amsterdam, observe violations of rules by civilians. These violations are summarized in a charge. These charges have to be processed in order to be sure that the civilian pays the fine. The managers of the RCCT are in search for support in deciding how these processes should be carried out. Several reasons for this can be indicated.

- First and most important, the Amsterdam Municipal Police Force wants to cut down the processing times for the charges.
- Second, the managers want a tool to analyze the possibilities of different ICT applications to speed up the process. Personal handhelds, for example, could be used by the police officers to summarize the violations. This would replace the written charges and could result in less errors due to unreadable charges and incomplete charges. Another ICT application could be the automatic recognition of license plates on photographs taken of cars violating traffic rules.
- Third, the RCCT must deal with a dynamic environment, requiring the RCCT to adapt their processes to the changes in the environment. One of the most important influences is the amount of charges entering the RCCT. The amount of charges that have to be processed is very fluctuating. Some examples are described to show this fluctuation. A first example is that the number of charges increases at the end of the month, because police officers have to meet targets on the number of charges they have issued. Another example is the formulation of projects by the Amsterdam Municipal Police Force to attract special attention of civilians to certain areas. Speeding, for example, might be a special topic for a month, resulting in more charges on speeding that month. These fluctuations have an unpredictable nature. Another example is the reorganization of processes: the issuing of charges

for parking at the wrong spots was delegated to parking officers, but will be added to the tasks of police officers. This means an increase of charges for the RCCT.

To deal with the situations mentioned above, the RCCT managers need an automated tool for designing the processes to meet the expected flow of charges. This tool must be able to simulate the flows through the organization over time. The tool must have a good representation of the processes within the organization to be able to predict the performance. Ultimately, the tool must allow the managers to define 'what if' scenarios. Two major goals for the system are to provide a visual development environment and to display the results in an easy to understand visualization.

In the case study, a dynamic model has been built in Arena®, a dynamic modeling language allowing for the combination of simulation and animation. A conceptual model has been built by analyzing the processes and interviewing employees. The data for the simulation model was collected by using existing databases and by measuring several indicators of the processes, such as processing times. With the conceptual model and the data, the simulation model was built. The model was verified and validated and after some slight modifications a good simulation model resulted.

The simulation model was combined with visualization through the three different modules. First, an animation of the simulation model has been built. The animation shows the flow of the charges through the organization, shows the number of employees working on the charges and shows the different queues of the charges. Figure 2 shows a screendump of the animation. The animation shows the seven different divisions within RCCT as well as the work load for each division.

Second, a visual input module for changing the simulation model has been developed. The visual input module does not offer possibilities to change the structure of the simulation model; only the data used in the simulation model can be changed. The visual input module has been realized using VBA (Visual Basic for Applications) and is connected dynamically to the simulation model. This means that even during a simulation run changes to the model can be made, instead of only after a simulation run. Interaction with the simulation model is possible. An interface has been designed in which the managers can change the number of employees working at each process, the processing times, the number of charges entering the RCCT, and so on. Figure 3 shows a screendump of the visual input module. The visual input module contains several of these sheets to change all data relevant.

And third, a visual output module for presenting the statistical output of the simulation model has been built. Relevant information to managers on, for example, throughput times, queue lengths, and occupation of employees are presented graphically over time so an analysis of the performance of the system can be made



Figure 2: Animation of the RCCT

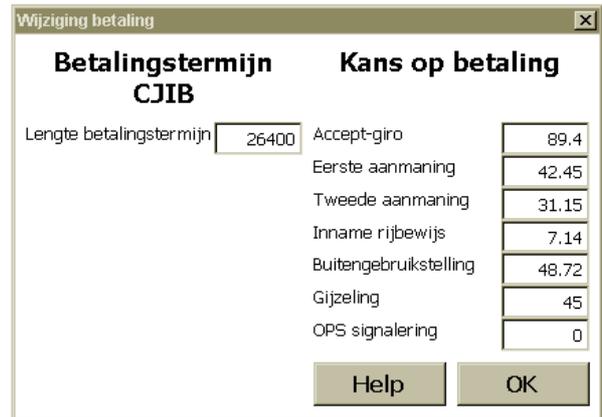


Figure 3: Visual Change Module of the RCCT-Case

by the decision makers.

The visual output module has been realized in Microsoft® Excel. The visual output module only reads in the results of a simulation run after the run has finished. The status of the system during the simulation run is visible through the animation, but a thorough analysis of the results is possible through the visual output module. Figure 4 shows a screendump of the visual output module.

In order to evaluate whether the visualization elements indeed allow non-modeling experts to use simulation models in their decision making, the simulation model and visualization modules were presented to the managers of the RCCT. After the presentation, a questionnaire was filled in by the managers to get a

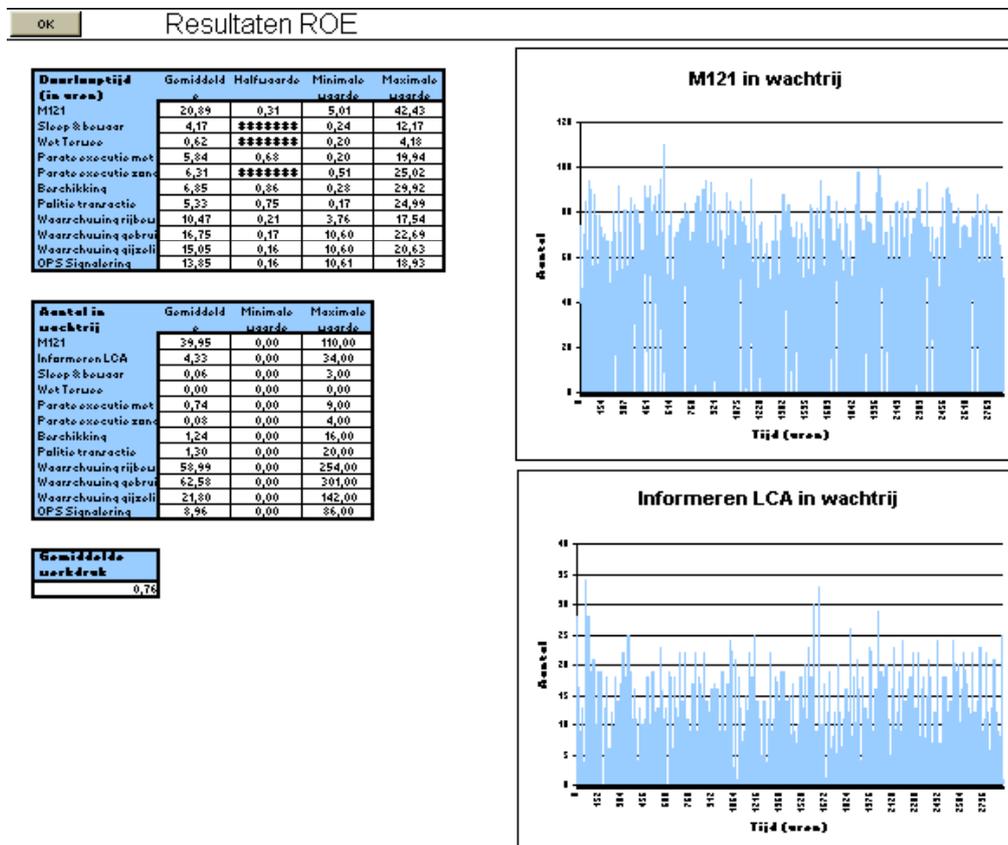


Figure 4: Visual Output Module of the RCCT-Case

feeling of the usefulness of the visualization elements. The questionnaire as well as the results are presented in the next section.

#### 4. RESULTS OF THE RCCT STUDY

The visual modules were evaluated using a questionnaire. In the questionnaire, the managers could indicate how they evaluate the different modules on different criteria. Each module is discussed in a separate paragraph, starting with the animation. The design of the interface is generally considered to be good. One manager indicated that the design was bad. Compared to many computer games available today, the design indeed looks bad, with robotic people sitting behind a desk, but for the purpose of the animation, most managers agree about the design to be good. The arrangement and the clarity of the animation elements are considered to be good, as well as the completeness. Furthermore, the animation is considered to be user-friendly.

Table 2: Results on the Animation (n = 10, 1 is very bad, 5 is very good)

Criteria	1	2	3	4	5	$\mu$	$\sigma$
Design	0	1	0	6	3	4,1	0,88
Clear arrangement	0	0	0	9	2	4,2	0,42
Clarity	0	0	0	8	1	4,1	0,32
Completeness	0	0	0	7	3	4,3	0,48
User-friendliness	0	0	1	7	2	4,1	0,57

The results for the visual input module are presented in the table below. Especially the design is considered to be very good. The elements are arranged in a clear manner and the clarity of the elements is considered to be good as well. With regard to completeness and user-friendliness, the overall opinion is that the visual input module is good. However, some managers are not yet really convinced of this, since they have a neutral opinion.

Table 3: Results on the Visual Input Module (n = 10, 1 is very bad, 5 is very good)

Criteria	1	2	3	4	5	$\mu$	$\sigma$
Design	0	0	0	4	6	4,6	0,52
Clear arrangement	0	0	1	8	1	4,0	0,47
Clarity	0	0	0	6	2	4,0	0,67
Completeness	0	0	2	5	5	4,5	0,53
User-friendliness	0	0	3	6	1	3,8	0,63

Finally, the visual output module is evaluated, the results of which are presented in Table 4. The overall opinion is that the visual output module is good. However, the standard deviation is slightly higher than for the other modules. This is true especially for the clarity and the user-friendliness of the module. The enormous amount of statistical output will partly be due to this.

Table 4: Results on the Visual Output Module (n = 10, 1 is very bad, 5 is very good)

Criteria	1	2	3	4	5	$\mu$	$\Sigma$
Design	0	0	2	3	5	4,3	0,82
Clear arrangement	0	0	1	4	5	4,4	0,70
Clarity	0	1	2	4	3	3,9	0,99
Completeness	0	0	0	6	4	4,4	0,52
User-friendliness	0	2	1	5	2	3,7	1,06

The questionnaire filled in by the managers was concluded with the overall question, whether they believed that the tool is usable in their daily practice. With no exception, they all agreed that the tool is usable in their decision making processes.

#### 5. CONCLUSIONS AND FUTURE RESEARCH

The case study described in this paper showed that a simulation model can be used by non simulation experts by adding visual elements to the simulation model. Prior case studies in which a simulation model and visualization were combined, show that usually the analyst still fulfills the role of human interface between the model and the non-modeling expert. From this, it can be concluded that the visual input module and the visual output module have added value for decision makers. More case studies and experiments, however, should be carried out to give more depth to this conclusion.

The visual modules used in the case study, however, only support decision makers in a part of the decision making process. Problem diagnosis, construction of alternatives, and experimentation are activities that are supported by the visualization. The construction of alternatives is only partly supported, since the decision maker can only change the data of the simulation model and not the structure of the simulation model. Furthermore, the activities concerned with building, verifying and validating the simulation model are not supported well enough yet by visual elements to allow the non-modeling experts to work without the support of the analyst. These activities still must be carried out by a simulation specialist. Further research should focus on ways to expand the visual support to other activities as well.

During the construction of the simulation model and the visual modules, a trade-off existed between the flexibility and power of the simulation model, and the user-friendliness of the visual modules for non-modeling experts. One of the reasons for this trade-off is that most simulation languages are focused on offering the opportunity to build a simulation model. What simulation languages should focus on more and more is the opportunity to build a Decision Support System based on a simulation model.

#### REFERENCES

- Ackoff, R.L. 1981. 'The Art and Science of Mess Management', in: *TIMS Interfaces*, vol. 11, no.1, 20-26  
 Banks, J., J.S. Carson, B.L. Nelson, and D.M. Nicol. 2000. *Discrete-Event System Simulation*, Upper Saddle River, New

York: Prentice Hall

Bell, P.C., A.A. Taseen, and P.F. KirkPatrick. 1990. 'Visual Interactive Simulation Modeling in a Decision Support Role', in: *Computers & Operations Research*, vol. 17, no. 5, 447-456

Checkland, P.B. 1981. *Systems Thinking, Systems Practice*. Chichester: John Wiley & Sons

Grant, J.W. and S.A. Weiner. 1986. 'Factors to Consider in Choosing a Graphically Animated' *Industrial Engineering*, vol. 18, no. 8, 36-68

Johnson, M.E. and J.P. Poorte. 1988. 'A Hierarchical Approach to Computer Animation in Simulation', *Simulation*, vol. 50, no. 1, 30-36

Kuljis, J., R.J. Paul, and C. Chen. 2001. 'Visualization and Simulation: Two Sides of the Same Coin', *Simulation*, vol. 77, no. 3-4, 41-152

Law, A.M. and W.D. Kelton. 1991. *Simulation Modeling and Analysis*, New York: McGraw-Hill

Macal, C.M. 2001. 'Introduction to Special Issue on Simulation and Visualization', in: *Simulation*, vol. 77, no. 3-4, 90-92

McHaney, R. 1991. *Computer Simulation. A Practical Perspective*, San Diego, California: Academic Press Inc.

Ozden, O. 1991. 'Graphical Programming of Simulation Models', in: *Simulation*, vol. 56, no. 2, 104-116

Paul, R.J. 1991. 'Recent Developments in Simulation Modelling', in: *Journal of the Operational Research Society*, vol. 42, no. 3, 217-226

Paul, R.J., V. Hlupic, and G. Giaglis. 1998. 'Simulation Modeling of Business Processes', in: D. Avison and D. Edgar-Neville (eds.), *Proceedings of the 3rd U.K. Academy of Information Systems Conference*, Lincoln, U.K.: McGraw-Hill

Pegden, C.D., R.E. Shannon, and R.P. Sadowski. 1990. *Introduction to Simulation Using SIMAN*, McGraw-Hill

Simon, H.A., G.B. Dantzig, R. Hogart, C.R. Plott, H. Raiffa, T.C. Schelling, K.A. Shepsle, R. Thaler, A. Tversky, and S. Winter. 1987. 'Decision Making and Problem Solving', in: *TIMS Interfaces*, vol. 17, no. 5, 11-31

Sterman, J.D. 2000. *Business Dynamics. Systems Thinking and Modeling for a Complex World*, USA: McGraw-Hill

Vreede, G.J. de. 1997-1998. 'Collaborative Business Engineering with Animated Electronic Meetings', in: *Journal of MIS*, vol. 14, no. 3, 141-164

Vreede, G.J. de and A. Verbraeck. 1996. 'Animating Organizational Processes: Insight Eases Change', in: *Journal of Simulation Practice and Theory*, no. 4, 245-263

Wenzel, S. and U. Jessen. 2001. 'The Integration of 3-D Visualization into the Simulation-Based Planning Process of Logistics Systems', in: *Simulation*, vol. 77, no. 3-4, 114-127

Coordination in Container Transport; A Chain Management Design" in 1999. She has presented her work at a number of national and international conferences.

HERMEN GEERTS studied Systems Engineering, Policy Analysis, and Management at Delft University of Technology in the Netherlands. His final thesis project was with the Amsterdam Police Force to investigate the added value of dynamic modeling in their decision-making processes.

## BIOGRAPHIES



MARIËLLE DEN HENGST (m.den.hengst@tpm.tudelft.nl) is an assistant professor in the department of systems engineering of the Faculty of Technology, Policy and Management at Delft University of Technology in the Netherlands. Her research interests include collaborative systems, decision support, and

business modeling: collaborative business engineering. She obtained her Ph.D. from Delft University of Technology on the subject of "Interorganizational

# EMERGENCE OF SELF ORGANIZATION AND SEARCH FOR OPTIMAL ENTERPRISE STRUCTURE: AI EVOLUTIONARY METHODS APPLIED TO AGENT BASED PROCESS SIMULATION

Marco Remondino  
Department of Computer Science  
University of Turin  
C.so Svizzera 185  
10149 Turin, Italy  
E-mail: remond@di.unito.it

## KEYWORDS

simulation, model, intelligent agent, genetic algorithm, classifier system, complex behaviour, process

## ABSTRACT

Enterprise simulation allows what-if analysis and helps in business process re-engineering. There are mainly two approaches to simulation: process based, which is strictly deterministic and generally used to model well known parts of enterprises or mechanical/electronic systems and agent based, which allows to study the emergence of aggregate behaviour, through the creation of models, known as artificial societies. In order to simulate enterprises where the environment and human factor are relevant, a hybrid formalism is proposed: Agent Based Process Simulation. Colonies of intelligent agents, modelled using evolutionary methods derived from the AI field, are put side by side with the representation of processes, modelled as symbolic and deterministic agents built using paradigms derived from Propositional and Modal Logic. The main goal of this work is to study how this approach can allow the emergence of aggregate behaviour and, by using some performance parameters, can help to find the optimal organization for an enterprise. In particular, agents built using Genetic Algorithms and Classifier Systems can evolve to find local maximum of functions representing situations whose rules are not entirely known a priori, such as the ones behind the optimal organization of an enterprise.

## INTRODUCTION

In (Ostrom 1988), simulation is described as a third way to represent social models, being a powerful alternative to other two symbol systems: the verbal argumentation and the mathematical one. The former, which uses natural language, is a non computable way of modelling though a highly descriptive one; in the

latter, while everything can be done with equations, the complexity of differential systems rises exponentially as the complexity of behaviour grows, so that describing complex individual behaviour with equations often becomes an intractable task. Simulation has some advantages over the other two: it can easily be run on a computer, through a program or a particular tool; besides it has a highly descriptive power, since it is usually built using a high level computer language, and, with few efforts, can even represent non-linear relationships, which are tough problems for the mathematical approach. According to (Gilbert, Terna 1999):

*“The logic of developing models using computer simulation is not very different from the logic used for the more familiar statistical models. In either case, there is some phenomenon that the researchers want to understand better, that is the target, and so a model is built, through a theoretically motivated process of abstraction. The model can be a set of mathematical equations, a statistical equation, such as a regression equation, or a computer program. The behaviour of the model is then observed, and compared with observations of the real world; this is used as evidence in favour of the validity of the model or its rejection”*

Computer programs can be used to model either quantitative theories or qualitative ones; simulation has been successfully applied to many fields, and in particular to social sciences, where it allows to verify theories and create virtual societies. In particular, the simulation of an enterprise can give very good results, regarding case studies, what-if analysis and business process re-engineering. It is possible to identify two different approaches to computer simulation, both of which lead to the creation of a computational model of a social or complex system, starting from a very different point: Process Simulation and Agent Based Simulation. Both of them can be used to model enterprises or firms, but with some fundamental differences, which will be discussed in detail. Agent Based Modeling is the most interesting and advanced approach for simulating a complex system: in a social context, the single parts and the whole are often very

hard to describe in detail. For this reason, process simulation is not the ideal tool to model these complex environments; besides, there are agent based formalisms which allow to study the emergency of social behaviour with the creation and study of models, known as artificial societies. Thanks to the ever increasing computational power, it's been possible to use such models to create software, based on intelligent agents, which aggregate behaviour is complex and difficult to predict, and can be used in open and distributed systems. A software agent can be described as a flexible system, capable of dynamic, autonomous actions, in order to meet its design objectives, that is situated in some environment. The main features for a software agent are: situatedness, that is ability to perform actions according to a particular input received from outside, which can, in turn, change the environment itself; autonomy in performing actions, without intervention of humans; flexibility and adaptability. Some particular agents can also be proactive, which means they are goal-directed, and social, in the way they can interact with other artificial agents, robots, and humans. Such an intelligent agent can be referred to as a Belief-Desire-Intention (BDI) one. There are many agent based paradigms that can be applied to computer simulation:

- *Symbolic*: highly structured agents, described through expressions of Propositional and Modal Logic. This is perfect when there is a single agent, which must interact with the environment, but it's not versatile when used to simulate big communities
- *Sub-symbolic*: simple agents, which can be described through metaphors. A multi-agent context of this kind allows the emergency of complex behaviour and self-organization. Intelligent behaviour is a product of the interaction among agents and environment, and of the interaction among many simple behaviours. It can be really hard to describe the real world under every aspect: some fundamental macro-actions can thus be defined on single agents, which allow cooperation with the environment and with other agents. The concept of *Multi Agent System* for social simulations is thus introduced: the single agents have a very simple structure. Only few details and actions are described for the entities: the behaviour of the whole system is a consequence of those of the single agents, but it's not necessarily the sum of them. This can bring to unpredictable results, when the simulated system is studied.
- *Hybrid Architectures*: at the lower levels, we find reactive agents, like the ones described above, while at the upper levels there are more complex and structured agents. In this way, we can combine reactive capabilities with planning.

The approach proposed in this work can be considered as a hybrid architecture, since it uses symbolic agents as process based blocks, and sub-symbolic agents to model those parts of an enterprise which are not fully known, part of the environment, or even the human beings involved in the organization.

## **SIMULATION: TWO DIFFERENT APPROACHES**

Both process simulation and agent based simulation are powerful approaches for creating models of enterprises and complex systems, but they also have some flaws. In order to overcome the limits of both the simulation approaches, the possibility of a hybrid methodology is studied in (Remondino, 2003). In the present work I'll concentrate the discussion on enterprise simulation and I'll discuss how intelligent agents, based on AI paradigms, in particular genetic algorithms and classifier systems, can show an emergent aggregate behaviour when put side by side with formal and deterministic processes. While deeply describing both the approaches is beyond the purpose of this paper, I'll analyze the main differences among them, which will lead to the hybrid formalism that I'm studying. Usually, process simulation is used to model a very well structured and known situation, in order to perform a what-if analysis: it's used to create models of parts of enterprises or mechanical/electronic systems. Its greatest advantage is that it starts from a basic scheme, often derived from existent documents, through which it becomes very easy to bring a real situation into a process simulator: usually, a model to be used for process simulation looks like a flow chart, in which a token passes from one box to another one, in a deterministic way, on the basis of the given rules. This kind of approach is widely spread and allows to deeply analyze a part of a whole, studying the expected behaviour of a system, when some change is operated. This is why process simulation is a great support to decisions; the simulator can answer to many questions and what-if problems, that would require big efforts in the real environment; for example, a part of a manufacturing plant can be simulated, by dividing it into its main processes, and then it will be possible to check what would happen on the final output if some change occurs. According to (Helsgaun, 2000), the process based approach, when building deterministic simulations, is alternative to the *event based* and the *activity based* ones. In the former the model consists of a collection of events and each event models a state change and is responsible for scheduling other events that depend on that event. Each event has associated an event time and some actions to be executed when the event occurs. In the activity based approach the model consists of a collection of activities: each activity models some time-consuming action performed by an entity. Each activity has associated a starting condition, some actions to be executed when the activity starts, the duration of the activity, and some actions to be executed when the activity finishes. In the process based approach the model consists of a collection of processes. Each process models the life cycle of an entity and is a sequence of logically related activities ordered in time. Since processes resemble objects in the real world, process based simulation is often easy to understand; implementation, however, is not always easy and execution efficiency may be poor if the implementation is not done properly.

Unfortunately there isn't a universal modelling language for process simulation and this often requires deep translations for the models to be ported from one tool to another. Another disadvantage is that, in order to use this approach to simulate a process, this must be very well known; if a part of the process is uncertain, then it's impossible to validate a simulation as a model of the real world to be represented. Besides, this method is quite static, meaning that the relations between the various parts involved in the model must be described in deep and there is no possibility of emergent behaviour and self-organization.

When the system to be simulated has a complex aggregate behaviour, not easy to describe just studying and modelling the single entities, agent based simulation is the only usable approach. In complex systems the sum of the parts is often not enough to describe the whole, and usually from the interaction of many simple entities a complex behaviour emerges. So, if we want to model an enterprise in which also the human factor is present, or we want to consider also the influence of the environment, it will be impossible to do that with a process based approach, thus leaving agent based simulation as the only feasible method. While in process simulation the stress is on the function of the single parts, that are deeply modelled as resembling the reality, in agent based simulation the most important side is interaction among entities, which creates the aggregate behaviour. The single agents can even be very simple, with few rules and directives. For example, an artificial stock market can be simulated by creating some different types of intelligent agents, which follow inner rules; some of them can simply act randomly, while others will "study" the trend before acting. Some of them, on the contrary, could use advanced techniques, such as *stop loss*. By observing the general trend of an artificial stock market created with these rules, one can be amazed, by seeing that it resembles in many ways a real one. On the other side, agents can be modelled with inner reasoning and learning capabilities, for example using *neural networks*, *genetic algorithms* or *classifier systems*, which create an evolutionary environment. Each agent has the capacity to reason on the global effects of local actions, or even to create its own forecasts on the actions that will be performed by other agents. The agents built using this approach can decide on which action to perform, according to the stimuli coming from the environment, and not only according to their internal rules. According to (Bonabeau, 2002), Agent Based Modeling has three main benefits, over other approaches: it captures emergent phenomena; it provides a natural description of a system; and it is flexible.

## AGENT BASED PROCESS SIMULATION

In (Bonabeau, 2002), we read that agent based paradigm can be used successfully to model different situations, like flows, markets, organizations, social diffusion of phenomena; on the other hand, process

based approach has proved to be very useful for detailed, but static and deterministic, machinery and firm simulations. There are many intermediate situations, though, in which neither process simulation nor agent based approach can be applied with good results. Besides, the works of evolutionary economics, which use agents to represent industrial processes, also have a vision which is opposite to the static equilibrium, but are not meant to describe an enterprise or a machinery in detail. In (Remondino 2003), some examples can be found, about situations that can be described neither with pure process based nor pure agent based approaches, but could be modelled using a hybrid derived approach. Here I will only present a general framework for Agent Based Process simulations, shown in Figure 1. The market is the environment for the enterprise; there are buyers, i.e. the customers, and sellers, i.e. the suppliers. In a traditional process based simulation, these actors would be left out of the model, and the stress would be put on the way the single enterprise works. On the contrary, by using a pure agent based approach, we could model all these entities, but we couldn't model the real structure of enterprises in detail.

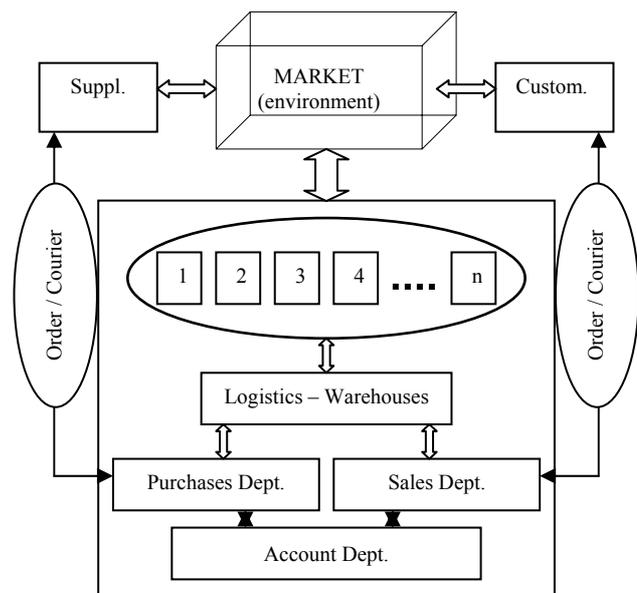


Figure 1: A General Framework for ABPS

Combining the two approaches, we can have a detailed model of the whole enterprise, with its production units, sales, purchases and account departments, logistics, warehouses and so on, modelled with a process based approach, and the environment, customers and sellers behaviour simulated using agent based technology. Besides, also the workers of the enterprise, i.e. persons in charge of machineries, department directors, disposers and so on. For example, sales and purchased departments could be modelled as shown in Fig. 2 and 3, while both customers and suppliers could be simple agents, acting on the basis of a probability function, based on real

data coming from market studies or simply randomly, if we want to see how the modelled enterprise reacts to whatever situation, even not realistic, coming from outside.

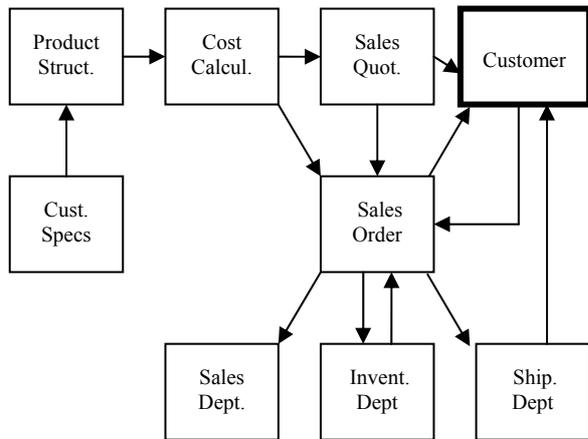


Figure 2: Typical Process Based Model of Sales Department

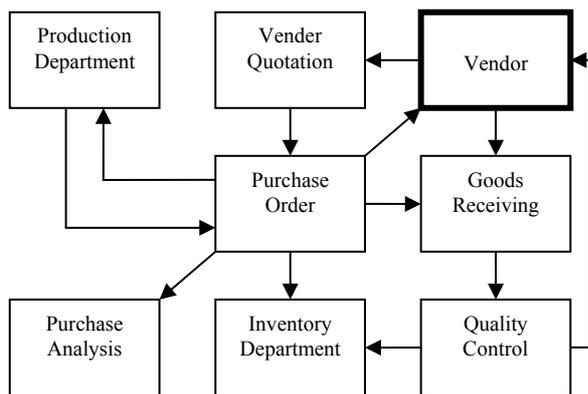


Figure 3: Typical Process Based Model of Purch Department

The blocks constituting the departments are processes, modelled in a deterministic way; their structure is made of elementary building blocks that use formalisms derived from Propositional or Modal Logic.

### SYMBOLIC AGENTS REPRESENTING PROCESSES

Usually, since processes can be modelled as deterministic flows, my proposal is to use both Propositional and Modal Logic to describe their structure. In (McCartney 2001) we read that:

*“The basis for most current systems of formal logic is Propositional Logic, also known as Propositional Calculus or PC. PC describes truth-based rules using*

*the fundamental ideas of not and or, and derivations of the concepts of and, implication, and strong implication. A common extension to PC is predicate logic. Predicate logic includes variables as well as non-truth-based validity; or mapping variables into values other than the Boolean true or false. Another non-truth based logic is modal logic, which is based on PC and introduces the concepts of necessity and possibility. Modal logic is closely related to PC and predicate logic, but is able to describe states that would be indescribable in either of these languages”*

In order to model a deterministic process, the Propositional Logic could be enough, since it allows to create truth tables of the single sub-processes. Modal Logic allows having a more versatile environment, allowing to determine if a proposition is true for sure, false for sure or sometimes true and sometimes false (i.e. it’s possible). In my framework I will only suppose the use Propositional Logic, to model simple processes: this allows to describe a process, create a model of it and simplify the transition to programming code required to port it into a working simulation. A sub-block of a process produces output\_1 if the logic formula is True, or output\_2 if it’s False; one of the two outputs can be simply Void. In this way, a part of a whole process can be like exemplified in Figure 4.

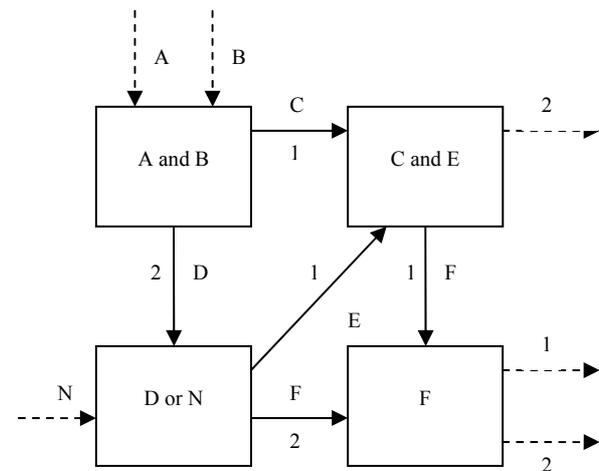


Figure 4: Propositional Logic Based Sub-block

Passing from this kind of representation to a programming language is a very easy step, since all the single boxes can be represented with if-then functions. In this way a very complex deterministic process can be modelled starting from very simple building blocks. Modal Logic can even add concepts of probability and necessity, so that a particular output going out from a basic building block can always occur or it’s possible that it occurs. In this case a probability function can be given, representing the views on the possible modal worlds, to specify how often an output can be produced, given the initial rule.

This approach allows to model machineries and the production units of an enterprise; the most difficult part to simulate, but probably also the most interesting for which regards the emergence of aggregate behaviour and self organization, is the human factor, e.g. the workers involved in the structure of the enterprise. Finding the optimal organizational structure of an enterprise is a very difficult task to accomplish, though a critical subject. There exist some tools, derived from AI studies, that allow embedding some sort of reasoning and learning capabilities into software agents. In particular, I will discuss about Genetic Algorithms and Classifier Systems.

## GENETIC ALGORITHMS AND CLASSIFIER SYSTEMS

In some situations, effective results can be obtained just by building simple agents, whose behaviour is randomly determined or is built by applying fixed pre defined reaction rules; this could be the case, for instance, of Heatbugs, one of our canonical Swarm demonstrations ([www.swarm.org](http://www.swarm.org)):

*“It’s an example of how simple agents acting only on local information can produce complex global behaviour. As we read on Swarm main site, each agent in this model is a heatbug. The world has a spatial property, heat, which diffuses and evaporates over time. In this picture, green dots represent heatbugs, brighter red represents warmer spots of the world. Each heatbug puts out a small amount of heat, and also has a certain ideal temperature it wants to be. The system itself is a simple time stepped model: each time step, the heatbug looks moves to a nearby spot that will make it happier and then puts out a bit of heat. One heatbug by itself can’t be warm enough, so over time they tend to cluster together for warmth”*

This is a useful approach when we wish to simulate situations in which we give the rules of the environment and we want to observe some emerging aggregate behaviour arising from simple entities; of course, the way the agents will act tends to be deeply dependent on the choices made by the programmer. As an alternative we can choose to create agents with the ability to compute rules and strategies, and evolve according to the environment in which they act; in order to model them, we can use some methods derived from the studies on artificial intelligence, such as artificial neural networks and evolutionary algorithms. While the former is a collection of mathematical functions, trying to emulate nervous systems in the human brain in order to create learning through experience, the latter derives from observations of biological evolution. Genetic Algorithms derive directly from Darwin's theory of evolution, often explained as "survival of the fittest": individuals are modelled as strings of binary digits and are the encode for the solution to some problem. The first generation of individuals is often created

randomly, and then some fitness rules are given (i.e. better solutions for a particular problem), in order to select the fittest entities. The selected ones will survive, while the others will be killed; during the next step, a crossover between some of the fittest entities occurs, thus creating new individuals, directly derived from the best ones of the previous generation. Again, the fitness check is operated, thus selecting the ones that give better solutions to the given problem, and so on. In order to insert a random variable in the genetic paradigm, that’s something crucial in the real world, a probability of mutation is given; this means that from one generation to the next one, one or more bits of some strings can change randomly. This creates totally new individuals, thus not leaving us only with the direct derivatives of the very first generation. Genetic Algorithms have proven to be effective problem solvers, especially for multi-parameter function optimization, when a near optimum result is enough and the real optimum is not needed. This suggests that this kind of methodology is particularly suitable for problems which are too complex, dynamic or noisy to be treated with the analytical approach; on the contrary, it’s not advisable to use Genetic Algorithms when the result to be found is the exact optimum of a function. The risk would be a convergence to some results due to the similarity of most the individuals, that would produce new ones that are identical to the older ones; this can be avoided with a proper mutation, that introduces in the entities something new, not directly derived from the crossover and fitness process. In this way, the convergence should mean that in the part of the solution space we are exploring there are no better strategies than the found one. It’s crucial to choose the basic parameters, such as crossover rate and mutation probability, in order to achieve and keep track of optimal results and, at the same time, explore a wide range of possible solutions. Classifier Systems derive directly from Genetic Algorithms, in the sense that they use strings of characters to encode rules for conditions and consequent actions to be performed. The system has a collection of agents, called classifiers, that through training evolve to work together and solve difficult, open-ended problems. They were introduced in (Holland 1976) and successfully applied, with some variations from the initial specifics, to many different situations. The goal is to map if-then rules to binary strings, and then use techniques derived from the studies about Genetic Algorithms to evolve them. Depending on the results obtained by performing the action corresponding to a given rule, this receives a reward that can increase its fitness. In this way, the rules which are not applicable to the context or not useful (i.e. produce bad results) tend to loose fitness and are eventually discarded, while the good ones live and merge, producing new sets of rules. In (Kim, 1993) we find the concept of Organizational-learning oriented Classifier System, extended to multi-agent environments with introducing the concepts of organizational learning. According to (Takadama 1999), in such environments agents should

cooperatively learn each other and solve a given problem. The system solves a given problem with multi-agents' organizational learning, where the problem cannot be solved simply by the sum of individual learning of each agent.

### EVOLUTIONARY METHODS APPLIED TO ABPS

Agent Based Process Simulation is a way to model deterministic structures, made up of single processes, divided into Propositional Logic based building blocks, and having them interact with agents belonging to the sub-symbolic paradigms. This allows to simulate situations in which not only the deterministic structure, but also unpredictable situations could arise, caused by the environment or the human factor are important; we can think about many different situations, that couldn't be represented by a pure process based approach, and would result too difficult and inaccurate to be modelled just using self organizing agents. For example, agents could be part of the structure of an enterprise modelled with process based approach; they could be regarded as parts acting more like human beings than like machines. We may think of a generic enterprise, in which many sub-systems, i.e. units, can be described with a process based approach. The interaction between these basic subsystems, though, is usually really complex, and generally involves a human or non deterministic participation. This would be very difficult, or even impossible to represent with a process based model; but it would also be useless to use a pure agent based approach, since many parts could be only modelled with structured and Logic based processes. That's where we can use agent based connections between the process based sub-systems. These agents should be quite simple, but structured ones, able to act starting from stimuli coming from the environment (i.e. the output of a sub-system modelled with process based approach), and to produce an output, that will effect the way other sub-systems will work. In a simulation built in this way, we can see what happens if we change the way we manage the warehouses, if we use more experienced employers or, for example, if the workers are on a strike. With the same approach, we can go down to a micro level, for example by inserting agents into models of the single machineries and business units. If we think of a single, but very complex machinery, not all the parts are strictly deterministic, in the sense that they can be affected by some unforeseen influence coming from the environment. By using a process based approach, it is possible to model the machinery quite deeply, but just in a deterministic, static situation, which is the optimal environment, in which nothing can change its way of working: we can simulate the variation of the output by varying the input, or by improving some part of the system. Or we can prove the resistance and endurance of the machinery in optimal conditions. Though, such a simulation, for its

nature, wouldn't be able to consider any chaotic or unforeseeable action, coming from outside, that could compromise the machine operations (e.g. damages caused by moist, fire, and so on). A representation of a typical process based model of a machinery is given in Figure 5.

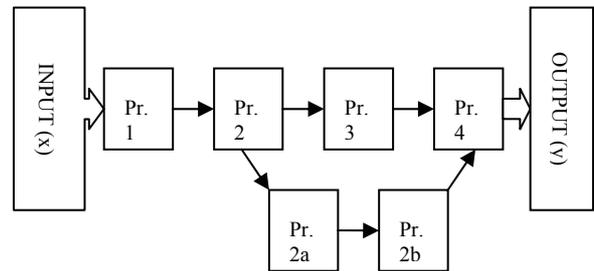


Figure 5: Example of Structure for Process Based Machinery

The model acts as a function which receives an input, that is the independent variable ( $x$ ), processes it and produces a stochastic output, which is the dependent variable ( $y$ ). Though very powerful and easy to validate, this is not always realistic. By considering certain parts of the machinery as very simple agents (Figure 5), it would be possible to create a more realistic model of the object, that will be able to react to the stimuli coming from the environment according to certain rules, written in the single agents, that would give the whole machinery a complex, and less deterministic behaviour, just as the one it would have in the real world. An example for this is given in Figure 6, derived from the previous one, with the insertion of some agents into the process based structure.

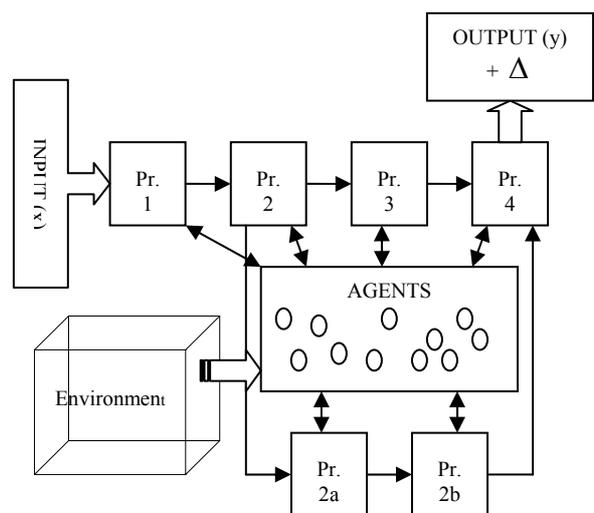


Figure 6: The Same Structure, with the Addition of Agents

Some agents are now put side by side with processes: while the main flow remains unchanged, now there is an influence coming from a hypothetical environment, just like in the real world. The agents can react to the stimuli coming from outside, which can be the rest of the enterprise, another machine, or even the person running the simulation. In this way, the model is not strictly static anymore, in the sense that given an input (x), same as before, the output is not a linear function of just the independent variable, but also of all the other ones that can be processed by the agents. The output is not the same (y) as before, but (y) plus a delta, which is caused by external, non deterministic influences on the agents. Of course these agents must act logically, on the basis of what could possibly happen in the real world; a totally random acting agent would be, obviously, useless. This kind of approach allows to build models which are more realistic and dynamic, as opposed to the static ones, where only the deterministic flows are simulated. The greatest difficulty, with this approach, is model validation, using data from real experiments, because the unexpected circumstances are difficult to reproduce more than once. The validation could then go top-down, in the sense that we observe certain data in the reality and then try to reproduce the same situation in the model, by using the agents in a piloted way. If the same results occur, it is possible to calculate a standard statistical error and validate the model for those particular situations. We can then extrapolate the results, and consider the model valid also for those situations that can't be controlled and created in the real world. If the agents involved in the simulation, at each level, are modelled using evolutionary methods such as Genetic Algorithms and Classifier Systems, we could achieve two main goals:

the agents simulating the environment could evolve and self organize, thus creating a realistic situation. Besides, if we use Classifier Systems to model the agents, we could find the optimal rules for the organization of a given enterprise, which is modelled through deterministic processes. In order to do that, we start from some basic parameters and performance indicator, that will serve as the rules to determine the fitness of the agents involved. The agents that will produce the higher local results will survive and merge, in order to create new generations derived from them; after many simulated steps, we should be able to find an optimal global organization of the simulated enterprise (or business unit, or even machinery), modelled using processes. For example, we can choose to maximize the local output of a production unit, given an input; the production unit will be modelled with a process based approach, using Propositional Logic formalisms for the building blocks. The human beings involved in each production unit are modelled as agents based on Classifier Systems, as represented in Figure 7. In this framework the single agents, which act as self evolving connections between the processes within the various business units, are modelled using Genetic Algorithms. By using some simple reference parameters, in particular the local output, that's the output produced by the single business units, it's possible to assign a fitness value to the agents. When the simulation starts, a population of random agents is created (random binary strings) for each unit. They produce certain effects operating on the process based parts of the production unit through their actions: we could say that they operate the units in a random way.

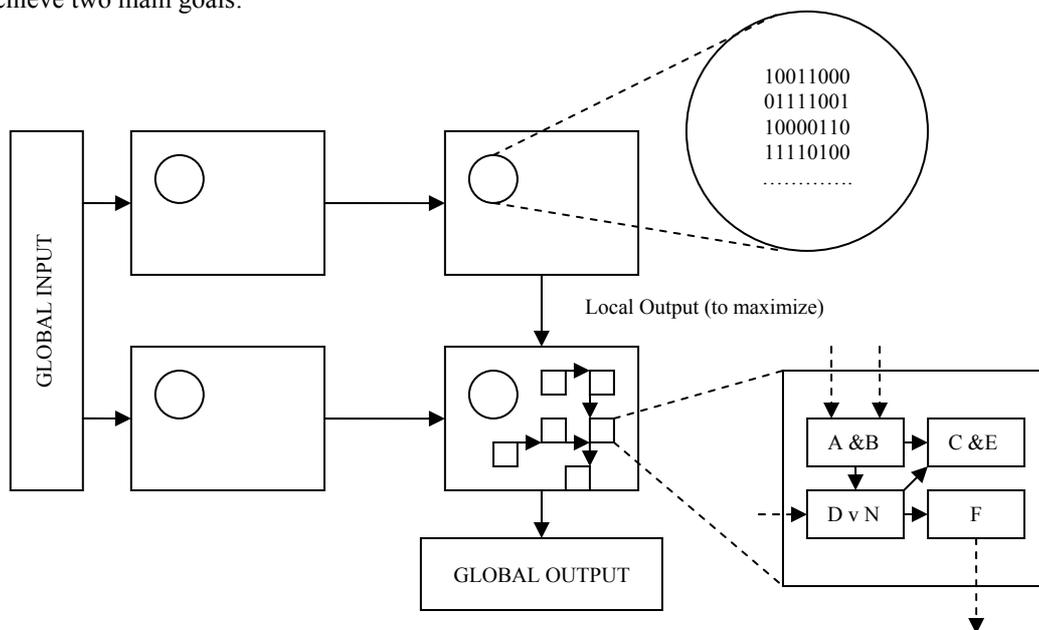


Figure 7: AI Evolutionary Methods Applied to ABPS

The agents that produce the best local output, given a certain input, are saved and used for the crossover among them. The resulting agents now compete against the previous local optimum, and again only the ones with better results are kept and crossed. The mutation factor is also important, since it introduces a variability that won't be present with the simple crossover between the subjects involved in the simulation. After several simulation steps, all the business units will have maximized the local output (or at least the agents will have reached the best possible result among the solution space observed), and thus also the final output, that is the result of the interaction between the various units, would be maximized. At this point, by looking which kind of agents survived to the selection and produced the optimal results, we can understand what the best way to operate each unit is.

Instead of using Genetic Algorithms inside the production units, we could use Classifier Systems to map the rules of the single processes; since the rules, modelled with Propositional Logic, can be encoded as if-then conditions, the Classifier Systems can map them successfully and evolve them in order to find the optimal organization of the process based structures. Again, this could be based on very simple performance indicators, like the local output, given a local input or the time required to complete an operation. Other constraints could be found for both the approaches; for example, it's possible that some of the found structures are not applicable to the real world. In this case, those should be discarded even if their local performance is higher than others.

## CONCLUSIONS AND FUTURE DIRECTIONS

Evolutionary Methods derived from the studies in AI fields have proven to be effective problem solvers, particularly for situations where a near optimum result is enough and the real optimum is not needed. This suggests that they can be successfully applied to those situations in which the traditional approaches (e.g. mathematical ones) are inapplicable, since the system is too complex, dynamic or fuzzy. Enterprise simulation, done through the proposed Agent Based Process approach, can be an application field for Genetic Algorithms and Classifier Systems; here an enterprise or a part of it is split into a set of processes, made of basic building blocks, each of one modelled as a very simple rule of Propositional Logic. These are put side by side with reactive and evolving agents, in order to find an optimal structure for the modelled enterprise through self organization. The agents are evolved on the basis of a fitness function, representing some core performance indicator (e.g. the local output of the single production unit, the time used for the single process and so on). There are at least two main configurations for the proposed approach: the first one uses Genetic Algorithms for the agents operating every business unit in the simulation; the second one employs Classifier Systems to optimize the single

processes, by mapping the Propositional Logic rules and making them evolve. In the future a working example of both the approaches will be created and some practical results will follow, showing that evolutionary methods can be a great help for decision making and business process reengineering. A meta-model for Agent Based Process Simulation is also in the works, to be considered as the prototype for all the models built in this way, showing which parts of a generic enterprise can be modelled using Logic based deterministic processes, and which require the use of intelligent learning agents.

## REFERENCES

- Bahrami, A., Sadowski D. and Bahrami S. 1998. "Enterprise architecture for business process simulation", *Proceedings of the 1998 Winter Simulation Conference*
- Bonabeau, E. 2002. "Agent-based modeling: Methods and techniques for simulating human systems", *PNAS 99 Suppl. 3: 7280-7287*.
- Gilbert, N. and Troitzsch, K.G. 1999. "Simulation for the Social Scientist", Open University Press
- Gilbert, N. and Terna, P. 2000. "How to build and use agent-based models in social science", *Mind & Society 1, 57-72*
- Helsgaun, K.2000. "Discrete Event Simulation in Java", *Writings on Computer Science, Roskilde University*
- Holland, J.H.1976. "Adaptation", In R. Rosen and F. M. Snell, editors "Progress in theoretical biology", New York: Plenum
- Kim, D. 1993. "The Link between individual and organizational learning", *Sloan Management Review*, pp. 37-50.
- McCartney, R. 2001. "A Short Introduction to Modal Logic", *UNCG CSC 656, Spring*
- Remondino, M. 2003. "Agent Based Process Simulation and Metaphors Based Approach for Enterprise and Social Modeling", *ABS 4 Proceedings, SCS Europ. Publish. House*
- Takadama, K. et al. 1999. "Making Organizational Learning Operational: Implication from Learning Classifier System" in *J.Comp. and Mathematical Organization Theory*, Vol. 5, No. 3, pp. 229-252.
- Terna, P. 2002. "jVEFrame: a Virtual Enterprise Frame in Swarm", *SwarmFest 2002 Conference*, working paper

## AUTHOR BIOGRAPHY

**MARCO REMONDINO** was born in Asti, Italy, and studied Economics at the University of Turin, where he obtained his Master Degree in March, 2001 with 110/110 cum Laude et Menzione and a Thesis in Economical Dynamics. In the same year, he started attending a PhD at the Computer Science Department at the University of Turin, which will last till the end of 2004. His main research interests are Computer Simulation applied to Social Sciences, Enterprise Modeling, Agent Based Simulation and Multi Agent Systems. He has been part of the European team which defined a Unified Language for Enterprise Modeling (UEML). He is also participating to a University project for creating a cluster of computers, to be used for Social Simulation.

# THE SMALL BUSINESS MANAGEMENT FLIGHT SIMULATOR IN AN ENVIRONMENT OF FINANCIAL INDISCIPLINE

Mirjana Pejic Bach  
Department for Business Computing  
Faculty of Economics, University of Zagreb  
Trg J.F.Kennedya 6, 10000 Zagreb, Croatia  
E-mail: mpejic@efzg.hr

## KEYWORDS

System dynamics modelling, simulation games, small business finance, accounts receivable management, financial indiscipline

## ABSTRACT

The article describes the features of a system-dynamics based game for financial strategies of small business in an environment characterised by severe financial indiscipline and with the restricted access to financial resources. The problem of small firm striving to succeed in such conditions is discussed in the framework of a system dynamics model that is converted into a simulation game. Player makes the decisions on bank credit, accounts receivable policy, profit payout, and time to pay the suppliers.

## INTRODUCTION

One of the typical problems for the transition countries of Central and Eastern Europe is financial indiscipline, when firms delay payments to suppliers, banks, employees and the government (Begg and Portes, 1993). It is very hard for a small business to survive in such an environment. If its buyers do not pay on time, the firm will initially face liquidity crises and will eventually go bankrupt. Restricted access to financial markets is the additional problem that small businesses face (OECD, 1996).

System dynamics is a powerful tool that enhances learning about company, market and competitors, portrays the cognitive limitations on the information gathering and processing power of human mind, facilitates the practice of considering opinions, and supports building of "What if" scenarios (Sterman, 2000).

Over the past twenty years, the growth of computer technology has facilitated the wide application of system dynamics modelling as sophisticated tools for simulating business environments and situations. The basic goal of management simulation games is to apply experiential learning to the commercial world. They are designed in order to allow the player to experiment with the model on a compressed time basis while reducing costs and personal risk. The participant is able to see the consequences of his/her actions in few minutes or hours. In real world such consequences are visible only after much longer time (months or years).

The purpose of this article is to describe and demonstrate applicability of system dynamics models as decision and learning support tools for small businesses in an environment of financial indiscipline that permit controlled experimentation and enhance understanding of reality.

## MODEL OVERVIEW

Several system dynamics studies have examined the impact of financial policy on business success. Lyneis (1980) proposes the use of system dynamics models in deciding which action (e.g. capital rationing, increasing prices, reducing the collection period) would be best in the case where a firm faces a shortage of financial resources. Indeed, this is a common problem for all growing companies. Thompson (1986) examines the impact of cash flow on the success of a small business, and suggests that system dynamics can provide an overview of the complex relationships between inventory, receivables, payables and cash. Kolay (1991) recommends a system dynamics approach in managing working capital crises, and demonstrates the influence of debt collection efforts, restricted credit policy, and deferred payment to creditors on working capital. Bianchi and Mollona (1997) show how the coupling of the dynamics of growth with that of net working capital management creates recurring problems, especially in the context of small entrepreneurial firms. An interactive learning environment has also been created, whose focus is on understanding the dynamics generated by commercial and financial policies on sales revenue and profitability on one hand, and net working capital and liquidity on the other hand (Bianchi and Bivona, 1999).

The system dynamics model of a small firm in the transitional Croatian economy striving to succeed in an environment characterised by severe financial indiscipline and with restricted access to financial resources. The goals of the study were to help the owners of the firm to decide which financial policy would be best in the situation of financial indiscipline and to demonstrate an application of system dynamics methodology in a small business environment. The model is discussed at length by Pejic-Bach (2003), and is converted into a simulation game that consists on six major sectors: sales force, accounts receivable policy, finance, demand, inventory, and financial sources. Model sectors shall be described shortly.

## Sales force sector

Sales force sector describes sales force size and hiring effort (Figure 1). The owner compares the current and

required number of workers and employs new staff in order to reduce the difference. The number of workers required depends on the expected demand.

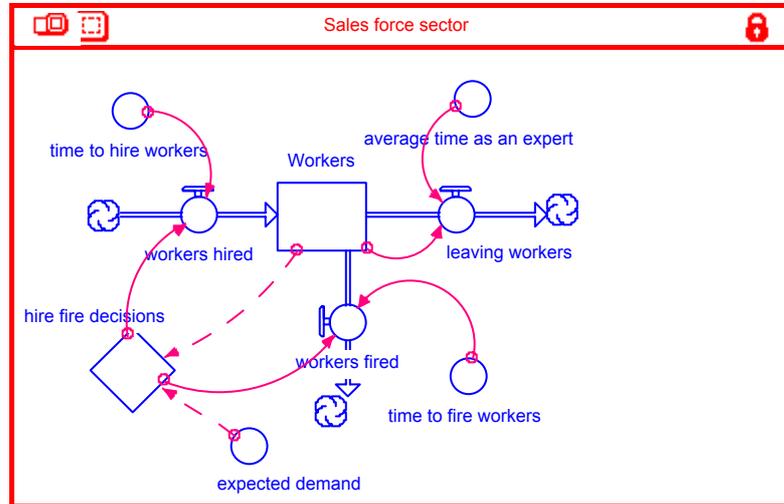


Figure 1: Stock and flow diagram of sales-force sector

## Accounts receivable policy

Accounts receivable policy sector explains effect of financial discipline on level of accounts receivable, and how financial discipline could be controlled with accounts receivable policy.

Increase in accounts receivable inflows into the level of accounts receivable, and it depends on the number of products sold and the selling price (Figure 2). Three outflows reduce the level of accounts receivable: (1) payment in advance, (2) the accounts receivable that are collected, and (3) bad debts. Payment in advance depends on the selling and payment in advance fraction that is

Financial discipline influences level of accounts receivable by the following means: time to collect accounts receivable, percentage of bad debt and percentage of early payment. Normal financial discipline was defined as a situation where customers pay on time, where 30% of sales are collected in advance, and where 3% of accounts receivable are bad debts that cannot be collected despite

influenced by financial discipline. The amount of accounts receivable collected depends on the current level of accounts receivable and the usual time to collect accounts receivable, which is influenced by financial discipline. The current level of the accounts receivable and bad debt fraction determines the amount of bad debt. However, bad debt does not reduce the accounts receivable immediately. The owner of the firm will put pressure on bad customers after the usual collection time has passed. For example, bad debts reduce the level of accounts receivable after 9 months if the time to collect accounts receivable is 6 months and the owner puts pressure on bad customers for an additional 3 months.

additional efforts. When financial discipline is lower than normal, the time needed to collect accounts receivable is longer than agreed, fewer than 30% of customers pay in advance, and more than 3% of accounts receivable are considered as bad debts.

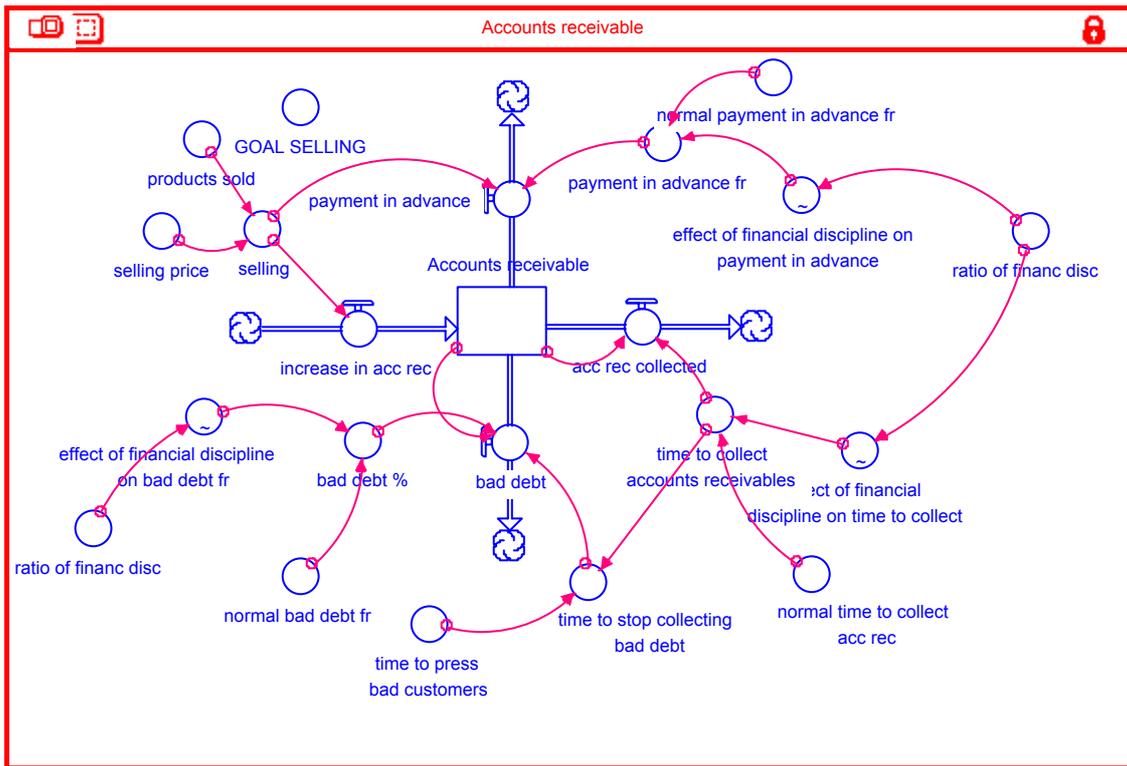


Figure 2: Stock and flow diagram of accounts receivable

How can financial discipline be controlled? Is it possible at all? What are the consequences of such actions? Accounts receivable policy is one of the means to influence financial discipline of firm's customers. It comprises credit standard, the collection policy, and cash discount. It would

be too complicated to model all the aspects of the management of accounts policy. Therefore, we modelled the accounts receivable policy as one variable that could vary between two extremes – a liberal and a restrictive accounts receivable policy (Figure 3).

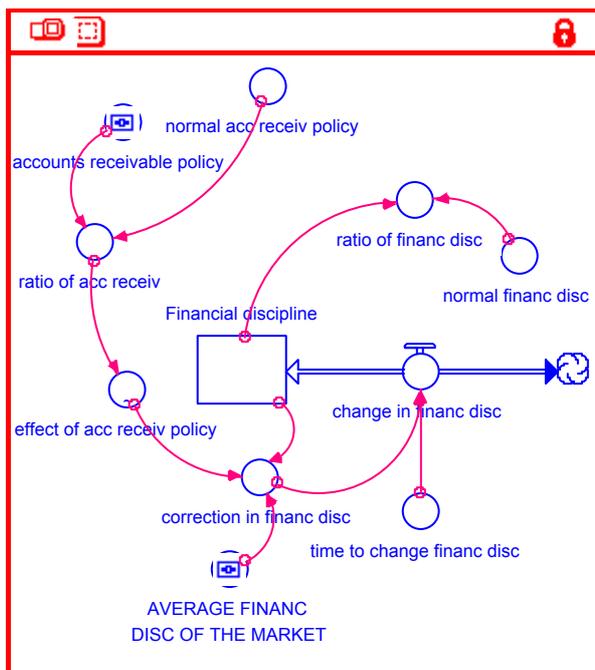


Figure 3: Stock and flow diagram of accounts receivable policy

An accounts receivable policy is liberal if the credit period is long, credit standards are low and if the collection policy is loose. Easing the credit policy stimulates sales, but carrying costs and bad debt and/or cash discount expenses may also rise. An accounts receivable policy is restrictive in the opposite case where the credit period is short, credit standards are high and the collection policy is tightened. Tightening accounts receivable policy increases financial discipline of customers, but the firm risks to lose its customers.

Players make the decisions on accounts receivable policy that is expressed in pressure units. If financial discipline is normal, than the player retains accounts receivable policy also at the normal level (50 pressure units). The accounts receivable policy is expressed in pressure units. If the accounts receivable policy is completely liberal, (the firm sells goods to everybody, does not put pressure on

customers, and the credit period is long), its value is 0 pressure units. More than 50 pressure units represent a restrictive accounts receivable policy.

### Finance sector

Finance sector contains selling price calculation, cash flow, income statement and ratio analysis. We shall explain only the cash level, that is increased by the inflow and depleted by the outflow (Figure 4). Inflows to the cash level are: payment in advance, accounts receivable collected, and pre-tax return. In addition, if the firm borrows money from the bank, the cash level is increased by the amount borrowed. Outflows from the cash level are: paying profit, paying administrative and selling costs, paying suppliers, and paying taxes. If the firm has borrowed money from the bank, the cash level is decreased by the amount repaid and by interest payments.

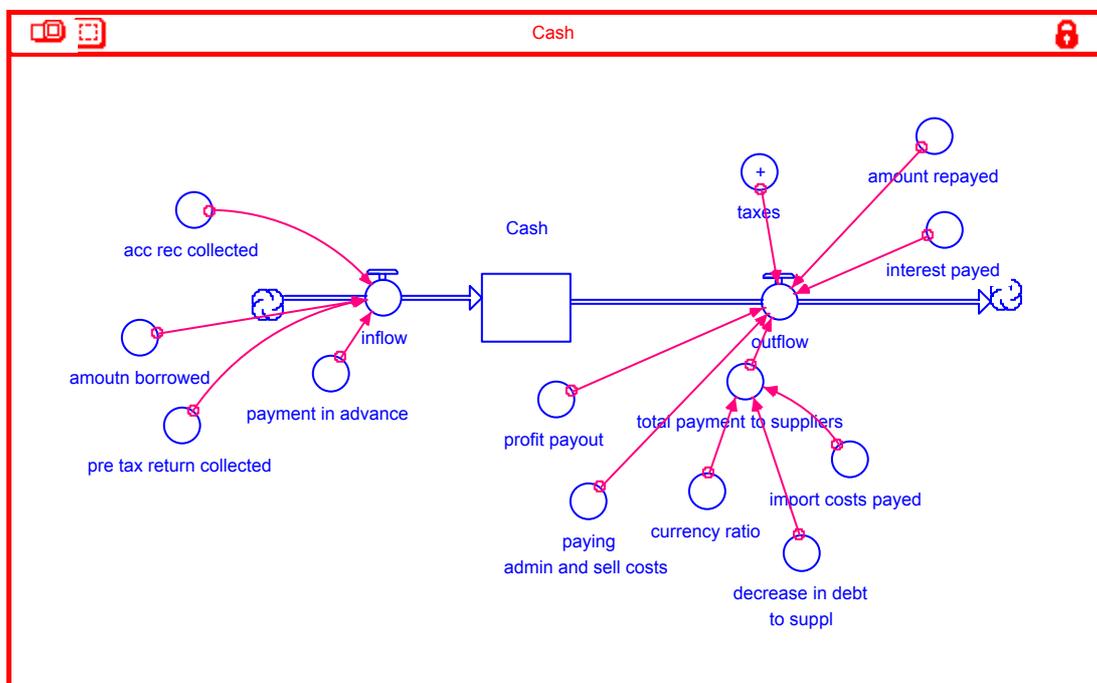


Figure 4: Stock and flow diagram of the cash level

### Demand sector

Demand sector explains customer creation and retention, and main factors that influence demand are: (1) word of mouth, (2) accounts receivable policy, and (3) marketing expenses (Figure 5). In the context of financial discipline the accounts receivable policy and marketing expenses influences the demand to the biggest extent. When accounts receivable policies are liberal, sales increase, but

more customers with poor financial discipline are attracted. If the accounts management policy is restrictive, sales decrease, but customers of greater financial discipline are more likely to be encouraged. The main way to influence demand is through marketing expenses. If marketing expenses are higher than the similar expenses of competitors, demand increases very quickly. The goal of the model is not unrestricted growth, but growth only to the target level.

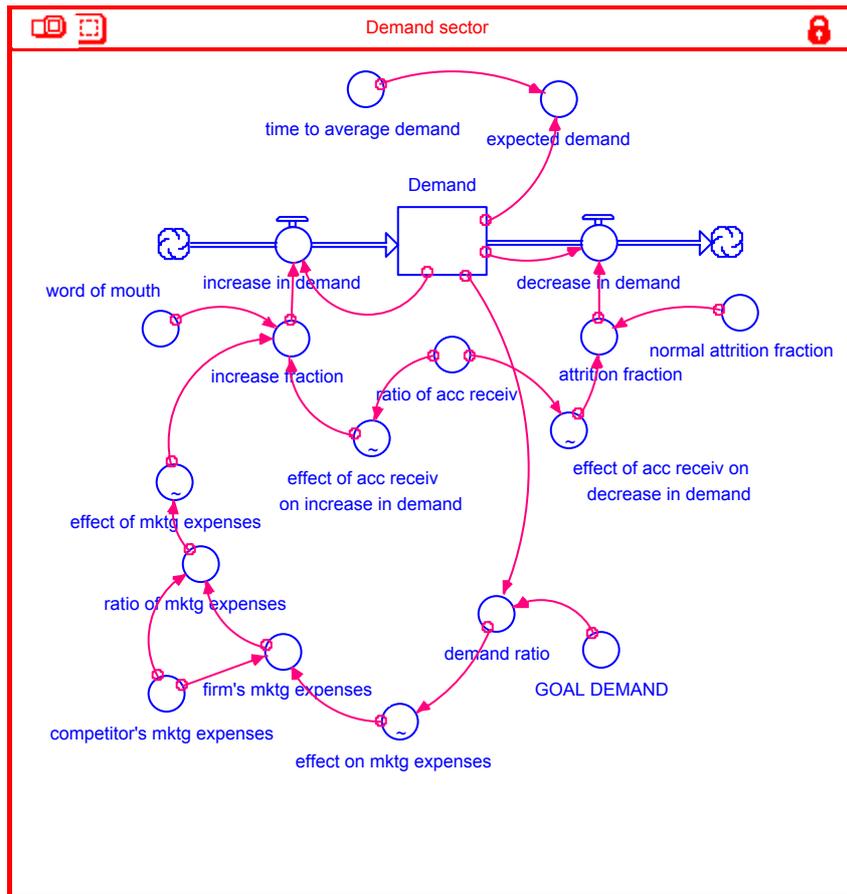


Figure 5: Stock and flow diagram for demand sector

### Inventory sector

The inventory level is increased by the products received and is depleted by the products sold (Figure 6). The owner

takes the expected demand into account when she orders new products. She compares the current and target inventory and orders sufficient products to reduce the difference.

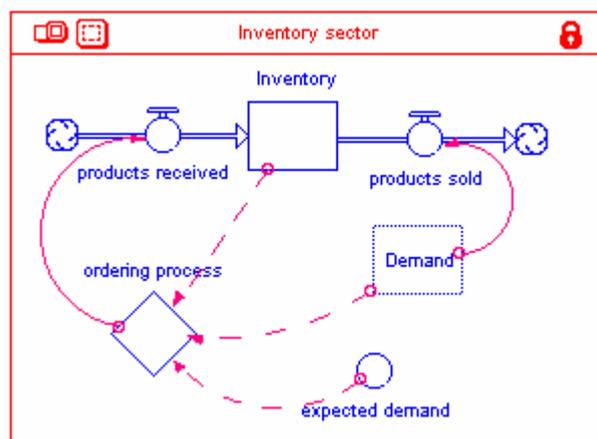


Figure 6: Stock and flow diagram for inventory sector

### Financial sources sector

Three financial sources are available to the small businesses: suppliers (Figure 7), bank credit (Figure 8) and profit retained (Figure 9). Players make decisions on time to pay suppliers, bank credit and profit payment policy.

When cash flow is lower than acceptable, the player can: (1) increase time to pay suppliers, but never above the highest limit that suppliers will tolerate, (2) retain profit to finance day-to-day business, and (3) take credit from the bank, but only the maximum amount approved by the bank.

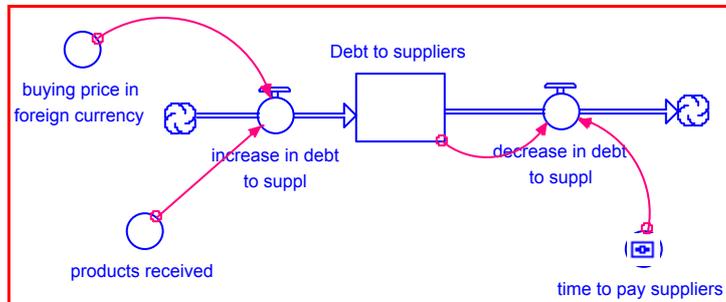


Figure 7: Stock and flow diagram for debt to suppliers

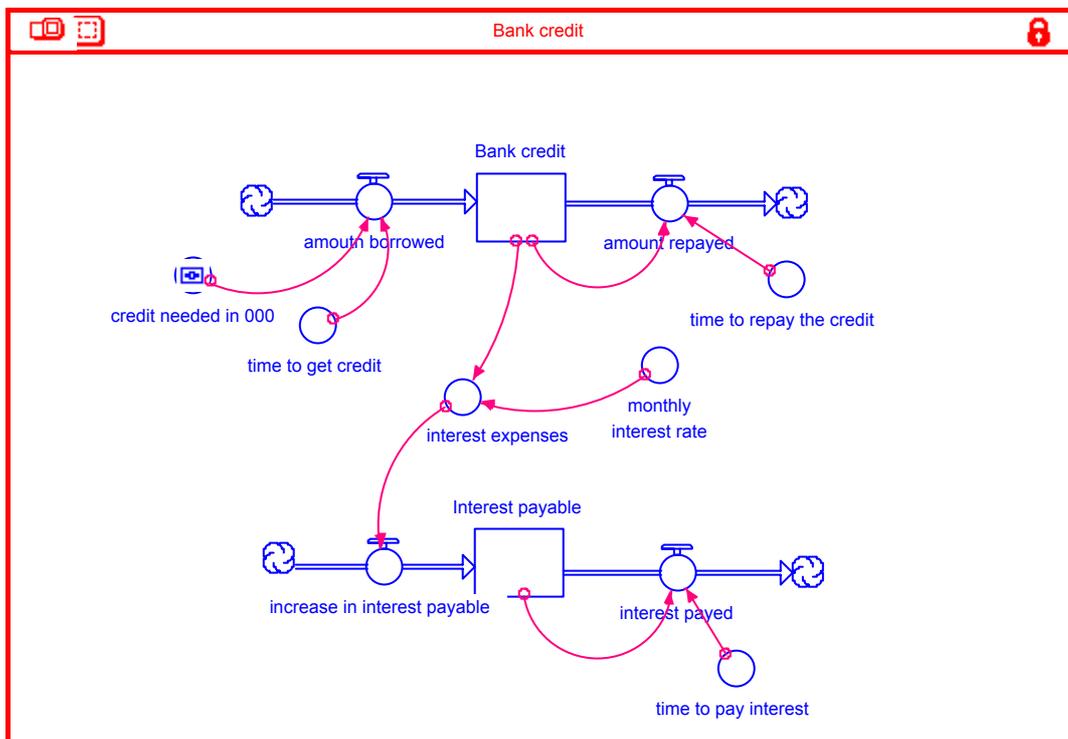


Figure 8: Stock and flow diagram for bank credit

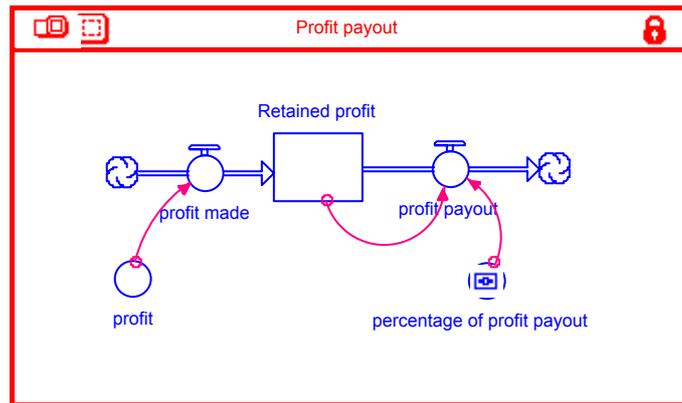


Figure 9: Stock and flow diagram for profit payout

### THE SMALL BUSINESS MANAGEMENT FLIGHT SIMULATOR

Basic assumption of the model is that the financial discipline is sensitive on accounts receivable policy. There are four controllable variables in the model: (1) % profit payout, (2) bank credit, (3) time to pay suppliers, and (4) accounts receivable policy.

The simulation game has been designed and played with the objective to sensate management students and practitioners to the problems of financial indiscipline. Team of players or one player makes decisions about four control variables. The players are encouraged to increase turnover, maximise profit while maintaining positive cash flow. Every six months they make decisions.

Game begins with discussion about the financial indiscipline, and how it influences firms profitability and liquidity. Then, system dynamics model of small business is presented. Players gather around a machine for one or

two hours. Goal of the game is to make as large profit and demand as possible, and in the same time remain the firm in the liquid position in the conditions of financial indiscipline. After the game the players are asked to explain what they have observed.

Itthink is the system dynamics software that is simple to operate and has interface that encourages exploration. It offers user-friendly input and output devices that empowers the player to make decisions about controllable variables. Control Panel for decision-making is presented (Figure 10.) Input devices are Slider Inputs, which allow the player to make decisions. Every six months players make the decisions on credit needed, percentage of profit payout, time to pay suppliers, and accounts receivable policy. Output devices show the consequences of player's decisions on Numeric Display or graph. Players monitor time to collect accounts receivable, bad debt percentage, cash, and profit or loss on Numeric Displays. Demand, profit and cash are monitored on graphs.

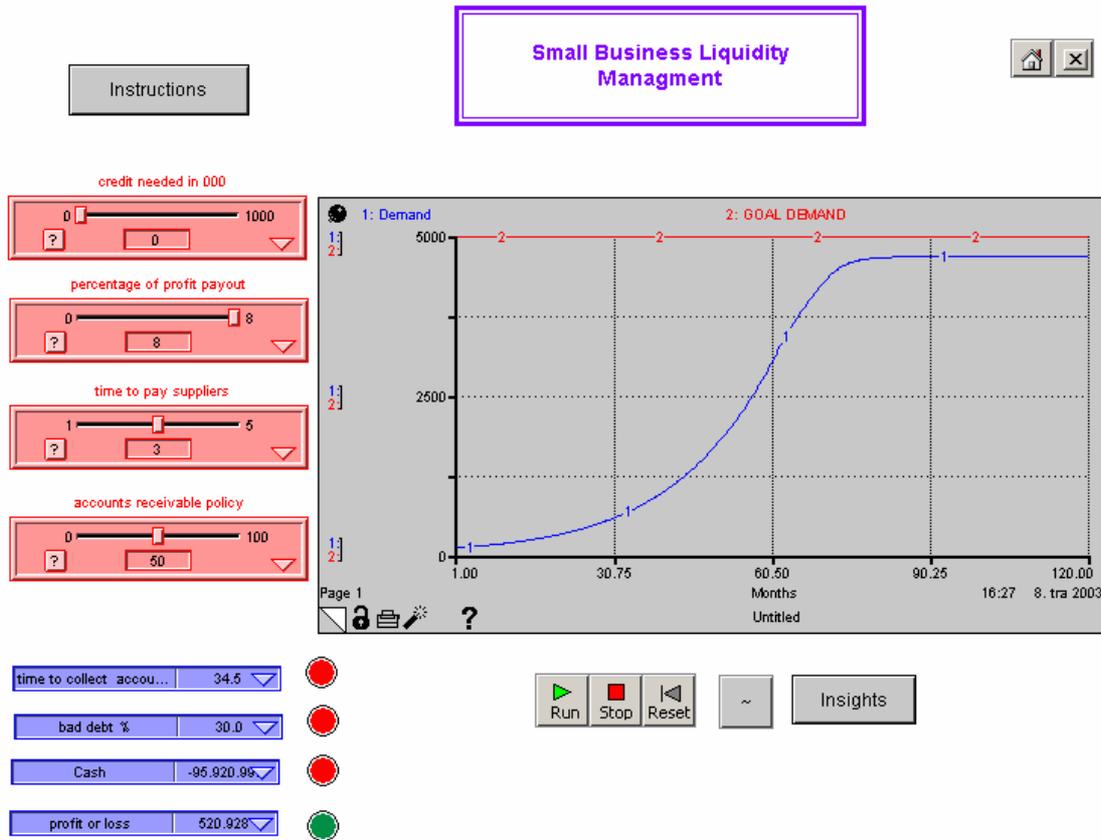


Figure 10: Control panel of the game

## LESSONS OF THE GAME

In an ideal situation every customer would pay on time, there would be no bad debt, and the fraction of early payments would be fairly high. It is presumed that a sudden and strong decay of financial discipline was introduced.

Following strategies in case of financial discipline problems are considered:

- Strategy 1 - Retaining profit
- Strategy 2- Retaining profit and borrowing money from the bank
- Strategy 3 - Retaining profit, borrowing money from the bank and tightening the accounts receivable policy (75 pressure units)

- Strategy 4 - Retaining profit, tightening the accounts receivable policy (65 pressure units) and delaying payments to suppliers (5 months)

Game is played with employing above strategies, and the results are following (Table 1). Equilibrium values at the end of the simulation for demand, time to collect accounts receivable, bad debt %, cash and profit are presented in the table. As far as demand is concerned, Strategies 1 and 2 are better than Strategies 3 and 4. But, in terms of the time to collect accounts receivable, bad debt, and profit Strategies 3 and 4 are better. However, the firm faces serious liquidity problems in Strategies 1 and 2.

Table 1. Business performance in different Strategies

	Demand (number of products sold per month)	Time to collect accounts receivable (months)	Bad debt %	Negative cash position (local currency in millions)	Profit (local currency)
Strategy 1	4687	26.2	22.5	Yes (-52.07)	698,986
Strategy 2	4687	26.2	22.5	Yes (-49.57)	658,176
Strategy 3	3980	5.6	5.2	No	1,078,253
Strategy 4	4149	6.7	5.7	No	1,110,523

The players would make the best results if they tighten the accounts receivable policy and, at the same time, delay payments to suppliers (Strategy 4). However, such a strategy is good for the firm only in the short term, because suppliers will not tolerate delayed payments indefinitely. As a result of such actions, the firm would survive, but it would lose its market share and would build up debts to its suppliers. All of the profit would be retained and used to finance the day-to-day running costs of the business. Most of the firms in transition countries that suffer from financial indiscipline operate in the same way. Firms delay payments to suppliers and they themselves turn into bad customers. Therefore, the circle is closed. In the end everybody delays in making payments, and financial indiscipline spreads even more widely. The conclusion of the game is that in conditions of financial indiscipline, there is no win-win solution for everybody.

Players are encouraged to think about available strategy options in the conditions of financial indiscipline, instead of sudden changes of the strategy. Main goal of the simulation experiments is to demonstrate the following conclusions for small business firms (Pejic-Bach, 2003):

- When a firm's customers start to delay with their payments, the worst solution would be to do nothing. In that case, every firm would very quickly become illiquid and would not be able to pay its taxes, suppliers, employees, and creditors.
- Credit from the bank would not be sufficient to cover the negative cash position because of the restricted access felt by most transition countries to the financial market.
- A good solution would be to tighten the accounts receivable policy, which would yield the following results: an increase in the fraction of early payments, a decrease in the fraction of bad debt and time to collect payments. Still, because of a restrictive accounts receivable policy the firm would eventually lose its market share.
- The other solution for the firm would be to use informal sources of credit and to delay payments to suppliers, which is the most likely reaction. Because of the inefficiency of the legal system, most firms decide to pay debts late simply

because the cost associated with late payment is smaller than the cost of alternative sources of finance. Underpaid suppliers usually do not terminate further shipments for fear of losing their clients, and, as a result, mutual arrears become a universal practice.

## CONCLUSIONS

Traditional methods of teaching tend to equip the students with knowledge that could eventually help them in solving their future business problems. On the other hand, simulations are designed in order to initiate active, student-oriented learning. Students seek information that is useful in achieving a goal of the game, and during that process their understanding of the system increases.

Simulation games hold out the promise of new and advantageous ways of learning, and could be useful in education of credit managers. System dynamics model of small business firm has been designed in order to allow players to make decisions in conditions of financial indiscipline in non-risk environment on a compressed time basis. Players are encouraged to decide on credit needed, percentage of profit payout, time to pay suppliers, and accounts receivable policy in order to maximise profit and demand while maintaining positive cash flow. Proposed simulation game could be useful in developing decision-making skills, and could increase understanding of possible strategy options available for small business firm in the conditions of the financial indiscipline.

Finally, players are encouraged to reach the following conclusions. In an environment of financial indiscipline and restricted external sources of finance, most small firms facing liquidity problems will probably restrict their customer base and delay payments to suppliers, so decreasing their market share. If the legal system is inefficient, such a reaction will lead to an illiquid economy in which the small business sector is weak. This would be a severe problem because small business firms usually play an important role in every economy. Furthermore, everybody would eventually become a bad debtor.

## REFERENCES

- Begg, D. and R. Portes. 1993. "Enterprise debt and financial restructuring in Central and Eastern Europe". *European Economic Review* 37, 396-407.
- Bianchi, C. and E. Bivona. 2000. "Commercial and financial policies in family firms: The small business growth management flight simulator". *Symulation & Gaming* 31, 197-292.
- Bianchi, C. and E. Mollona. 1997. "A behavioural model of growth and net working capital management in a small enterprise". In *Proceedings of the 1997 International System Dynamics Conference*. Istanbul, 269-272.
- Kolay, M.K. 1991. "Managing working capital crises: A system dynamics approach". *Management Decision* 29, 46-52.
- Lyneis, J.M. 1980. *Corporate Planning and Policy Desing: A System Dynamics Approach*. Pugh-Roberts, Cambridge.
- OECD. 1996. *Micro-credit in transitional economies*. OECD, Paris.
- Pejic-Bach, M. 2003. "Surviving in an environment of financial indiscipline: a case study from a transition country". *System Dynamics Review* 19, 47-74.
- Sterman, J.D. 2000. *Business dynamics: System thinking and modelling for a complex world*. Irwin McGraw-Hill, Boston.
- Thompson, R. 1986. "Understanding cash flow". *Journal of Small Business Management* 24, 23-30.

## BIOGRAPHY

Mirjana Pejic-Bach is a Research Assistant in Business Informatics at the Faculty of Economics—Zagreb, Croatia. She received a doctorate in Economics from the University of Zagreb, Croatia, and completed the Guided Study Program in System Dynamics organised by the MIT System Dynamics in Education Project. Her present research work focuses on business dynamics. She is currently working on a research project devoted to exploring the potentialities of system dynamics modelling as a tool employed in management education.

# SIMULATION FOR UNDERSTANDING COLLABORATION IN A VIRTUAL PUBLIC COUNTER

Marijn Janssen  
Faculty of Technology, Policy and  
Management,  
Delft University of Technology,  
Jaffalaan 5, NL-2628 BX, Delft,  
The Netherlands,  
Tel. +31 (15) 278 1140,  
E-mail: [MarijnJ@tbm.tudelft.nl](mailto:MarijnJ@tbm.tudelft.nl)

Jaap Beerens  
D3K Simulations and Consultancy,  
Delftechpark 26, NL-2628 XH Delft,  
The Netherlands,  
Tel. +31 (15) 2600952,  
E-mail: [Jaap@d3k.nl](mailto:Jaap@d3k.nl)

## KEYWORDS

Simulation, collaboration, e-government, animation, decision support.

## ABSTRACT

*E-government is emerging as the new way for government to provide services to its constituents. Local government organizations have to collaborate with each other in a virtual public counter to offer a one-stop shop. Many hurdles need to be taken and decisions need to be made before virtual counter can be realized. Public servants are, however, often not aware of the collaboration problems they have to deal with when designing a virtual counter. In this research simulation is used as an instrument to make government officials aware of collaboration problems. In this paper a simulation study aimed at making government officials aware of collaboration problems they have to deal with when designing a one-stop shop public business counter is presented. First the background and nine collaboration issues are discussed. Thereafter simulation models are presented modeling these collaboration issues. The effectiveness of these models has been evaluated using a workshop and a survey. The models were found to be effective to widen the horizon of government officials and to focus on organizational issues of collaboration instead on a limited number of mainly technical oriented issues.*

## INTRODUCTION

Collaboration between government agencies is necessary for integrated services provision. Currently the service provision in the Netherlands has a highly fragmented nature. Politicians pay an overwhelming attention for more customer-oriented services provisions. Public administrations are urged by politicians to stay closer to citizens' every-day life, and act more proactively (Peristeras and Tarabanis 2000). In this respect, a virtual business counter should provide a one-stop shop for interacting with multiple agencies providing various kinds of services.

During the various years, there have been three pilot projects initiated to design an integrated, one-stop shop virtual counter. The benefits remain limited despite the

high ambitions and investments made in these projects. The implementations of the three virtual business counter remains largely limited to a website containing hyperlinks to the information and services on the websites of the organizations involved. The organizations feel that the problems are mainly organizational and not technical in nature. So far decision-making between the three parties has largely been focused on technical issues, like web-hosting, the layout of the webpage and which of the existing services should be presented on the webpage. This research is aimed at creating awareness of complex collaboration issues in a virtual business counter required for coherent service provisions using simulation. An important part of this research is the evaluation of the effectiveness of simulation, as the civil servants are not familiar with and used to simulation.

In the first part of this paper the main collaboration problems that need to be resolved in the business counter are investigated. We clustered the problem into nine issues. These issues are interdependent and should not be considered as classes that exclude each other.

1. Redefining and allocating roles and responsibilities;
2. Legal responsibility and liability;
3. Organizational structure;
4. Integration and ownership of services;
5. Position of the physical and virtual counters;
6. Range and types of services;
7. Service levels and quality assurance;
8. Customer trust and loyalty;
9. Use of Information and Communication technology (ICT).

In the second part of this study, simulation models of the current situation and of a situation with a virtual business counter are constructed. These models are made for creating awareness of the nine collaboration issues. The model with the virtual business counter is aimed at attracting attention to the collaboration issues and provides no solution for the collaboration issues open, e.g. no responsibilities are defined or showed, no data is integrated. In the last part the contribution of the simulation model for creating awareness of

collaboration issues is evaluated using a workshop and interviews. At the beginning and ending of the workshop the participants were asked to identify collaboration problems influencing the design of a virtual business counter. The insight gained during the workshop is evaluated by looking at the difference between issues.

## **COLLABORATION IN THE VIRTUAL COUNTER**

Collaboration is the reciprocal and voluntary agreement between distinct public sector agencies to deliver public services. At an institutional level a complex mixture of cooperation and conflict emerges when organization start collaborating across traditional organizational borders (Kumar and Dissel 1996).

Three types of organizations cooperate in the virtual business counter; municipalities, the regional Chamber of Commerce and the regional Dutch Taxes. Civil servants of the three organizations have little incentive to improve their services. During the period 1999-2002, central government has provided funding to create incentives necessary to create a virtual counter. Public organizations rarely change in any top-down manner (Andersen 2002). Therefore the organizations have been given the freedom to design their own virtual counter and three different pilot projects have been funded. In this way gain experience with an integrated public services provision to businesses is created. The business counter was aimed at accomplishing the following three goals.

1. The reduction of the administrative cost of businesses;
2. An increase of customer satisfaction of government service provision;
3. The increase of efficiency of government organizations.

The evaluation of these three project showed that the current structures of the virtual business counter often reflect the history of the organizations and only a small portion of the high ambitions are realized. The services provided can be positioned in the catalogues and transaction phase of Layne and Lee (2001). Overall, the three virtual counter projects have created a web-presence containing product information, there are some downloadable forms for a limited number of services and for a limited number of services it is possible to conduct online transactions. In the latter case, transactions are performed without any direct integration of front- and back-office applications and the information and services are provided using hyperlinks to the websites of each organization. The stages of horizontal and vertical integration of Layne and Lee, characterized by integration of information systems across different functions and departments, are still far away.

The evaluation of the pilot projects showed that the benefits obtained remain limited despite the high ambitions and investments made. The evaluation showed that mainly technical issues are addressed, whereas the organizations feel that the problems are mainly organizational and not technical in nature. Integrated service provision requires that various collaboration issues having an organizational nature are addressed. Before these can be addressed, participants should become aware of the collaboration problems.

## **COLLABORATION ISSUES**

A generic problem of virtual counter is that they are fairly difficult to operate as a virtual counter is a boundary- and function-spanning endeavor. One request and business process drives more than one organization. Individual organizations are expected to do what is best for the virtual counter, this is not necessarily what is best for them. Information must be made available to the right companies and somebody has to be responsible and given the means to control the status and progress of processes. The current problem is that there is no structured way to provide insight into collaboration issues surrounding the realization of the virtual counter. Insight is necessary to create awareness of the issues and to make them discussable. Only thereafter they can be addressed and solved. Simulation should support creating insight into the nature of the complex collaboration issues that are discussed hereafter.

### **1. Redefining and Reallocating Roles and Responsibilities**

Collaboration requires rearrangement of organization roles and responsibilities and identifying new responsibilities. In the pilot projects responsibilities are assigned by trial-and-error experiments. A more fundamental approach is necessary where roles and responsibilities are identified and assigned consciously.

New cross-organizational roles and responsibility need to be established, e.g. who is responsible for monitoring service levels, who initiates action to increase service levels. Organizations need sound organizational procedures, such as requiring the consent of two individuals before changes can be made influencing more than one record.

### **2. Legal Responsibilities and Liability**

Many of the organizations roles are founded in laws and regulations. These organizations have a legal duty and are responsible for services provisioning. Laws and regulations often block collaboration in a virtual counter. For example, the Dutch taxes have the duty to collect sales taxes, which blocks the road for collecting sales taxes using another legal entity. It is unclear

which functions might be executed by a virtual counter and who is liable.

### 3. Organizational Structure

Each government organizations have already created an own website in the past. In the pilot project no new organizational structure has been created. The organizations who took the initiative, the chamber of commerce, established a website merely consisting of hyperlinks to the individual websites of the three government organizations involved. A virtual counter consists of content available at other websites in the pilot. There are other structures possible which could resemble a network organization (Alstyne 1997). In a fully virtual counter the processes are arranged in such a way that the virtual counter is experienced as one organization by its' customers. They would no be aware which organization has provided a service. Processes are dynamically established and seamlessly integrated across organizations. This collaboration issue should draw the attention to the need to organize the virtual counter as a network organization.

### 4. Integration and Ownership of Services

In the pilot projects services are still provided by each organization and no interdependent services were developed. Three forms containing similar data are used to request services from the three organizations. A virtual counter provides the opportunity to use one form to request services from all three organizations. The use of one form gives rise to the ownership question. Various kinds of conflicts exist with the integration of services.

- Ownership of the data in the authentic registries;
- Ownership and control of the network connecting the authentic registries;
- Control and ownership of the web-applications.

Data is treated by the organization as a valuable asset. The collaboration faces issues about the ownership of the rights of data. The Dutch government uses the principle of *authentic registrations*. This principle states the organization who gathers the information at the sources, is responsible for keeping information up-to-date and for distributing the organizations to other organizations. The core-business of government is often based on the information in these authentic registers. The services currently provided by a municipality are only provided by that municipality because it owns, controls and maintains the citizens authentic register. When another organization, like the virtual counter, would control and maintain the citizens authentic register they might lose their right to exist. Simulation should draw the attention to this issue.

### 5. Position of the Physical and Virtual Counters

Currently the three organizations have their own physical business counter. A virtual counter demands

the support of a physical counter, as often consults are necessary to support the provision of services. Currently the chamber of commerce provides consults to entrepreneurs, as this is part of their core business. The tax organization only provides consults incidentally, as they are primarily responsible for collecting taxes. The municipality provides consults to certain industries, e.g. they advice about safety measures to the hotel and catering industry.

The operating of the virtual business counter should take into account that consults might be necessary. The virtual counter should make the correct reference to the physical counter of each individual organization or to the possible one-stop physical counter. Simulation should provide insight into the different possible positions of and relations between the virtual and physical counters for providing a one-stop shop. In this way this collaboration issue should become part of the discussion.

### 6. Range and Types of Services

Service provisioning is very limited and has a low level of integration in the pilot projects. The services brought online are some of the services currently provided by the individual organizations and did not include services requiring the collaboration between the three organizations. As such the virtual counter is only seen as new channel for making a selection of existing services online, instead of making all services online and providing new, innovative and/or integrated services.

A simple new service that solves the customer problem of entering similar data multiple times was not solved by two of the three pilot projects. Data can be re-used by using a virtual counter and collecting data centrally. Another example of a new service could be the simultaneous registration by entrepreneur in the trade registry of the chamber of commerce and in the tax register of Dutch taxes. Providing progress and status information about the status of a services is more complicated, as the status is determined by individual organizations. In the example, information about the status of the registering procedures of both organizations should be collected and appropriate measures should be taken when registering takes too long or the resubmitting of the registering request when information is loss. This collaboration issue should make the case for new types of services.

### 7. Service Levels and Quality Assurance

Each organization executes its own processes and has its own responsibilities. The relationships between the collaborating parties are interdependent and need to be specified. This quality assurance can be achieved by service agreements levels (SLAs) or contracts between organizations. SLAs express that service of a certain level has to be provided (Looijen 1998). SLAs can

express criteria like availability, access to each other's data, response time, reliability etc.

### 8. Customer Trust and Loyalty

A (virtual) business counter provides a new way for businesses to deal with the government. Decisions made by the government can have a profound effect on the management of businesses. The careful and skillful handling of business- interactions can influence the success of the business counter. Trust plays an important role in the development of electronic relationship (Hoffman et al. 1999). As the businesses have experience with the municipality, taxes and chamber of commerce and no or limited human interactions are possible within the virtual counter the creation of trust is essential.

### 9. Use of Information and Communication Technology

The use of ICT is about technical issues like the web server needed for hosting the website, the design of the web pages, the maintenance of hyperlinks, the protocols for exchanging data and the use of standards. Technology standards affect the threshold to participate. The choice of standards can affect the costs of organizations negatively. When an existing standard of an organization is chosen and another organization do not support it yet, high costs can be necessary to comply to these standards. Most of these issues except standardization are already addressed by the three pilot projects.

### SIMULATION OF THE VIRTUAL COUNTER

The organizations involved in establishing a one-stop shop virtual counter needs support to be able to determine what make up a customer-oriented virtual counter. The *raison d'être* of businesses is to make money and the profitability can be compared, however, this is not the case for government. The local government organizations have no strong incentive to improve their service provision to increase customer satisfaction. For improvement a prerequisite is that they agree on the decisions. As a result they need insight into how collaboration can be best done from the customer perspective *and* their own perspective. The participants should share a long-term commitment and manage their interwoven relationship. The interwoven dependencies are time-dependent and should be dynamically visualized to identify collaboration problems.

*Simulation* of business processes constitutes one of the most widely used applications of management science / operational research, as it allows for understanding the essence of business systems, identifying opportunities for change, and evaluating the effect of proposed changes on key performance indicators (Law and Kelton 1999).

Lack of communication has been linked to numerous project failures (Pinto and Pinto 1999). An essential ingredient of communication is the use of visualizations to support communication between all kinds of stakeholders, investors, management, information architects, designers and programmers, many lacking technology knowledge. Visualization can help to translate the outcomes of the model to soft explanations, conclusions, recommendations, and requirements (Vreede and Verbraeck 1996). As such visualization can be used for critical debate about the fit between information system and organizational processes. Visualization support is often a standard feature of simulation. A visualization model is a graphical model of an empirical model of a problem situation. The main purpose of *visualization* is to facilitate the process of acquiring insight into the dynamic interactions of the modeled system, and to facilitate communication between parties involved in a dynamic modeling study (Vreede and Verbraeck 1996).

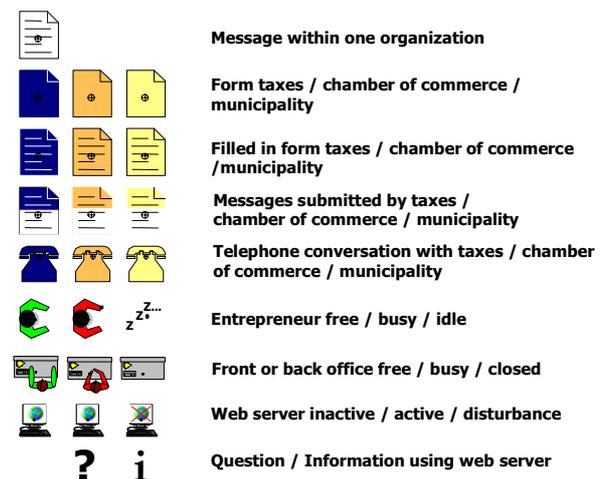


Figure 1: Explanation of Symbols

The *process* paradigm implies a way of looking at organizations based on the processes they perform rather than the functional units, divisions or departments they are divided into. In the current organization the three local government organizations can be viewed as functional units, each with a highly specialized set of responsibilities and expertise. Even the simplest business tasks tend to cross the functional units and require the coordination and cooperation of different functional units. The virtual counter should support cross-organization processes and simulation should visualize processes instead of functions and responsibilities. Only after the participants are aware of the processes necessary, tasks and responsibilities should be assigned to organizations. It is thus essential that in the simulation models the processes of a virtual counter will not be viewed as belonging to an organization. We do not want to give normative statements about ownership and responsibilities.

To support insight into the collaboration issues two simulation models were developed. Three types of services were incorporated in the models that were considered as suitable for providing insight into the collaboration issues. The explanation of symbols is shown in figure 1.

The simulation model of the 'as is' or 'current' situation should make the organizations aware of the need for integrated service delivery. A screenshot of the simulation model of the 'as is' situation including an explanation is shown in figure 2. At the bottom of

the figure the back-offices of the three government agencies, municipality, chamber of commerce and taxes, are shown. At the right side of the back-offices the status of the requested services are shown. In the middle the post office is shown, with acts as an intermediary between the entrepreneurs and government agencies. At the right side of the counter the activities performed are written down. At the top three entrepreneurs having a need for services from the virtual counter are modeled. Also miscommunication, mistakes in data and processing faults are modeled.

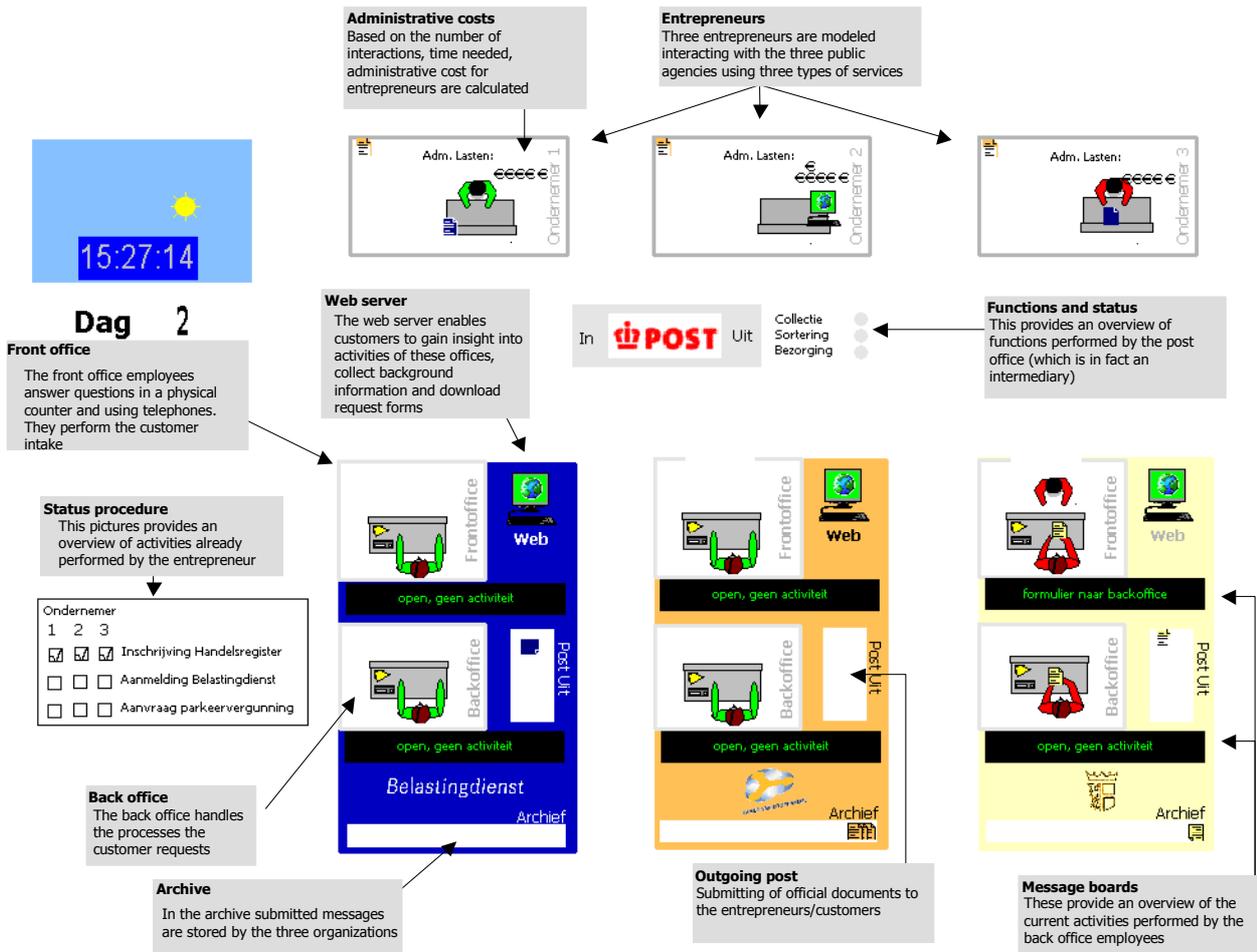


Figure 2: Screenshot of the 'as is' Model

The simulation of a virtual business counter and a physical business counter should make people aware of the nine collaboration issues. The simulation model of the virtual and physical counter is shown in figure 3. In the middle of this figure the physical and virtual counter are shown with make up the one-stop virtual business counter. At the right side the activities performed by the one-stop business counter are shown. Which government agency is responsible for the one-stop business counter is deliberately not included in the animation, as the aim is to start up a discussion about responsibilities and structures, e.g. collaboration issues one and three.

A number of functional components making up the application architectures are shown as well. The identification component is used to identify entrepreneurs as well as government employees. This could also be extended to a customer relationship management (CRM) system, to register all customer data, all interactions with the customer and to collect customers' information. From a customer perspective this could be extended a digital safe to store and reuse companies data. Apart from identification, an essential characteristic of this component is that data can be entered once and reused.

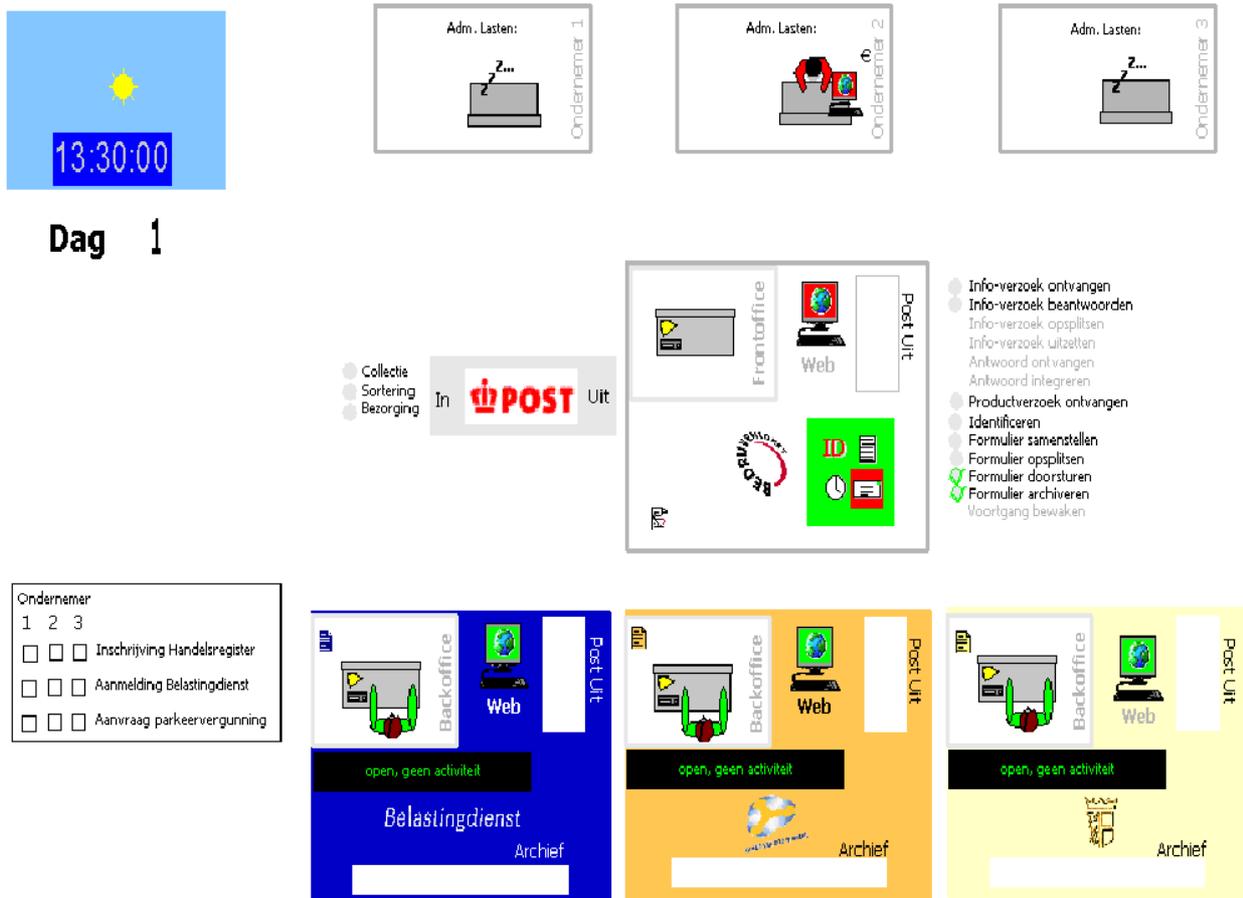


Figure 3: Screenshot of the 'to be' Model

Another component is an agenda system that can be used to let customers schedule an appointment with a government representative. Employees can provide data about their availability and entrepreneurs can make appointments. Long queues before the physical offices can be avoided by reserving time-slots prior to arrival.

The web server component is responsible for the communication between humans and information systems and can also include process orchestration and the tracking of status information of a service delivery. The messaging component is responsible for exchanging data between information systems. In a general sense, legacy information systems can communicate with other systems using messaging. The communication with state-of-the-art components can be done using web-services. The messaging component can translate data formats into other data formats and can asynchronously exchange data based on message queuing. When an information system is not available the messaging component can queue up messages and submit the message at another time when the receiving component is available. The data formats are stored in the messaging application, so that control and maintenance of these formats can be done at one (central) location.

## EVALUATION

The simulation model should help the participants to address the right collaboration issues while designing an integrated business counter. The evaluation aims at evaluating the effectiveness of the use of the simulation models discussed in the preceding section to create awareness of the nine collaboration issues. Our evaluation neglects the quantitative part of a simulation study. We want to prove that more insight is gained into collaboration issues using simulation than without using simulations. Checkland (1981) highlights the problem of proving that a support environment is better than without, but concludes that it cannot be proved. He suggests that two development teams could work independently on developing a system; however, the very fact that there are two sets of developers will undoubtedly influence the results. As a result Checkland concludes that we can only make plausible that our approach is 'better'. We follow Checkland and evaluate the effectiveness by looking at the collaboration issues that are known by the participants before and after the showing of the simulation models during workshop. This means that we are not able to measure a difference when the participants already

address the right issues and the simulation models do not bring about a shift into the focus on collaboration issues. We do not consider this as a major problem, because as the right collaboration issues are already addressed simulation models have no added value. The construction is then an extra effort that could have been avoided.

The models will be evaluated using two instruments, a workshop and a survey. In the evaluation workshop participants are asked to identify collaboration issues at the beginning and ending of the session. During the session the simulation models are shown and discussed. The difference or shift in issues tells us something about the effectiveness of the models. The survey is performed after the session and is used to collect the perception of the workshop participants about the different aspects of the added value of the simulation models.

In figure 4 the design of this whole research project, including the workshop is shown. For the purpose of clarity the first four steps, which are already discussed in the preceding part of this paper, are shown. First the existing situation was analyzed, than an 'as is' model was build, thereafter collaboration issues were identified and ultimately 'to be' simulation models were build. Thereafter the evaluation workshop and survey was performed.

### Evaluation Workshop

The workshop is aimed at evaluating the effectiveness of simulation models by comparing the collaboration issues identified at the beginning and ending of the workshop. The workshop lasted one morning and was part of a one-day session about the design of a virtual business counter. A group of participants that were not involved in the design of the simulation models were invited to take part in a workshop. This group consisted of 10 participants involved in policy formulating of a virtual business counter and representing the tax organization, chamber of commerce and municipalities.

The first step of the workshop was the introduction including some background about the purpose of the session. During the second step the participants were asked the question to 'identify collaboration issues influencing the design of a virtual business counter to large extends'. This resulted in a number of collaboration issues perceived as being important to address in the design of a virtual counter. In the following step the simulation models of the 'as is' situation and 'to be' situation were presented and based on these model a plenary discussion was held. At the end of the workshop the participants were again asked the question to identify collaboration problems influencing the design of a virtual business counter to a large extends.

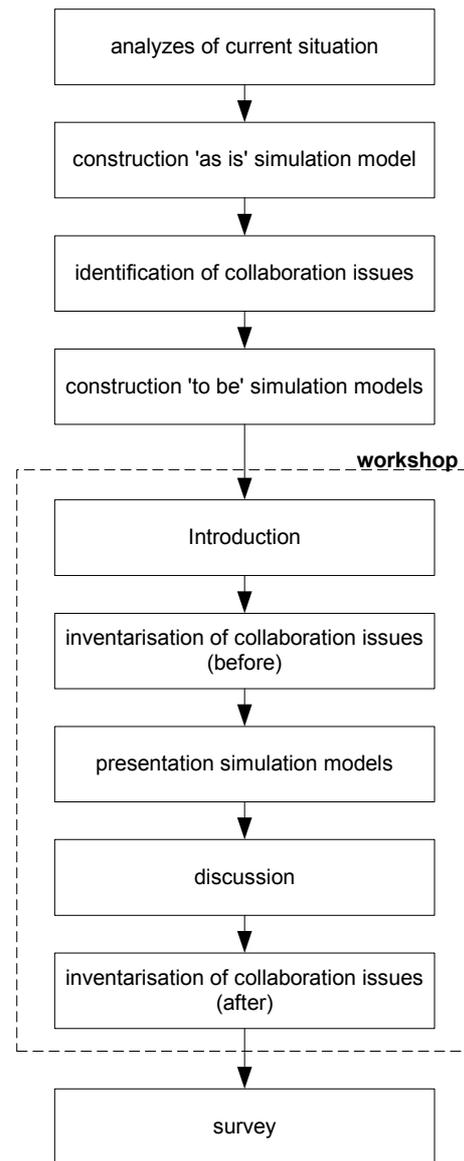


Figure 4: Workshop Design

The problems mentioned at the beginning and ending of the workshop were mapped to the nine collaboration issues discussed in the preceding part of this paper by the researchers. The main reason for this was to make the issues mentioned at the beginning and ending of the workshop comparable and to avoid spoon-feeding the participants. At the beginning of the workshop, they need to identify issues with their existing knowledge. For the purpose of mapping the comments on the collaboration issues we had to make some interpretations and abstractions, e.g. the response time is mapped to the collaboration issues 'service levels and quality assurance'. The results of the questions ask at the beginning and end of the workshop are shown in table 1.

At the beginning of the workshop the participants came only to a number of three issues. This limited number provides indications for the need for simulation models providing insight into collaboration issues. After the

workshop the participants generated six collaboration issues. Note that the issue ‘use of information and communication technology’ was mentioned at the beginning of the workshop as an issue influencing the design of a virtual counter, whereas this issue was not mentioned at the end of the workshop. When taking this into account, the participants became, the participants became aware of four extra collaboration issues, instead of becoming aware of three extra collaboration issues. In their opinions the ICT issue was not found to be an important collaboration issue influencing the design of the design of virtual counter. The new issues were ‘Redefining and -allocating roles and responsibilities’, ‘Integration and ownership of services’, ‘Range and types of services’ and ‘Customer trust and loyalty’.

The participants did not found three issues important enough. The issues ‘organizational structure’ and ‘synergy between physical and virtual counter’ were not mentioned at the beginning and at the ending of the workshop. During the informal discussions after the

workshop some participants indicated that these issues were not mentioned, because they already seem clear to them. The current organizational structures used in the pilot projects were not suitable and ideal seemed to be the establishment of a new organization consisting of both a physical and virtual part. The collaboration issue ‘position of the physical and virtual counters’ was already paid a lot of attention to. The participants were aware that operating of the virtual business counter needs also a physical counter parts as customer have a need for consults. Currently the chamber of commerce provides consults to entrepreneurs and the taxes and municipalities hardly provides any consults. The tax organization even does not want to provide consults, as they consider this the field of tax counselors. The municipalities are primarily interested in providing better services and reducing their administrative burden. Consequently the participants seem already to have agreed that the chamber of commerce should take care of operating an integrated business counter and providing a physical counter.

Table 1: Insight gained by simulation

Collaboration issue	Before models	showing	After models	showing
1. Redefining and -allocating roles and responsibilities			✓	
2. Legal responsibilities and liability	✓		✓	
3. Organizational structure				
4. Integration and ownership of services			✓	
5. Position of the physical and virtual counters				
6. Range and types of services			✓	
7. Service levels and quality assurance	✓		✓	
8. Customer trust and loyalty			✓	
9. Use of Information and communication technology	✓			

When comparing the issues mentioned at the beginning and end of the workshop in the light of evaluation of the effectiveness of simulation models to create awareness, it becomes clear that the participants are aware of more collaboration issues. Instead of the three issues they were aware of at the beginning of the workshop, the participants were at least aware of six of the issues. Additional interviews showed that the participants were aware of the three other issues, however, considered them as less influencing the design of a virtual counter. Most remarkable is that the use of ICT is mentioned as a collaboration issues at the beginning of the workshop, but is not viewed as an important collaboration issue at the end of the workshop. With the necessarily care we conclude that a broadening of the horizon to a focus on more organization-oriented issues has appeared. We have strong indications that our simulation models are effective instruments for creating awareness of collaboration issues in the virtual business counter.

### Survey

After the workshop, the participants were asked to fill in a survey, containing questions about the usability and the added value of the ‘as is’ and ‘to be’ simulation models for each participant and for the contribution to the process of implementing the virtual public counter. Participants were asked to indicate in what degree they agreed or disagreed to each statement by giving each statement a score from 1 (strongly disagree), 2 (disagree), 3 (neutral), 4 (agree) to 5 (strongly agree). They could also add comments to support the selection of their remark.

The survey consisted of nine statements. These statements are based on the two main aspects (1) the suitability of the simulation models as communication instrument (questions 1 till 3) and (2) the contribution to the insight into the collaboration issues (questions 4 till 9).

### Suitability as Communication Instrument

- 1) The lay-out used in the animations is understandable.

- 2) The way business processes are performed is understandable.
- 3) In my opinion, the use of animations is of valuable importance to the decision making process on the virtual public counter.

**Insight into Collaboration Issues**

- 4) The animation ‘as-is’ provides an accurate illustration of the situation ‘as-is’ in reality.
- 5) The animation ‘to-be’ provides a realistic illustration of a possible future situation.
- 6) The animation ‘to-be’ resembles my vision on the virtual public counter, before having seen the animations.
- 7) The animations have changed my vision on the virtual public counter.
- 8) After having seen the animations, I think realization of the virtual public counter will be more complex than I thought before.
- 9) The animations have pointed out at least one issue that I did not have in mind before.

The nine statements and the mean score for each statement is shown in the table below. Where 1 is the lowest and 5 is the highest score.

Table 2: Survey Results

Statement	Score
<b>Suitability for communication</b>	
1. Understandable lay-out	3.7
2. Understandable processes	3.9
3. Animation valuable tool	3.6
<b>Insight into collaboration issues</b>	
4. Realistic illustration as-is	3.7
5. Realistic illustration to-be	2.8
6. To-be resembles own vision	2.7
7. Vision has changed	2.8
8. Realization virtual counter more complex	2.7
9. New issues identified	3.2

When looking at the suitability as communication instruments the participants are positive about the simulation and animations models. The participants are positive about the understandability of the lay-out used in the simulation models and the business processes depicted by the model and found animation to be a valuable tool. In the first and third statements there was one neutral and one disagree score. It seems that those persons had trouble with understanding the layout of the animation models as shown if figure 1 and 2. This cause of this score could be due to a bad explanation process or due to the unclear models. We were not able to identify the main cause.

When looking at the insight created into the collaboration issues, the participants were positive about the realistic illustration of the ‘as is’ situation. Only the persons who have had troubles with

understanding the layout out and business processes had selected the neutral and disagree option. What is striking is that the participants consider the animation of the ‘to be’ situation to be less realistic (statement five). When looking at the detailed data, we found that two participants disagree strongly with this statement. From the participants view the ‘to be’ models were too abstract and have less relation to reality. For example they found the choice to make no organization responsible for execution the physical and virtual counter not realistic. However, no organization was deliberately made responsible for executing the virtual counter, as the aim was to make participants aware of collaboration issue one, redefining and -allocating roles and responsibilities.

Another explanation for this low score on statement four can be found when looking at the even lower average score for statement six. It is likely that participants judge negatively on the contribution to the insight of the animation because the animation of the ‘to be’ situation does not resemble their own vision close enough.

When looking at the overall indication it seems that the participants were positive about the models understandable, however, they were less positive about the ‘to be’ models as the models were abstract, did not resemble their vision.

Statements seven and eight score also just below a neutral score. The goal of the workshop was not to change somebody’s vision on the virtual public counter, moreover, the ‘to be’ model was made with the idea to have no or as little as possible normative resembles with some kind of implementation or vision. The score of statement 8 can be explained when looking at the starting points of this simulation projects, the implementations of the three pilot projects remained limited to a website containing hyperlinks to the information and services on the websites of the organizations involved.

The average score of statement 9, the number of new collaboration issues, identified is above average. Most participants did identify new issues concerning the virtual public counter and after all, the use of simulation was evaluated as valuable; none of the participants disagreed with this statement. This outcome seems to be conforming the outcomes of the evaluation workshop shown in table 1.

Informal communication after the workshop learned that the participants were quite positive about the use of simulation. One participant noticed that he would like the models to be capable of handling more scenarios and various visions. One participant stated that the models rather support the creation of awareness and by doing so support the decision making process. Another participant would like to see more detail in the animation of the ‘to be’ situation to point out the different functions of the virtual public counter.

## CONCLUSIONS

In this research awareness of collaboration issues in a virtual business counter for integrated service provisioning was created using simulation models. From a government perspective the use of simulation should provide a vehicle to realize a higher level of ambition and from a design perspective the use of simulation models should help to make stakeholders aware of collaboration issues and in this way help them to focus on the right issues.

First collaboration issues were identified, thereafter, simulation models were constructed aimed at providing insight into these issues. The developed models were evaluated on effectiveness to create awareness of the nine collaboration issues using a workshop. This workshop showed that simulation models broadened the scope of the participations and helped to create a shift from an emphasis on technical issues to an emphasis on more organizational-oriented issues. Our simulation models seem to be effective instruments for providing insight into collaboration issues of the virtual business counter. The models helped to widen the horizon of government officials and to focus on organizational issues of collaboration instead on a limited number of mainly technical oriented issues. As such simulation keeps its promises and proved to be a valuable instrument for providing insight.

Now the participants are aware of the collaboration issues, further research can address the creating of a shared vision on the concept of a virtual counter. Various scenarios and a quantitative evaluation of each scenario to compare the added value can support decision-making. Another issue that needs to be addressed is the link to information and communication technology. The simulation models should also include the application architecture, the user-interactions and the interactions between applications to enable a growth path from an information and communication technology perspective.

## REFERENCES

- Alstynne, M. van 1997 "The State of Network Organization: A Survey in Three Frameworks." *Journal of Organizational Computing and Electronic Commerce*. 7, no. 2/3, 83-152.
- Anderson, K.V. 2002. "Public Sector Process Rebuilding Using Information Systems." In; Traunmüller, R. and Lenk K. (eds). *EGOV 2002*, pp. 37-44.
- Beerens, J. 2003. *Dynamiek in Dienstverlening*. Master Thesis, Delft, University of Technology, Delft, The Netherlands.
- Checkland, P. 1981. *Systems Thinking, Systems Practice*. Wiley, Chichester.
- Galbraith J.R. 1995. *Designing Organization. An executive briefing on strategy, structure, and process*. Jossey-Bass, San Francisco.
- Hoffman, D.L., Novak, T.P. and M. Peralta. 1999. "Building Consumer Trust Online." *Communications of the ACM*. 42, no. 4, 80-85.
- Kumar, K. and H.G. van Dissel. 1996. "Sustainable Collaboration: Managing Conflict and Collaboration in Inter-organizational Systems." *MIS Quarterly* 20, no. 3, 279-300.
- Layne, K.J.L. and J. Lee. 2001. "Developing fully functional E-government: A four stage model." *Government Information Quarterly* 18, 122-136.
- Law, A.M. and D.W. Kelton. 1999. *Simulation Modeling and Analysis*. McGraw-Hill, New York.
- Looijen, M. 1998. *Information Systems: Management, control and maintenance*. Kluwer Business Information, The Netherlands.
- Peristeras, V. and K. Tarabanis. 2000. "Towards an Enterprise Architecture for Public Administration using a Top-down Approach." *European Journal of Information systems*. 9, no. 4, 252-260.
- Pinto, M.B. and J.K. Pinto. 1999. "Project Team Communication and Cross-functional Cooperation in New Program Development." *Journal of Product Innovation management* 7, 200-212.
- Vreede, G.J. de and A. Verbraeck. 1996. "Animating Organizational Processes: Insight eases change." *Simulation Practice and Theory* 4, no. 4, 245-263.

## AUTHOR BIOGRAPHIES

**DR. MARIJN JANSSEN** is an assistant professor in the field of information systems and government at the section of Information and Communication Technology at Delft University of Technology. He has been a consultant for the Ministry of Justice and received a Ph.D. in information systems. His main research interests are the design and modeling of intelligent, information architectures in networks of organizations.

**JAAP BEERENS** is a partner at D3K Simulations and Consultancy. His company specializes in the development of simulation models in various research environments. He graduated at Delft University of Technology in Systems Engineering and Policy Analysis.

# TOWARDS E-GOVERNMENT - THE ROLE OF SIMULATION MODELING

Jurij Jaklič, Aleš Groznik, Andrej Kovačič  
University of Ljubljana, Faculty of Economics  
Kardeljeva pl. 17, Ljubljana, Slovenia  
E-mail: {jurij.jaklic, ales.groznik, andrej.kovacic}@uni-lj.si

## KEYWORDS

Business Process Reengineering, Discrete-Event Simulation, E-government, Process Modelling, Process Renovation, Simulation Modelling

## ABSTRACT

The main goal of the paper is to present the characteristics of business renovation efforts, readiness for e-government in Slovenia, and how the simulation modelling can be used for these purposes. It is clear that successful e-government implementation requires not only introduction of modern information technology, but also business renovation, business process reengineering and e-business strategy. The case of business renovation project in one of the Slovene Ministries, where the process modelling and simulation were extensively used, is also presented. The simulation modelling proved useful since it shows the process as a whole, drawbacks of the existing process, bottlenecks in the process execution, provides critical insight into process execution etc. The results of the simulation modelling are a good foundation for a business process reengineering as a next step towards e-government.

## 1. INTRODUCTION

E-government is the execution by electronic means of interactive, inter-organizational processes and represents a shift in business doctrine that is changing traditional organizational models, business processes, relationships and operational models that have been dominant in the public sector in the past decades. The new doctrine of e-government requires organizations to integrate and synchronize the strategic vision and tactical delivery of services to its clients with the information technology and service infrastructure needed to meet that vision and process execution. In the next few years, successful countries will restructure their public sector, process and technology infrastructure for successful e-government execution.

Past experience in introducing e-government in the most developed countries (Singapore, Canada, Australia, New Zealand...) in this field has shown us that the root of the problems, which have to be solved in introducing e-services, has moved from the technological into the organizational and process domain (Government Centre

for Informatics, 2002). The essence of e-government is to radically change the ways and mechanisms of operating administration and, as a result, also basic principles, on which these mechanisms have been developing in the last decades or even centuries. Therefore, the business renovation (BR) or business process renovation methods should be used in the framework of e-services introduction.

The simulation of business processes is suggested for use in BR projects as it allows the essence of business systems to be understood, the processes for change to be identified, process visions to be developed, new processes to be designed and prototyped and the impact of proposed changes on key performance indicators to be evaluated (Greasley and Barlow 1998). The reasons for the introduction of simulation modelling into process modelling can be summarized as follows: simulation allows for the modelling of process dynamics, the influence of random variables on process development can be investigated, re-engineering effects can be anticipated in a quantitative way, process visualization and animation are provided, and simulation models facilitate communication between clients and an analyst. The final reason for using simulation modelling is the fact that it can be increasingly used by those who have little or no simulation background or experience (Irani et al. 2000).

In our paper a case study of simulation modelling usage in the field of e-government enrolment in Slovenia is presented. In the Section 2 the role of BR in current efforts and plans for e-government enrolment in Slovenia are presented, followed by Section 3 in which the value of simulation modelling in BR projects are briefly discusses. The main part of the paper is Section 4, where the case business renovation project in one of the Slovene Ministries is presented. Benefits and problems of the simulation modelling usage in such projects are analysed and presented.

## 2. THE E-GOVERNMENT STRATEGY IN SLOVENIA AND BPR

By adopting the "Strategy of E-commerce in Public Administration for the Period 2001-2004, SEP-2004" (Government Centre for Informatics 2001), in February 2001, the Government of Slovenia has defined the primary strategic orientations for the next essential phase of informatization of public administration, which

is the development of e-government. As a result, Slovenia is following a number of most developed European countries, which are approaching the accelerated development of e-government in a similar way.

Although, as a result, Slovenia has started a new developmental cycle of technological modernization of administration and have launched a number of new projects, we have concluded that development is not progressing as planned and expected. This is not only a problem in Slovenia, but based on analyses carried out in EU, also a problem in mostly all other countries (Government Centre for Informatics, 2001). After a year or two, we can see that in most countries it was relatively easy to achieve the first (information) stage, which refers to the introduction of information services, as this step does not require specific changes in internal operations of administration and in business processes and procedures. (Government Centre for Informatics 2001). Much more complex is the introduction of more demanding, so-called transaction services, which enable all phases of a selected administrative procedure or process to be executed electronically. As a rule, this requires a complete renovation of administrative operations, internal business processes and procedures, the integration of registers and public databases, the alteration and completion of material legislation and the development of new organizational regulations, classifications, and standards. At this point, the development of e-government in most developed countries has come to a standstill, which is evident from viewing web portals of these countries. We can find very little transaction services. The same has also occurred in Slovenian public administration (Government Centre for Informatics, 2001).

Problems, which need to be solved as soon as possible, are, in a minor sense, of technological nature. They predominantly extend to the internal renovation of administration operations, its reorganization, greater process orientation and close coordination and cooperation among various departments, and even branches of power (executive, legislative, and also judicial). It has to do with deep structural changes in the operation of administration, which will be successfully and quickly implemented only with a total and well-considered approach, as used in the modernization and reformation of administration up to the present. BR projects should be focused on all related key business elements: business processes, people and finally the technology. E-government is not only enabling the redesign of internal organizational processes, but is extended into inter-organizational processes.

Within the framework of development of a new "organizational paradigm", which will be based on the operation of e-government, all State Bodies and other institutions from the public sector will have to analyze in detail all (action and other) administrative procedures

and processes and renovate them in accordance with defined starting points and principles of development of e-government, and the possibilities that information technology can offer as soon as possible (Government Centre for Informatics, 2002).

BPR is an organizational method demanding radical redesign of business processes in order to achieve better efficiency, quality and more competitive production (Hammer and Champy 1993). It is also a method of improving the operation and therefore the outputs of organization (Kettinger and Grover 1995). It means analyzing and altering the business processes of the organization as a whole. BPR was first introduced in a research program at MIT (Massachusetts Institute of Technology) in the early nineties. BPR was the buzzword of the mid-1990s, and although there were plenty of successes, there were many more failures (Hammer and Champy 1993). To many, BPR remains a dirty word, bringing back memories of head count reductions, budget cuts, facility closures, expensive consulting engagements and endless reorganizations that destroyed morale and confused employees, partners and customers. By the time it was recognized that successful BPR required careful change management, the damage was done.

Many leading organizations have conducted BPR in order to improve productivity and gain competitive advantage. A study by Dhaliwal (1999) showed that about 50% of firms surveyed in Singapore (in some cases comparable to Slovenia) were engaged in BPR projects, with 37% of the firms indicated their intention to take up BPR projects in next few years. However, regardless of the number of companies involved in re-engineering, the rate of success in re-engineering projects is slightly over 55%, however, European organizations are less successful with the average success rate less than 50% (Al-Mashari et al. 2001). Some of the frequently mentioned problems related to BPR include the inability to accurately predict the outcome of a radical change, difficulty in capturing existing processes in a structured way, shortage of creativity in process redesign, the level of costs incurred by implementing the new process, or inability to recognize the dynamic nature of the processes.

### **3. BPR THROUGH SIMULATION MODELING**

Simulation has an important role in modelling and analyzing the activities in introducing BPR since it enables quantitative estimations to be made on the influence of the redesigned process on system performances (Bhaskar et al. 1994). Many different methods and techniques can be used for modelling business processes in order to give an understanding of possible scenarios for improvement (Ould 1995). IDEF0, IDEF3, Petri Nets, System Dynamics, Knowledge-based Techniques, Activity Based Costing and Discrete-Event Simulation are only some examples

of business process modelling techniques widely used (Eatock et al. 2000). As noted by (Hommes and van Reijswoud 2000) the increasing popularity of business process modelling results in a rapidly growing number of modelling techniques and tools. However, the majority of simulation software implements a model using the discrete-event method.

In Kettinger et al. (1997) an empirical review was made of existing methodologies, tools, and techniques for business process change. The authors also developed a reference framework to assist the positioning of tools and techniques that help in re-engineering strategy, people, management, structure, and the technology dimensions of business processes. However, relevance is far more important than completeness (Davenport and Prusak 1998) and simple models are far more understandable for non-specialists. Process modelling tools must be capable of showing interconnections between the activities and conducting a decomposition of the processes. These tools must help users to conduct “what-if” analyses and to identify and map no-value steps, costs, and process performance (bottleneck analysis). They should be able to develop AS-IS and TO-BE models of business processes. They must be validated and tested before implementation.

Some of the benefits can be directly evaluated and predicted, but others are difficult to measure (intangible benefits). This research investigates some of the benefits and outcomes of introducing new processes (time savings, workload reduction and increased throughput) that could be measured in advance by simulation modelling.

#### **4. BUSINESS RENOVATION PROJECT AT THE MINISTRY OF EDUCATION, SCIENCE AND SPORT**

The Business renovation project at the Ministry of Education, Science and Sport (Ministry) started due to internal and external factors. Internal factors that caused business renovation were the integration of two ministries, Ministry of Education and Sport and Ministry of Science and Technology into the Ministry of Education, Science and Sport, versified business processes that were not well defined and duplication of activities. Externally, the project has been stimulated by the Slovenian government that started the anti-bureaucratic program on the governmental level. The goal of the program is, according to Action Plan E-government Up to 2004 (Government Centre for Informatics 2001), to remove inefficiencies in business processes, to change organizational structure and to introduce suitable information technology that will support renewed business processes.

The Business renovation project has three main phases:

- identification of key business processes and their modelling,
- analysis of key business processes on the basis of their models; and
- modelling renewed processes and proposing organizational changes.

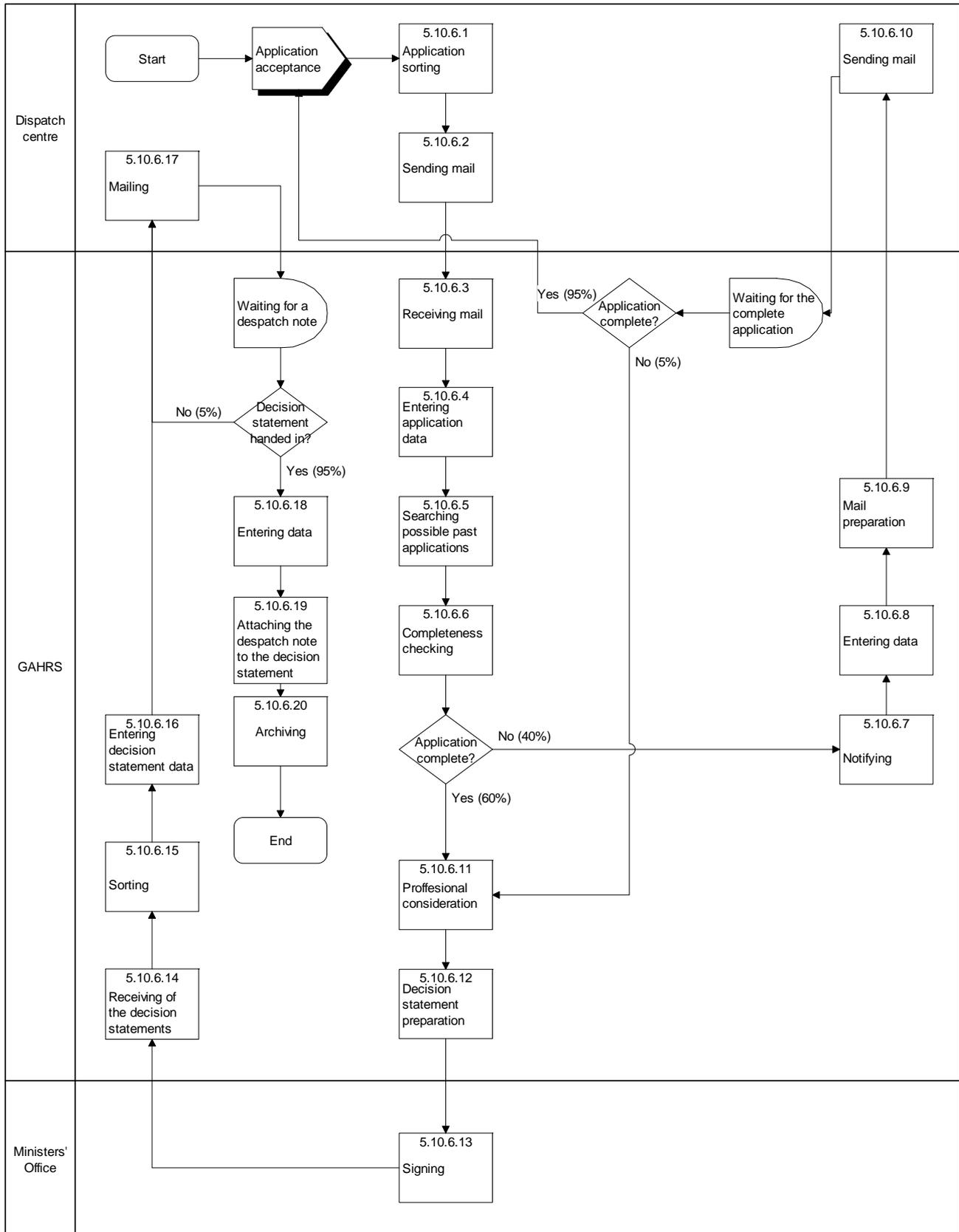
The project started with formation of project group consisted by members from the Ministry and consultants from Business Informatics Institute (BII), Faculty of Economics, Ljubljana. In the workshop, five key business process groups were identified by discussion and brainstorming: (1) Strategic planning, (2) Working program preparation, (3) Laws and provisions preparation, (4) Financial processes, and (5) Administrative processes.

The processes were modelled by interviewing people from the Ministry who perform the activities. This phase of the project was very difficult and lasted for almost six months and models had to be changed several times. Since the scope of the project is too big for the presentation in the paper, only a fragment, subprocess Promotion of the employees in education to a higher professional title of Administrative processes at General Affairs and Human Resource Service, will be shown in the next section.

In this study, iGrafx Process software was selected as the tool for business process and simulation modelling using the Process Maps technique. Process Maps are commonly used by many organizations, especially for business process analysis and modelling. They represent the standard modelling and analysis method for enterprise engineering and support particular reengineering activities such as simulation modelling. One of the major advantages of Process Maps is that little training is required for people to create and evaluate the process models (Chen 1999). Process Maps used by iGrafx Process provide a graphical interface to a behavioural modelling system, which requires no knowledge of a programming language; even unskilled people in business process modelling can easily understand and use this technique. Another major advantage of this technique is that it helps to identify the crossing of organizational boundaries, as it shows which organizational unit is responsible for each activity.

The experience of using different business process modelling and simulation tools (ARIS, Inco, iGrafx Process) in our research practices, shows that due to the high insensitivity of communication with employees, simplicity and understandability could be assumed as one of the most important advantages of the modelling technique. In addition to its simplicity, iGrafx Process was selected as it integrates powerful and complete discrete-event simulation functions.

**Figure 1: Promotion of the employees in education to a higher professional title**



Despite the advantages of iGrafx Process, some disadvantages have to be mentioned:

- There is no business process repository and business objects repository.
- There is no interface or tool to support the transformation of business process maps to information systems modelling tools (e.g. CASE tools).
- iGrafx does not provide realistic animations as some other simulation tools do.

#### **4.1 Promotion of the Employees in Education to a Higher Professional Title**

The Administrative processes group includes some of the most frequently executed processes and are therefore very interesting for a detailed examination and analysis in the BPR and informatization project as significant improvements in efficiency can be expected.

This group consists of more than 30 processes, however some of them are of the same type, but for different areas (e.g. elementary schools, high schools, universities) and therefore their substantial activities are executed in different departments. In the first phase of the analysis we have examined some processes with the highest application frequency in more details. One of them is "Promotion of the employees in education to a higher professional title" (Figure 1) which has about 2500 applications per year. The rate of complete application is 60%, after the completion of incomplete application this rate is 80%. The owner of this process is the General Affairs and Human Resource Service (GAHRS), where the application is professionally executed by four officers. The applications are always accepted only in dispatch centre. The application state is recorded four times, always twice: manually and using a computer program. The Minister signs the decision statement.

#### **4.2 Analysis of the simulation results**

The simulation of the process that we have carried out showed that the mean execution time for one application is 49 days. The effective work time is less than one day. The rest of the time is the delay in the process (signing, transfers of documentation, waiting for the completion of the application etc.). One of the important benefits of using simulation in BPR projects is the possibility to discover which inefficiencies are worth to deal with, as for some changes a lot of effort is required. In the analysis two different waiting times have to be observed: the one that the Ministry cannot change (e.g. waiting for the complete application) as they depend on the regulations and on the customer (applicant) behaviour and the delays that can be decreased by the improved organization and process flow. The former type of waiting times represents 40% of the total waiting time or 38% of the total cycle time.

A great part of the delay is caused by the relatively high rate of incomplete applications. As shown by the simulation with a changed scenario, with no incomplete applications the average cycle time would decrease from 49 to 41 days. Therefore it would be worth to put some effort to better inform the customers about the process itself. What can be also observed is that highly educated professional spend a lot of their working time on administrative work, such as searching possible past applications etc. Some changes in the process have already been implemented, for example the Minister has authorized the head of the GAHRS to sign the decision statements.

However, the quantitative results of the simulation experiment as presented in the simulation report, regardless of how precise and detailed the simulation may be, are only one aspect of the business process analysis. Business process maps themselves can frequently show many problems that have not previously been observed.

In the modelling phase, several problems were observed. Beside the problems with data collection presented in the previous section there are also some difficulties related to the tool as not all the situations from the real world can be directly modelled. Some examples are (Tarumi et al. 2000):

- Process flow can be interrupted by other predominant processes.
- Multiple processes compete for a common resource.
- Many other kinds of exceptions can occur, such as the absence of personnel.
- Human behaviour cannot be predicted (e.g. some persons start tasks as late as possible to meet the deadline).

Due to these stated problems, one must keep in mind that the results obtained when using simulation modelling of business processes should be used cautiously, as the figures cannot be considered exact values. As such, its primary use is in analysis and in understanding the process itself, in observing the problems in process operation (e.g. bottlenecks), in evaluating and comparing alternative scenarios, in supporting decisions on process informatization, renovation, and in the introduction of organizational changes, etc. It has to be understood that modelling and simulation is a discipline used to promote a deeper and more complete understanding of how things work; it does not provide answers.

## **5. CONCLUSION**

The results of the simulation show several drawbacks of the existing process, which are the consequence of

dysfunctional organizational structure, functional instead of process orientation, unnecessary activities, nonexistent tracking of document flow etc. This can be seen from the low ratio between the effective work time and the mean execution time. The simulation shows also the bottlenecks in the process execution (e.g. Minister's signature), which can be diminished by the process redesign. Apart from the quantitative view on the process, the qualitative analysis of the process simulation also adds value to the understanding of the process and the possible improvements.

The above mentioned results of the analysis will be used for two purposes:

- The first one is to redesign the existing business processes. The important part of the project is to start thinking about the process orientation instead of functional orientation, well define processes ownerships, and that they start taking processes as their responsibility. For this purpose an efficient tool is visualization of process simulation that efficiently shows the process as a whole.
- Successful e-government implementation requires interconnected and harmonized business process renovation, adequate information technology, and e-government strategy.

The goal of our project is process renovation, examination and reengineering of current business policies, procedures and activities before the e-government implementation. We believe the new e-government paradigm can be embraced only by:

- Creating an environment of technology, enlightenment and receptivity;
- Treat this as a holistic organizational transformation, not a technical issue;
- Challenge your core assumptions and value propositions;
- Proactively establish a distinctive internet presence.

With this case it has confirmed that the analysis and carefully used simulation of business processes is useful since it provides insight view of policies, practices, procedures, organization, process flows and consequently shifts people's minds from functional to process organization.

## 6. REFERENCES

Al-Mashari M.; Z. Irani; and M. Zairi. (2001). "Business process reengineering: a survey of international experience." *Business Process Management Journal*, Vol. 7, No. 5, 437-455.

- Al-Mashari M. and M. Zairi. (1999). "BPR implementation process: an analysis of key success and failure factors." *Business Process Management Journal*, Vol. 5, No. 1, 87-112.
- Bhaskar R.; H.S. Lee; A. Levas; R. Petrakian; F. Tsai; and B. Tulskie. 1994. "Analysing and Reengineering Business Processes Using Simulation." In *Proceedings of the 1994 Winter Simulation Conference* (Lake Buena Vista, Florida, 1994). 1206-1213.
- Chen M. 1999. "BPR Methodologies: Methods and Tools". In *Business Process Engineering*, Elzinga D. J. et al. (Eds.). Kluwer Academic Publishers, Massachusetts, 187-212.
- Davenport T. H. and L. Prusak. 1998. *Working Knowledge*. Harvard Business School Press, Boston.
- Dhaliwal J. 1999. "An empirical review of the application of business process reengineering." In *Proceedings of the Fifth International Conference Integrating Technology & Human Decisions* (Athens, Greece). 1573-1575.
- Eatock J.; G.M. Giagliis; R.J. Paul; and A. Serrano. 2000. "The Implications of Information Technology Infrastructure Capabilities for Business Process Change Success." In *Systems Engineering for Business Process Change*, P. Henderson (Ed.). Springer-Verlag, London, 127-137.
- Government Centre for Informatics. 2002. *Action Plan E-government up to 2004 (AP-2004)*. Version 1.1, Government Centre for Informatics, Slovenia.
- Government Centre for Informatics. 2001. *Strategy of E-commerce in Public Administration for the Period 2001-2004 (SEP-2004)*. Version 1.0, Government Centre for Informatics, Slovenia.
- Greasley A. and S. Barlow. 1998. "Using simulation modelling for BPR: resource allocation in a police custody process." *International Journal of Operations & Production Management*, Vol. 18, No. 9/10, 978-988.
- Hammer M. and J. Champy. 1993. *Reengineering the corporation*. Harper Collins Books, New York.
- Hombres B., V. van Reijswoud. 2000. "Assessing the Quality of Business Process Modeling Techniques." In *Proceedings of the 33<sup>rd</sup> Hawaii International Conference on System Sciences* (January 4-7, 2000, Maui, Hawaii), Vol. 1.
- Irani Z.; V. Hlupic; L.P. Baldwin; and P.E.D. Love. 2000. "Re-engineering manufacturing processes through simulation modeling." *Logistics Information Management*, Vol. 13, No. 1, 7-13.
- Kettinger W. J. and V. Grover. 1995. "Toward a Theory of Business Process Change Management." *Journal of Management Information Systems*, Vol. 12, No. 1.
- Kettinger W.J.; J.T.C. Teng; S. and Guha. 1997. "Business process change: a study of methodologies, techniques, and tools." *MIS Quarterly*, 21: (1), 55-80.
- Ould M.A. 1995. *Business Processes: Modelling and Analysis for Re-engineering and Improvement*. John Wiley & Sons, New York [etc.].
- Porter M. E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, New York.
- Prasad B. 1999. "Hybrid re-engineering strategies for process improvement." *Business Process Management Journal*. Vol. 5, No. 2, 178-197.
- Tarumi H.; T. Matsuyama; and Y. Kambayashi. 2000. "Evolution of business processes and a process simulation tool". In *Proceedings of the Asia-Pacific Software Engineering Conference* (Takamatsu, Japan, December 7-10). 180-187.

# **SIMULATION AND INFORMATION SYSTEMS MODELLING: A FRAMEWORK FOR BUSINESS PROCESS CHANGE**

MOJCA INDIHAR-STEMBERGER, ALES POPOVIC  
University of Ljubljana, Faculty of Economics,  
Department of Information & Management Science  
1000 Ljubljana, Slovenia  
E-mail: mojca.stemberger@uni-lj.si, ales.popovic@uni-lj.si

VESNA BOSILJ-VUKSIC  
University of Zagreb, Faculty of Economics,  
Department of Business Computing  
10000 Zagreb, Croatia  
E-mail: vbosilj@efzg.hr

## **KEYWORDS**

simulation modelling, business process modelling, information system, business process change, ARIS, Corporate Modeler

## **ABSTRACT**

Many different methods and techniques can be used for modelling business processes in order to give an understanding of possible scenarios for improvement. The simulation modelling shows the process as a whole, drawbacks of the existing process, bottlenecks in the process execution and provides critical insight into process execution. The results of the simulation modelling represent a good foundation for a business process reengineering as a next step towards e-business introduction. The main goal of the paper is to present and discuss the level of information system modelling and simulation modelling methods and tools integration in the conditions of dynamic e-business environment. The paper also stressed the necessity for integrating simulation modelling and information system modelling. The examples of business process modelling and simulation tools are also presented.

## **INTRODUCTION**

The nineties of the last century had a focus on changing the business processes hand in hand with the introduction of new information technology. In the 90s, BPR focused on internal benefits such as cost reduction, the downsizing of a company and operational efficiency, which are more tactical than strategically focused. Nowadays, e-business renovation (BR) strategies focus on the processes between business partners and the applications supporting these processes. These strategies are designed to address different types of processes with the emphasis on different aspects (Kalakota and Robinson, 2001): customer relationship management, supply chain management, selling-chain management, and enterprise resource planning.

Many authors have shown that the awareness of IT capabilities and information systems modelling techniques influenced the design of business processes (Davenport, 1993; Gigalis, 2001; Grant, 2002; Arora and Kumar, 2000). In addition to investing in information technology, a new type of information

systems models has to be designed. The dynamic structure of information systems demands the implementation of process-oriented methods and tools. Since prior to business process change, companies need to assess the costs of business process change and to compare it with the expected benefits, simulation modelling has an important role in the projects of business process reengineering.

The main objective of this paper is to present and discuss the level of information system modelling and simulation modelling methods and tools integration in the conditions of dynamic e-business environment. The paper is structured as follows. Following a brief overview of business renovation strategies, the main characteristics of simulation modelling methods and tools are summarized. A relationship between simulation modelling and information system modelling is described. Finally, the main findings of this research are discussed and concluding remarks are provided.

## **THE OVERVIEW OF BUSINESS PROCESS CHANGE**

The emphasis on business process change has gone through a number of phases in the last 15 years. First, there was the Total Quality Management that refers to programs and initiatives to emphasize incremental improvement in work processes and outputs over an open-ended period of time (Davenport and Beers, 1995). In the early 1990s BPR has become one of the most popular topics in organizational management, creating new ways of doing business (Tumay, 1995). Since improving the business performance was not achieved by automating existing business activities, many leading organizations have conducted BPR in order to gain competitive advantage. The first wave of BPR was focused on internal business processes radical change. Furthermore, it was particularly suggested that TQM should be integrated with BPR (Al-Mashari and Zairi, 1999).

The second wave of BPR began in 1996 when the Internet and World Wide Web phenomenon took off and provided IT internetworked infrastructure that enabled electronic business and new forms of Web-based business processes (El Sawy, 2001). To meet customer demand, companies depend on close cooperation with

customers and suppliers. BPR driven by e-business could not be based only on radical redesign of intra-organisational processes, but should be extended to the entire business network (internal and external).

An online partnership must extend far beyond presenting promotional and pre-sales activities on companies' Web sites. It has to drill deep into a company's processes in order to create totally different business models. Therefore, most companies need to re-evaluate and Web-enable core processes to strengthen customer service operations, streamline supply chains and reach new customers. Traditional companies are forced to change their current business models and create new ones. The use of the Web and supply chain management has opened up the opportunities for exchanging information and managing knowledge around the new processes.

### BUSINESS PROCESS CHANGE THROUGH SIMULATION MODELLING

Business process change involves changes in people, processes and technology. As these changes happen over time, simulation appears to be a suitable process modelling method. The list of the available business process modelling tools supporting simulation is as long as over 50 names (Hommes, 2001). Simulation is often called a technique of last resort because it is used when the system to be modelled is too complex for analytical models (Oakshot, 1997). The interaction of people with processes and technology results in an infinite number of possible scenarios and outcomes that are not possible to predict and evaluate using widely popular static process modelling methods. Kettinger et al. (1997) mention simulation as one of the modelling methods in their survey on business process modelling methods.

The reasons for the introduction of simulation modelling into process modelling can be summarized as follows (Pidd, 1996):

- **Dynamic** – process behaviour varies over time.
- **Interactive** – processes consist of a number of components which interact with one another.

- **Complicated** – the process consist of many interacting and dynamic objects.

The main advantage of simulation modelling is in its' integration of following functions: **analysis and assessment** of business processes, either in quantitative or qualitative terms; **development of "to-be" models** in order to examine "what-if" scenarios and **export to implementation platforms**, such as workflow management and enterprise resource planning systems. Modern simulation software tools are able to model dynamics of the processes and show it visually, which then can enhance generating the creative ideas on how to redesign the existing business processes. Such tools include graphic user interface (GUI) that enables process animation and graphical display of simulation results.

Several authors (Denis et al, 2000; Greasley, 2000, Giaglis and Paul, 1996) have reported the application of simulation for business process redesign. Despite the numerous advantages of simulation software, it is apparent that some user requirements are still not adequately met. The survey on the use of simulation software tools conducted by Hlupic (Hlupic, 2000) revealed that the main positive features are ease of model development and visual facilities, while main problems were lack of links with other packages (software compatibility) and lack of interfaces for data input.

### BUSINESS PROCESS MODELLING METHODS AND TOOLS

Business process modelling projects can have different goals and similarly those creating the models could use different methods and tools (Table 1). Methods and tools for business process reengineering do not adhere to one particular business process modelling standard, but it must be pointed out that most modelling techniques used in business today have been developed for industrial engineering, software engineering or information systems modelling environment.

Table 1: Focus of different BPR methods/tools

Focus of BPR methods/tools	Example
Strategic planning	Balanced Scorecard - BSC, Benchmarking
Accounting techniques	Activity Based Costing Analysis – ABC, Return on Investment – ROI
Continuous improvement	Total Quality Management – TQM, ISO Standard
Static process modelling or functional decomposition modelling	Data Flow Diagrams - DFD, IDEF0
Action coordination modelling	Action Workflow modelling method
Dynamic process modelling (simulation)	Petri Nets

Over the last three decades, a well-established procedure for modelling information systems was based on two complementary aspects of analysis: data modelling (entity-relationship modelling) and function modelling (data-flow diagramming). Since events which trigger a response in an information system come from within the organisation or from the external environment, it is obvious that a third representational framework is effectively a business process view (Scheer, 1994).

Giaglis (Gigalis, 2001) developed a Taxonomy of Business Process Modelling techniques where the modelling techniques are classified by the purpose that they would have when used in business process modelling projects. According to this taxonomy, modelling techniques could have informational (data), organisational (where, who), behavioural (when, how) and functional (what) focus, and can be used to fulfil different objectives: understanding & communicating, process improvement, process management, process development and process execution. It is obvious that there does not exist a single process modelling technique that covers all aspects of process modelling, specially the aspect of process dynamics.

Except the simulation that has been discussed in Section 3, there have been three generic approaches to solving the problem of system dynamics (El Sawy, 2001). One approach is to extend functional decomposition methods with event triggers in order to introduce task interdependence into the model. The example of this approach is the ARIS methodology.

The second approach is to extend action coordination methods with added workflow structure through Petri net activity representation, like it was done in the Role Activity Diagramming method or UML Activity Diagram.

The third approach is to develop new process modelling methods that are focused on process flow and process dynamics, such as IDEF3 and Activity Decision Flow diagrams.

#### **A FRAMEWORK FOR BUSINESS PROCESS CHANGE**

Process modelling is one of the most cost-effective and rewarding ideas to come along in years. On the other hand, the successful development of information systems requires an integrated approach, which includes modelling of business processes, as well as, information systems modelling and development. Therefore a rapid growing number of frameworks and modelling tools have been developed for an integrated modelling of the entire enterprise with the focus to both organisational modelling and information systems modelling (Hommes, van Reijswoud, 2000).

#### **Simulation Modelling and Information Systems Modelling: The Need for Integration**

Nowadays, the ability to develop and deploy simulation models quickly and effectively is far more important than ever before. As process modelling is very much a business rather than technical role, a modelling tool must be simple to use by a non-technical business user. However, a number of factors such as inefficient data collection, lengthy model documentation and poorly planned experimentation prevent frequent deployment of simulation models (Perera and Liyanage, 2001). In the majority of cases, the analysis of business process models is based on hand entered parameters such as time required to execute a given function, waiting time, availability and utilization of resources, etc. In cases where the business processes are supported by information systems, there is a transaction base which contains data on the processes, and it is necessary to develop an interface for the business process database, and to develop components with the task of exporting data from the production databases of a given information system and importing that data into the analytical bases, that is, to give parameters to the business process database.

The need for integration of simulation modelling and information systems modelling methods is evident in many cases. A flexible data collection link to a company's enterprise resource planning (ERP) database will undoubtedly improve the efficiency of model maintenance. Therefore, the methodology for rapid identification and collection of data structure for simulation modelling is developed by Perera and Liyanage (2001). It provides the link between the data conveniently stored in a database and the simulation model. This approach supports also the need for detailed model documentation via the use of standard modules from the functional model (IDEF0) library. Moreover, recent advances in simulation software (integration via VBA) afford the automatic creation of the entire simulation model.

Despite attempts to become user-friendly, dynamic discrete event modelling lends itself most readily to specific, single dimensional problems. Since the business practice has shown that there was no ideal simulation or business process modelling technique, the interfaces for automatic translation and integration of different techniques were developed. The examples are the software tools used for translation of IDEF diagrams into Petri nets: Design/IDEF, Design/CPN, WorkFlow Analyser, Service Model and WITNESS (Pinci and Shapiro, 1991; Shapiro, 1994). IDEF3 based descriptions were used to automatically generate WITNESS simulation code in the target language using ProSim (Painter et al, 1996).

Several frameworks have been developed which attempt to provide an open modelling architecture for general

models, but most of them deal effectively with non-dynamic modelling issues whilst dynamic modelling issues have traditionally only been addressed at the operational level. These include IDEF, CAM-I, GIM, ARIS, IEM, the ISO Reference Model, CIMOSA and GERAM (Vernadat, 1996; O’Sullivan, 1994). Therefore, the efforts are focused to apply simulation modelling in the enterprise modelling frameworks (Dewhurst et al, 2002).

The developers of the Unified Modelling Language (UML) have recognized the need for modelling methods which allow process modelling. Therefore diagrams like the use case diagram and the activity diagram have found their way into the UML (Lieberman, 2001). Activity diagrams combine the various approaches of different technique such as event diagrams of Jim Odell, state diagrams and Petri nets. The Event-Driven Process

Chain (EPC) method was developed at the Institute for Information Systems (IW<sub>i</sub>) of the University of Saarland, Germany, in collaboration with SAP AG (Loos and Allweyer, 1998). As the key component of SAP R/3’s modelling concepts for business engineering and customizing, it is based on the concepts of stochastic networks and Petri nets.

The above examples from business practice have shown the existence of large market space to improve BPM tools with the components for dynamic modelling and measuring the performance of the processes, and to integrate it with tools for developing information systems, which substantially decrease the time required to create the company’s information system. According to the trends recognized from current business practice and literature, the typical features of integrated BPM tool could be summarized as follows (Table 2):

Table 2: Features of integrated BPM tool

Feature	Description
Data Modelling	Providing the function of entity modelling, used to create logical data model to support business processes
Static Process Modelling	Used to build a “top-down” understanding of processes and to analyse an enterprise process model static analysis (i.e. direct calculation of critical measures – number of resources required, total process time, cost being incurred)
Dynamic Process Modelling	Used to design and communicate end-to-end business processes (a static process model can be modelling transferred easily into its corresponding dynamic model by entering time-related data)
Data and process modelling interface	Mapping business processes to logical data, describing relationships between processes, applications and organizations
Repository	Used to manage objects and models, enables multi-user working and sharing of object between different views
Publisher	Automatically documenting and publishing process and system changes in order to train the staff and to enable communicating the new business practice

### The Examples of Integrated BPM Tools

ARIS (IDS Scheer) and CorporateModeler (Casewise) software tools are used in this Section to explain the basic ideas underlying business process modelling and simulation modelling. These tools are selected on the base of authors’ participation in Croatian and Slovenian BPR projects and the great number of large companies using these tools. While ARIS is used to present the example of different BPR methods/tools integrated in the system (as explained in the Table 1), the description of CorporateModeler presents the key building blocks of an integrated BPM tool (as stated in the Table 2).

The **ARIS Toolset** (Architecture of Integrated Information System) version 6.1 of IDS Scheer stands for a group of systems, the essential feature of which consists in the functions of documenting, analyzing, changing, implementing and optimizing business processes. ARIS integrates business processes database

and disposes of a browser enabled Front-End. This means platform independence for users, worldwide availability, high scalability and low administration costs (Scheer, 2002; IDS Scheer, 2000). Knowledge about company processes is stored in the ARIS database objects. Using the ARIS Toolset the enterprise business processes are analyzed and described. Each object is defined through different perspectives: **organization, function, data and process view** and attributes which could be used as the input parameters for **ARIS Simulation, ARIS ABC** (Activity Based Costing), and **ARIS BSC** (Balanced Scorecard) tool. Since ARIS Simulation is fully integrated in the ARIS Toolset, the data relating to the processes, recorded in the ARIS Toolset could be used as a basis for the simulation of business processes. This simulation supplies information about the executability of processes, process weak points and resource bottlenecks. There is also the **interface toward Workflow management tools, CASE tools** (ORACLE Designer 6i) and **project**

**management tools. ARIS Process Performance Manager** (ARIS PPM) automatically identifies performance data from company processes, especially those which span systems, and thus makes it possible to analyze them. This information can be gathered from software systems, for example, for ERP, SCM, CRM, e-Business, or workflow management.

Another BPM tool to be presented in this paper is **Corporate Modeler 8e**. It supports six core diagram types. **Hierarchy Modeler** provides an overall picture of the business. Starting at the highest level, users can drill-down into the lowest level of detail for all object types. **Process Dynamics Modeler and Simulator** uses dynamics modelling to model activities and their dependencies within the end-to-end business process. It shows business events that trigger the process, the process flow, roles, and responsibilities mapped as swim lanes, which illustrate which department is responsible for each process step. Process models can be simulated to produce statistical analysis of resource utilisation, throughput times, costs, and overall performance. **Generic Modeler** allows the creation of user-defined notation style and symbology, which enables users to create their own diagram templates to model application architecture, EPC (SAP) Diagrams, and Use Case Diagrams. Work level procedures, such as flowcharts and Activity-Based Modelling (ABM) diagrams can also be modelled. **Data Flow Modeler** depicts the information flow between processes, external entities, and data stores. **Entity Modeler**: Enables users to design the data structure by defining tables, fields, and their properties. This is used to create an entity relationship diagram. **Matrix Manager**: Defines the relationships between processes, entities, locations, application technologies, organisations, and so on. The integration of the models to external applications is provided, including Sybase's PowerDesigner, Rational Rose, Staffware, Oracle Designer, ERwin, Telelogic, and Visio. Corporate Modeler 8e provides tight integration allowing processes to be transferred to a workflow application. It has an XML import/export capability that follows the standards established by the Workflow Management Coalition (WfMC).

### Integrated BPM Tools in Practice

There are about 40 Croatian and Slovenian large companies using ARIS in business process modelling projects, most of them from banking, financial, telecommunication and government sector. Similarly, Corporate Modeler is used in the Croatian Ministries (i.e. Ministry of Transport) and the telecommunication company. Both of the tools are used by organisations across the world to understand complex operations and optimize performance through improvement projects. Only two examples are briefly presented here.

Efficient usage of ARIS in Slovenian insurance company **Slovenica** is reported in Divic-Mihaljevic (2002) where simulation modeling was used in order to implement integrated information system. The aim was not only to graphically present business processes, but also to support them thoroughly with an appropriate IT solution. The project started in 1999, when Slovenica's gross written premium climbed for 22.4% and Board of Management decided to build an integrated information system to support business processes. Besides that the main goals were improved market position and acquisition of a larger market share, improved portfolio quality and improved financial strength and increased profit. They did not only perform the optimization of processes but prepared them for IT implementation.

In the case study of Croatian insurance company (Ivandic-Vidovic and Bosilj-Vuksic, 2003) ARIS Toolset was used to establish a single repository of business processes. A model of the company organizational structure was created, as were models of the business processes of the company at 5 levels. A comprehensive database of the company's business processes was created and used in the project of information system development and redesign. Following this, a system of managing business processes was implemented, namely, metrics and analysis were conducted on those processes. The parameterized model was transformed into simulation model, which was used by the process owners and managers to find the best solution for business process change.

### Discussion

According to the overview of the actual state of the art in the "business process modelling" field and the experiences from Croatian and Slovenian companies the advantages and disadvantages of integrated business process modelling tools could be summarized and discussed. The aim of using business modelling is to develop a **framework** that:

- interrelates several business process modeling methods and techniques,
- is easy to design and understand,
- encourages standardization,
- provides a single business process repository and the use of a common process vocabulary,
- is able to tune and optimize the processes of a company
- provides model analysis, validation and testing
- is formal enough to serve for software development purposes.

Most of integrated BPM tools meet these requirements and therefore are used in BPR projects, but the authors have also observed some difficulties in using these tools.

Very serious problem is **the inability to translate business models** into information (workflow) models. Except the potential benefits from process improvements, and maximization of process performance via implementing process change, the key goals of companies' projects are to model enterprise applications, integrate and interconnect different applications, providing an effective business environment that meets customers' performance demands. While business models should act as a basis for creating suitable information systems and defining engineering requirements, BPM tools should enable the export of business process models to implementation platforms, such as workflow management and enterprise resource planning systems. To support the transition between the business process modelling and the information system (IS) modelling, the direct mapping and transition of all entities and activities defined during business process modelling should be enabled. Although the software interfaces between process modelling and IS modelling are developed, these interfaces might provide some syntactical translation but they cannot bridge the semantic gap between business processes and IS models. Here the manual revision of IS models is often more efficient and useful than the use of interfaces, but the problem is expected to be solved by the producers of BPM tools using the appropriate rule-transformation approach and introducing the rule repository.

Rule repository is the core of a development environment providing appropriate tools for process, workflow, data and organisation modelling, process refinement, as well as import and export capabilities. It can also be regarded as an integration link between business modelling and IS modelling. The motivation to develop a rule repository is to establish an environment in which business rules can be traced from their origin in the business environment through to their implementation in information systems. This provides the information necessary for rapid information system maintenance and adaptations to changes in the business environment. Its purpose is to describe the activities that must be undertaken to achieve an explicit goal and establish a clear link between business process modelling and IS modelling.

Another problem noticed in Croatian and Slovenian companies was **the risk of over-analyzing** existing business processes which led to the long period of modelling (1-2 years), producing a huge documentation on "as-is" business processes and getting stuck in the business process analysis phase of the project (e.g., analysis paralysis) from which they were never able to move on. Therefore, the volume of business process models (i.e. number of models, number of diagrams and their levels) must be defined and strictly limited to the scope of the project.

## CONCLUSIONS

To realize the business process change, most of companies use different methods and tools, which integrate components for static and dynamic modelling and measuring the performance of the processes. Simulation modelling is used to benchmark the current, "As-Is" process, to verify model set-up and metrics and to test 'to-be' scenarios when re-designing business processes or supply chains. Changes can be easily and inexpensively examined and graphical presentations can be used to gain organizational commitment to change.

The integrated BPM tools combine formerly diverse areas of business process, IT, resource and financial modelling, enabling the companies to form a complete view of their operations and providing a framework for efficient development of a robust and complete enterprise architecture. Furthermore, the numerous interfaces are developed to enable the connection with tools for developing information systems, which substantially decrease the time required to create the company's information system and to permit fast and simple tracking of operations.

These possibilities are shown in this research with the example of the ARIS and the Corporate Modeler toolset. Since nowadays, the majority of Croatian and Slovenian companies are involved in the projects of business process change and ERP systems development, this research could serve to adopt a process centric approach introducing business process modelling standards and rules and developing information systems modelling standards based on integration with dynamic business process modelling tools and techniques.

## REFERENCES

- Al-Mashari, M., and M. Zairi. 1999. "BPR implementation process: an analysis of key success and failure factors", *Business Process Management Journal*, Vol. 5, No. 1, 87–112.
- Arora, S. and S. Kumar. 2000. "Reengineering: A Focus on Enterprise Integration", *Interfaces*, Vol. 30, No. 5 (September-October), 54-71.
- Casewise (2003), "Corporate Modeler 8e – from vision to process". <http://www.casewise.com/>
- Davenport, T. H. 1993. *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, Boston.
- Davenport, T. H. and M.C. Beers. 1995. "Managing Information about Processes", *Journal of Management Information Systems*, Vol. 12, No. 1, 57-81.
- Dennis, S.; B. King; M. Hind and S. Robinson. 2000. "Applications of business process Simulation and lean techniques in British Telecommunications PLC". In *Proceedings of the 2000 Winter Simulation Conference*, 2015-2021.
- Dewhurst, F.W.; K.D. Barber and M.C. Pritchard. 2002. "In a search of a general enterprise model", *Management Decision*, Vol.40, No. 5, 418-427.
- Divic-Mihaljevic A. 2002. "Process Design and Implementation with ARIS". In *Business Process*

- Excellence, ARIS in Practice*, A.W. Scheer et al. (Eds.), Springer-Verlag, Berlin, 149-174.
- El Sawy, O. 2001. *Redesigning enterprise processes for e-Business*, McGraw-Hill, New York.
- Giaglis, G. M. and R.J. Paul. 1996. «It's Time to Engineer Re-engineering: Investigating the Potential of Simulation Modelling in Business Process Redesign». In *Business Process Modelling*, B. Scholz-Reiter and E. Stickel (Eds.), Springer-Verlag, Berlin, 313-332.
- Giaglis, G.M. 2001. "A taxonomy of business process modeling and information systems modeling techniques". *International Journal of Flexible Manufacturing Systems*, Vol. 13, No. 2, 209-228.
- Grant, D. 2002. "A Wider View of Business Process Reengineering", *Communications of the ACM*, Vol. 45, No. 2 (Feb), 85-90.
- Greasley, A. 2000. "Effective uses of business Process Simulation". In *Proceedings of the 2000 Winter Simulation Conference*, 2004-2009.
- Hlupic, V. 2000. "Simulation software: An operational research society survey of academic and industrial users". In *Proceedings of the 2000 Winter Simulation Conference*, 1676-1683.
- Hommel, B. 2001. "Overview of Business Process Modelling Tools", <http://is.twi.tudelft.nl/~hommel/scr3tool.html>
- Hommel, B. and V. Van Reijswoud. 2000. "Assessing the Quality of Business Process Modeling Techniques". In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Vol. 1 (Maui, Hawaii, January 4-7), IEEE, Piscataway, N.J., 1-10.
- IDS Scheer. 2000, "ARIS Methods Manual; Version 5", Saarbrücken.
- Ivancic-Vidovic, D. and Bosij-Vuksic, V. (2003), "Dynamic business process modelling using ARIS", in *Proceedings of 25<sup>th</sup> Information Technologies Conference – ITI'2003*, Cavtat, Croatia, 607-612.
- Kalakota R. and M. Robinson. 2002. *E-Business 2.0: Roadmap for Success*, Addison-Wesley, Boston.
- Kettinger W.J.; J.T.C. Teng and S. Guha. 1997. "Business process change: a study of methodologies, techniques, and tools", *MISQ Quarterly* (March), 55-80.
- Lieberman, B. 2001. "Using UML Activity Diagrams for the Process View". <http://www.therationaledge.com/>
- Loos, P. and T. Allweyer. 1998. "Process Orientation and Object-Orientation – An Approach for Integrating UML and Event-Driven Process Chains (EPC)". Paper 144, Publication of the Institut für Wirtschaftsinformatik, University of Saarland, Saarbrücken, <http://www.iwi.uni-sb.de>
- O'Sullivan, D. 1994. *Manufacturing Systems Redesign*, Prentice-Hall, London.
- Oakshot, L. 1997. *Business Modelling and Simulation*, Pitman Publishing, London.
- Painter, M.K.; R. Fernandes; N. Padmanaban and R.J. Mayer. 1996. "A Methodology for Integrating Business Process and Information Infrastructure Models". In *Proceedings of the 1996 Winter Simulation Conference*, 1305-1312.
- Perera, T. and K. Liyanage. 2001. "IDEF based methodology for rapid data collection", *Integrated Manufacturing Systems*, Vol.12, No. 3, 187-194.
- Pidd, M. 1996. *Computer Simulation in Management Science*, John Wiley & Sons, Chichester.
- Pinci, O. and R.M. Shapiro. 1991. "An Integrated Software Development Methodology Based on Hierarchical Colored Petri Net". In *Lecture Notes in Computer Science, Vol. 524; Advances in Petri Nets 1991*, G. Rozenberg (Ed.), Springer Verlag, Berlin, 227-252.
- Ritchie-Dunham, J.; D.J. Morrice; J. Scott and E.G. Anderson. 2000. "A strategic supply chain simulation model". In *Proceedings of the 2000 Winter Simulation Conference*, 1260-1264.
- Scheer, A.W. 1994. *Business Process Engineering, Reference Models for Industrial Enterprises*. Springer-Verlag, Berlin.
- Scheer, A.W. 2002. *Business Process Excellence, ARIS in Practice*. Springer-Verlag, Berlin Heidelberg.
- Shapiro, R.M. 1994. "Integrating BPR with Image-Based Work-Flow". In *Proceedings of the 1994 Winter Simulation Conference* (Lake Buena Vista, Florida), 1221-1227.
- Tumay, K. 1995. "Business process simulation", In *Proceedings of the 1995 Winter Simulation Conference* (Washington DC), 55-60.
- Vernadat, F.B. 1996. *Enterprise Modelling and Integration*, Chapman & Hall, London.

## AUTHOR BIOGRAPHIES



**Mojca Indihar-Stemberger** received her Master in Computer and Information Science degree in 1996, and her Ph.D. in Information Science in 2000 from the University of Ljubljana, Slovenia. Currently she is an assistant professor at the Faculty of Economics, University of Ljubljana. Her research interests include business process reengineering, business renovation, e-business, decision support systems and business modelling. She is a president of Organising Committee at the Slovenian Informatics conference.

# UNDERSTANDING THE DYNAMIC INTERACTIONS BETWEEN BP AND IT USING SIMULATION

Alan Serrano  
Department of Information Systems and Computing  
Brunel University  
Uxbridge Middlesex  
UB8 3PH, London, UK  
E-mail: [Alan.Edwin.Serrano-Rico@brunel.ac.uk](mailto:Alan.Edwin.Serrano-Rico@brunel.ac.uk)

## KEYWORDS

Business Process, Information Systems, Discrete-event Simulation.

## ABSTRACT

Business Process (BP) design approaches claim that Information Technology (IT) is a major enabler of business process, a view also shared by the Information Systems (IS) community. Despite this fact, approaches in these domains do not provide clear indication of which modelling techniques could be used to detect IT opportunities within a business process context. This paper examines the advantages and limitations of a simulation framework used in a research project, namely ASSESS-IT, that aimed to depict relationships between BP and IT. It provides evidence that for some cases the relationships between BP and IT can be viewed by focusing in the relationship between BP and IS alone and assess the feasibility of using an alternative framework, namely BPISS, to address this relationship. Finally, this paper provides evidence that the BPISS framework could help BP and IS analyst to assess the benefits, or constraints, that the functionality of a given IS design may bring to the BP.

## INTRODUCTION

Business processes became the focus of continuous improvement efforts in the mid-40's (Davenport and Stoddard, 1994). It is argued, however, that process analysis started far before in 1911, when Frederick Taylor first advocated the systematic study of work procedures. From that time, the concept of process became very important. For example, process control and process techniques have been outlined in the quality movement (Juran, 1964; Garvin, 1988). Process skills and process consultancy have been very important in human relations and management of change schools (Schein, 1969). Operations management is concerned with the management of processes, people, technology, and other resources in the production of goods and services (Armistead et al., 1995).

It was at the beginning of the 1990's when the process movement became stronger. Business Process Reengineering also named Process Redesign, or Process Innovation is today one of the most popular concepts in business management (Davenport, 1993; Hammer and Champy, 1993). The study of business processes, however, is not isolated and has always been related to Information Technology. IT is considered one of the most important enablers of process change. For example, in one of the first articles about BPR, Davenport and Short (1990) argue that together, processes and information technology can be seen as a new industrial engineering that may revolutionise the way in which organisations operate. Similarly, Hammer and Champy (1993) claim, in one of the most renowned books on BPR, that IT is part of any reengineering effort, and they position IT as "an essential enabler".

Most of the advocates of the business process reengineering movement highlight the importance of the role that IT plays in the reengineering process. Many argue that IT should be seen as an enabler of organisational change rather than as a tool to implement business processes (Davenport, 1993). Childe et al. (1994), for example, state that the initiative to move towards BPR in many cases originates from the IT departments. In one of the first empirical studies on IT-enabled BPR, Grover et al. (1994) claim that the success of IT to enable BPR lies in IS-strategy integration. They contend that the success of IT-enabled BPR efforts will succeed only if they are directed through a strong integration with strategy. This relationship, however, is not fully explored in most of the existing business process methodologies nor in the IT domain. Trying to address this problem the ASSESS-IT project propose a simulation framework that aim to depict the interaction between BP and IT. The following section describes the ASSESS-IT framework, it analyses its advantages and limitations and provides the basis to propose an alternative framework to address the limitations found.

## THE ASSESS-IT PROJECT

The ASSESS-IT project assumed that the relationship between BP and IT could be seen as a three layered structure, namely Business Process, Information Systems and Computer Networks (CN). Business processes usually rely on the support provided by the information systems to perform many of the activities. Similarly, the information systems that support these processes also depend on the underlying communications infrastructure, namely computer network (see Figure 1).

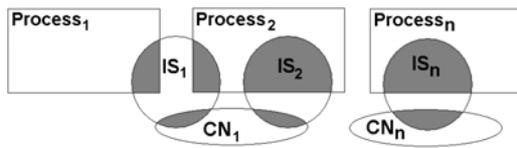


Figure 1 Business Process/IT Relationship

Figure 1 depicts the relationship between processes and IT domains. The rectangles represent the various processes that may be found in any organisation, the circles represent the information systems that support those processes and the ovals represent the computer network infrastructure.

Figure 1 suggests that the relationships between processes, information systems and computer networks are complex and that changes in any domain may have an impact on the others. Figure 1 also suggests that particular attention must be paid to the IS domain since it is the "connection" layer between the business processes and computer network layers. Because of this, changes to the business process may have a direct impact on the information systems, affecting indirectly the network infrastructure. Likewise, changes to the computer network infrastructure may have a direct impact on the information systems, affecting indirectly the process layer. Finally, changes to the information systems may have an impact on both the process and network layers. Thus, processes, information systems, and computer networks may be viewed in terms of two relationships, namely BP-IS and IS-CN, which are related by the information system layer. Recognising the complexity of these relationships creates difficulties in assessing the impact that changes to any of these domains may have to the others before implementing them.

ASSESS-IT aimed to provide practical solutions to support the process of change in organisations by providing on the expected impact of IT investments on business performance. The ASSESS-IT project proposed the use of an alternative simulation framework and tests it using a case study. The following is a résumé of the case study together with the simulation framework proposed in this project.

## The Case Study

The case study presented in ASSESS-IT consisted of two collaborating organisations in Greece. One company is a branch of a major multinational pharmaceuticals organisation (we will refer to this company as Org-A), while the other is a small-sized regional distributor of Org-A's products (we will refer to this company as 'Org-B').

The case study was carried out within a single business unit, which deals with hospital consumables. The business unit imports products from other Org-A production sites across Europe. The goods are stored in a warehouse that operates as a central despatch point for all products, which are then distributed to the company's customers via a collaborating distributor, namely Org-B. Org-B responsibilities include:

- Maintaining an adequate inventory of products to fulfil the orders.
- Distributing the ordered products to customer premises.

Org-B has to operate within rigorous deadlines. The agreement between the companies, stipulates that each order has to be fulfilled within 24 hours for products delivered within the city of Thessaloniki, or within 48 hours for the rest of northern Greece. Org-A management noted, however, that these targets are rarely met in practice. A brief analysis by the companies seemed to attribute the problems to some inefficiencies within the ordering system as well as difficulties being experienced by Org-B in maintaining their inventory at an optimal level. The effects that these inefficiencies caused were seen as a major source of customer dissatisfaction, so an in-depth analysis of the problem was commissioned. The main objectives of this study were:

- To examine the existing business processes that were felt to be responsible for long lead times for order fulfilment.
- To determine the sources of problems and propose alternative solutions.
- To evaluate the potential of introducing appropriate IT to improve communication between the two companies.

## The ASSESS-IT Framework

The basic idea behind the ASSESS-IT framework is simple. Changes to business processes usually involve changes to the information systems. Similarly, modifications to the information system architecture or

the insertion of a completely new software application increases or decreases the network traffic load. Consequently, depending on the network infrastructure, changes at the network traffic level may affect the communications between the network components along the network (nodes, servers, routers, communication lines etc). This affects the performance of the software applications that run over the network, including the information systems that support the business process under analysis, which in turn, may have unexpected consequences in the business process performance. The ASSESS-IT project investigated the suitability of using discrete event simulation models to assess the impact of the insertion of an IS in Org-A and Org-B. To this end, two discrete event simulation techniques were selected to model both business process and IT: Business Process Simulation (BPS) and Computer Network Simulation (CNS).

In order to calculate the business effects of changing the underlying IT, the ASSESS-IT framework developed a computer network model, including IS design, it then identifies the information that may be relevant to the BPS model and incorporates it in the latter (see Figure 2). That is, if a computer network model of the proposed IT system is built, the outputs from this model can be directly fed into the business process model, thus, reflecting the changes that the new IT would produce at the BP level. The ASSESS-IT framework is based on the steps for a simulation study suggested in (Banks et al., 2000). The steps proposed in the ASSESS-IT framework are depicted in Figure 2 and summarised next:

1. The problem formulation and setting of objectives and overall project plan should be performed together for both business process and computer network models.
2. Model conceptualisation and data collection steps should be performed separately for both BP and CN models.
3. A new step, BP/IT model conceptualisation is introduced. The aim of this step is to co-ordinate the conceptualisation of the BP with the CN models and vice versa so they reflect both process and information technology.
4. Before undertaking the experimental design step, the BP modeller should wait for the input from the CN model results (e.g. transmission times over specific network conditions). This information needs to be considered for the experimentation design phase in the business process model.

A complete description of the ASSESS-IT framework and analysis of the results can be found in (Eatock et al., 2001).

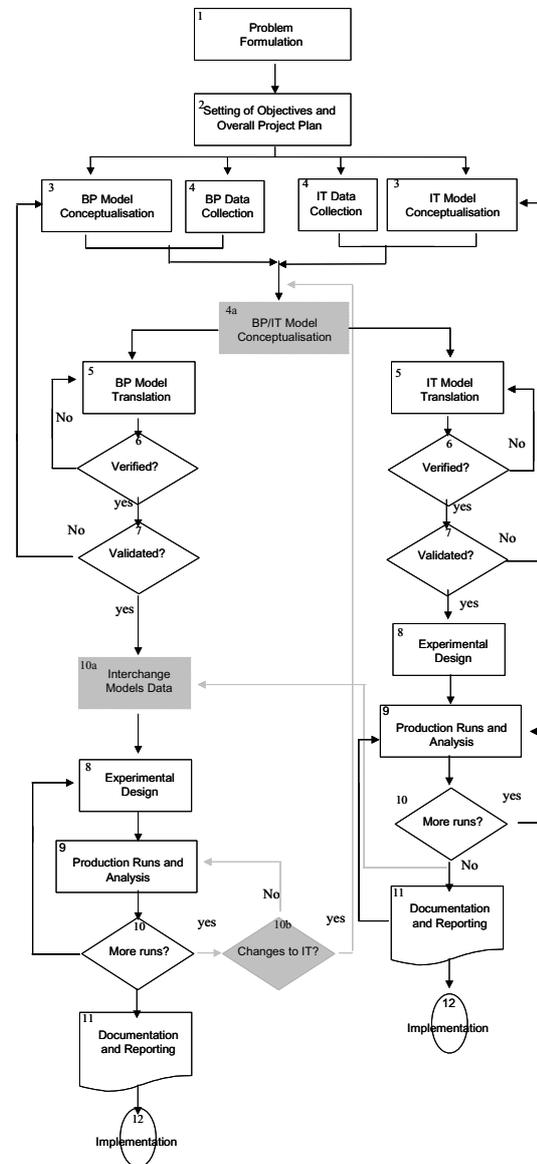


Figure 2 ASSESS-IT Framework

### Experimentation and Results

The ASSESS-IT framework is divided into two major phases. The first phase develops the Computer Network Simulation model. The second phase develops the Business Process Simulation Model and uses the outputs of the CNS model to reflect the impact of the IT in the BP. This section presents a resume of the results obtained in each phase.

### Phase One. The CNS Model

Once the computer network model was designed and verified a number of pilot runs were performed. The results showed that the time taken to complete an order was a maximum of 3.6 seconds and an average of 2.3 seconds. The fact that the figure was significantly low led to rethink the way the model was designed. The model represented the traffic flow generated by a single IS which in turn produces on average an order every 24 minutes. Thus, the traffic flow was considerably low. It was thought that in reality the traffic in a computer network would be composed of a series of applications. The fact that a number of IS applications run over a computer network will increase the network traffic which in turn may have a greater effect on the transmission times in the IS. Therefore, it was decided to experiment with a series of different network traffic utilisation levels. The computer network model was run using varying degrees of network utilisation, which correspond to different levels of underlying IT capability to support the business process workload imposed on the network. The network response times were recorded for each run.

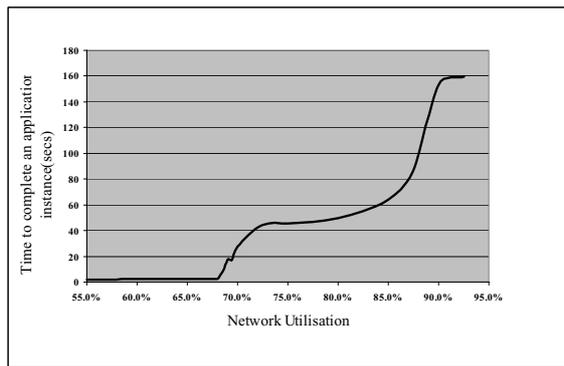


Figure 3 Application Instance Completion Time with Varying Degrees of Network Utilisation

From Figure 3 it can be seen that until network utilisation reaches a critical point (around 67%) application instance duration are relatively steady and utilise the network for about 2 seconds per application instance (number of transactions produced by a given node within the network); a response time that can be considered as acceptable from the end-user viewpoint. As network utilisation increases, however, it can be observed a considerable increase in application instance completion times, rising to almost 45 seconds for high network workloads (67 to 79%), to 60 seconds for very high network workloads (79% to 85%), followed by an extremely sharp rise as utilisation rises to 92% (indicating network congestion at such high utilisation).

### Phase Two. The BPS Model

Three scenarios that would possibly alleviate the problems with backorders were considered. These are described in the following list.

1. Faxing backorders. It was proposed to fax the backorders to Org-A, instead of sending them by post.
2. Org-B sends the backorder list twice a week. Instead of waiting until Friday to send the backorder list, this is sent Tuesday and Friday evenings.
3. A final scenario that included the use of an IS was designed. The two companies share the same database to allow Org-A to have up-to-date information on stock levels at Org-B, and hence adjust replenishment shipments accordingly. The backorder list is generated automatically so Org-A knows the real-time stock levels in Org-B warehouse.

The results obtained from running scenario 1 showed some interesting results. Although the time taken to send the back orders by mail was reduced from 2 days to a matter of minutes, the reduction in time taken to deliver the entire order was reduced by only 1 hour for orders and remained almost the same for backorders. After a more thorough analysis, it was found that this situation was due to two major reasons. Firstly, due to the system's business rules, orders were retained until Friday afternoon to be faxed. Despite the fact orders arrived in a matter of minutes to their destination, employees take about 8 hours to process an order list. Therefore, orders sent by fax on Friday afternoon did not have enough time to be processed on the same day and they have to wait until Monday. In the original scenario, orders took two days to arrive at its destination by normal mail. Considering that the mail works on Saturdays, the backorder list also arrived on Monday. Hence, both scenarios process the list almost at the same time. The second, and most important reason, is that there is only one employee working in the warehouse of Org-B, who, is busy nearly 97% of his/her working time. Consequently, an extra experiment was performed adding one more warehouse employee, resulting in an 11 hours decrease for orders. Backorders, however, remained the same.

The second scenario was to schedule the replenishment shipments to be sent twice a week instead of once. This resulted in a reduction in delivery times for backorders, but it was much smaller than anticipated (11 hours for back orders). This was due to the same problem identified in scenario 1 related to the warehouse employee. When the time was measured combining the scenarios of having two employees, faxing backorders twice a week, ordering

times were showed an 11 hours reduction and backorders a 40 hours reduction.

Scenario 3 addressed the only real IT-based solution, in which, both companies share a database. Following the ASSESS-IT framework guidelines, transmission times (to send an electronic order) had to be obtained from the computer network model. Subsequently, transmission times were used in the business process model. The network reported that to send an electronic order, or backorder, could be done in less than 30 seconds.

Sharing a database gives Org-A a better idea of the replenishment requirements of Org-B. The results did not show a noticeable reduction in the delivery times for the orders that had in-stock products, on the contrary, an increase of 29 hours was noticed for orders. The problem, as in the previous scenarios, was due to warehouse employee workload. It was reported that he/she was busy 99% of his/her working time. The increase of utilisation (more than 2%) was due to he/she had to deal with a slightly higher number of backorders. Those products that required back orders, however, showed a substantial reduction of nearly 74 hours. This was mainly because the backorder list would no longer need to be created, as it would be generated in conjunction with the normal replenishment shipment. A final experiment was created which combined the results from scenario 3 with an extra warehouse employee. The results were as expected, since there was a reduction of 10 hours for orders and 82 for backorders. The times reached in this scenario were the best in comparison to the as-is scenario, though, they were still distant from Org-A and Org-B targets.

### **ASSESS-IT Limitations**

The results from the computer network-modelling phase showed that the impact that the computer network infrastructure may have on the information systems strongly depends on time. These results also demonstrated that due to current network technologies, the information systems that could suffer from changes to network architecture are those that depend on time. The type of systems that the ASSESS-IT approach aims to address, however, do not fit within this category. The experiments showed that changes to the network infrastructure and to network traffic did not have a considerable effect on IS performance, and consequently, did not affect the BP performance. Similar results were found in the business process-modelling phase. The experiments did not show a significant improvement on business process performance despite the fact that the time for those activities that were aided by the IS was dramatically reduced. A deeper analysis of this situation suggested that the problem was due to the fact the ASSESS-IT approach concentrated on

depicting the way IT affects processing time, but not in the way IT affects process performance.

Business process modelling experiments showed that time was not a parameter that could affect process performance. The experiments showed, however, that there are other IS parameters that affect BP performance. For example, once it was detected that the backordering process was a major system bottleneck, it was proposed to use of an IS to alleviate this problem. Therefore, the IS was designed, amongst other things, to reduce the number of backorders. This information, though, could not be reflected in the business process model because the BP model was designed to represent the percentage of backorders statically (as a fixed number) and not dynamically. Reducing the backorders percentage manually (from 30% to 5%) demonstrated that the reduction of backorders would reduce the overall processing time. This figure was directly related to the way the IS would handle the backordering process.

Two conclusions can be obtained from the ASSESS-IT approach exercise.

1. The computer network infrastructure does not affect the performance of information systems used to support organisational processes. Consequently, the overall business processes performance is not affected, in a significant way, by changes to the CN infrastructure. Therefore, the use of a computer network model is, in the context of the ASSESS-IT approach, unnecessary.
2. The experimentation with different BP scenarios provided evidence that suggests that in order to portray the benefits that the use of an IS may bring to the business processes, it is necessary to obtain measurements of the way the IS behaves over time.

It can be derived from these conclusions that in order to provide a modelling approach that depicts the relationships and interactions between BP and IT, it is necessary to focus on the relationship between BP and IS alone. Furthermore, the insights gained from the experiments with different BP scenarios imply that the parameters that govern the relationship between business processes and information technology are not those that are related to time constraints but are instead those that are related to IS performance measurements. It was observed that time reduction on certain activities of the ordering process did not improve business process performance. On the other hand, IS performance measurements, such as the reduction of the number of backorders produced by the IS, improved the overall process performance.

These facts lead us to think that a new BP/IT integrated approach is needed. According to the results and conclusions presented here, the approach should focus on the relationship between BP and IS, and more importantly, should depict IS behaviour measurements. This means, it is necessary to investigate more about how to model IS performance. IS performance measurements are also known as information system's non-functional requirements. Most IS modelling techniques, however, aim to depict functional requirements. Non-Functional Requirements (NFR), on the other hand, are not easy to represent in a measurable way, thus, a limited number of techniques and approaches can be found.

The results rendered by the ASSESS-IT approach highlighted the importance to portray the dynamic behaviour of the IS as it evolves over time. The problem, though, is that the ASSESS-IT approach, as it stands now, cannot provide such information. The following section presents the rationale for an alternative framework, namely BPISS, that proposes a new approach that can be used to identify NFR that affect IS performance and to model the behaviour of the IS and the BP as they evolve over time.

### THE BPISS FRAMEWORK

The results derived from the ASSESS-IT framework suggest that the relationship between BP and IT can be described as the relationship between business processes and the information system that support those processes, and not as a three layered structured (BP, IS and CN) as it was thought in the ASSESS-IT framework. Furthermore, the results from the ASSESS-IT framework found that in order to depict the interactions between BP and IS it is necessary to portray IS non-functional requirements, in particular IS performance requirements. This section describes a new simulation framework, namely BPISS, to develop simulation models that depict business process and information systems performance. The BPISS framework is summarised as follows and is depicted in Figure 4.

1. Develop BPS model and identify possible IS scenarios. The framework uses a BPS model to represent current processes, identify process bottlenecks and propose possible IS solutions.
2. Identify IS functional and non-functional requirements. Once IS solutions are proposed, its functionality is described using current IS modelling techniques. Non-functional requirements are described and identified in two ways. First, current IS modelling techniques are used to derive a list of IS performance requirements. Second, the list is complemented using the BPS model to investigate

other parameters that may affect BP performance and that are related to the proposed IS.

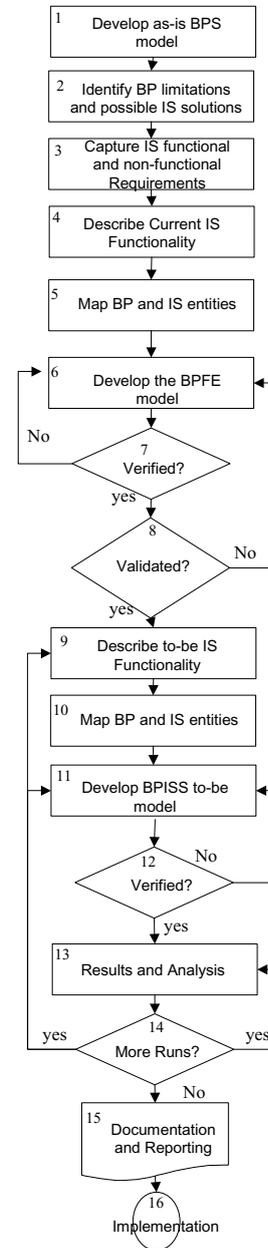


Figure 4 BPISS Framework

3. Develop the BPISS model. To develop a model that reflects both process and IS performance, the BPISS framework uses a preliminary model, namely BPF model to validate the model against the original BPS model. The BPF model aims to portray the way current business processes behave considering the changes that the functionality of the current system

(automated or manual) produce on the model. The BPFE model identifies those business entities, namely Record Entities (RE), that are used in the business process model and that are affected by the system. To reflect the way the system affects RE, the BPISS decomposes the RE into Field Entities (FE), which in turn represent the information contained in the RE. FE should describe the information used by the system to trigger events or to produce changes on business entities. Based on the BPFE, a new model is created so it reflects the functionality of the new IS (the BPISS model).

## Experimentation and Results

Because the ASSESS-IT had already developed a BPS model this was used in Step One of the BPISS framework. Similarly, the IS proposed in ASSESS-IT was used in the BPISS framework. The proposed IS should accomplish the aims described in the following list.

- The IS should automatically update inventory levels so a real-time inventory level could be monitored by both organisations.
- The IS should automatically produce a backorder whenever a given product is out of stock.
- When a backorder is produced or the inventory level of a given product is below the figures established by the organisations, the IS should request the replenishment of this product.
- When a replenishment cargo is delivered, the inventory levels should be updated and reflected in real-time.

Two IS non-functional requirements related with performance measurements were identified for the proposed IS. The first and most important requirement was requested by Org-A and establishes that the overall delivery time, including backorders, should be 24 hours, for products delivered within the city of Thessaloniki, or within 48 hours for the rest of northern Greece. This means that it is expected that the introduction of the new IS would reduce current delivery times so they fit the requirements previously mentioned. The second requirement was obtained during experimentation of the BPS model. It was detected that the backordering process was a major system bottleneck, and that delivery times depended on this process. It was demonstrated that when reducing the backorders percentage from 30% to 5%, the overall processing time was significantly reduced. Therefore, a performance requirement that was not identified before is related to the percentage of backorders

produced by the IS. The following paragraphs describe the results obtained from the BPISS model.

The results of the BPISS model reported a significant reduction in the totality of lead times, in particular, backorder lead times. Table 1 shows that backorder lead-time for both, Thessaloniki and Northern Greece were reduced in more than 80%.

Table 1 Table 1 BPISS Model Results

	BPFE Model	BPS/ISS Model	Difference	Difference (in %)
Orders (Thessaloniki) in hrs.	29.605	23.659	-5.946	-20.0844452
Orders (Northern Greece) in hrs.	44.344	44.127	-0.217	-0.489355944
Backorders (Thessaloniki) in hrs.	276.971	39.998	-236.973	-85.55877691
Backorders (Northern Greece) in hrs.	279.147	46.498	-232.649	-83.34282654
Total (Thessaloniki) in hrs.	87.136	36.102	-51.034	-58.5682152
Total (Northern Greece) in hrs.	97.407	52.885	-44.522	-45.70718737
Number of orders produced	3097.333	3166	68.667	2.216971827
Number of backorders produced	801.333	1125.667	324.334	40.47430968
% of backorders	25.87170963	35.55486418	9.683154553	

The reductions on lead-time, however, were still below the organisational targets. An interesting observation in Table 1 is that the percentage of backorders produced by the BPISS model reported an increase of nearly 10%, a situation that contradicts the assumptions made in Step Two. A possible reason that causes this situation is that the minimum product stock level used in the model (10 products) produces a greater number of backorders. This event, though, does not affect backorder lead-time because the new system schedules delivery times in a more accurate manner than the manual system. Experimenting with the BPISS model showed that a possible way of reducing backorders and consequently lead time is to increment the minimum stock level for each product. The results suggested a minimum stock level of 100 items for each product reduced the number of backorders having only 5% of backorders.

## CONCLUSIONS AND FURTHER RESEARCH

This paper provided evidence that despite the fact BP and IT interact in practice, existing BP and IS design approaches and modelling techniques do not provide a clear guidance of how to address the relationship between BP and IT. Trying to address this problem, the ASSESS-IT framework proposes the use of BPS and CNS to coordinate the design of business process and IT simulation models and depict the effect that changes on any of these domains may have on the others.

The results derived from the ASSESS-IT framework suggest that the relationship between BP and IT can be described as the relationship between business processes and the information system that support those processes, and not as a three layered structured (BP, IS and CN). Furthermore, the results from the ASSESS-IT framework found that in order to depict the interactions between BP and IS it is necessary to portray IS non-functional requirements, in particular IS performance requirements.

This paper used this knowledge to propose and test a new simulation framework, namely BPISS, to develop simulation models that depict business process and information systems performance.

The BPISS simulation results demonstrated that it is possible to obtain performance measurements of the IS and depict the way the insertion of IS affects BP performance. For example, the model provided quantifiable metrics of the IS, such as the number of backorders that the IS produces over a given period of time given a particular organisational context. These measurements helped to investigate the way IS may affect process performance. For example, new backordering delivery lead times that considered the effects that the IS has on the backordering process were obtained.

The experiment showed that depicting the behaviour of the IS and the effects that the latter would have on the processes is feasible, however, this was not an easy task. Despite the simplicity of the case study used to test this framework the development of the model proved to be complex. It is thought that the higher the complexity of the IS the harder the construction of the simulation model. Regardless of these drawbacks, the results of the experiment provided alternative information to assess the impact of IS on process performance. Furthermore, it was noticed that process and IS functionality are intrinsically related and further research is needed to analyse this relationship in more detail.

Finally, one of the major constraints when developing the simulation models was due to the fact that the discrete-event simulation tool used in the project was designed to simulate business process, hence it offered limited capabilities to simulate the information system functionality. Further research in this area is also needed in order to identify simulation tools that offer better capabilities to model the elements required to simulate the IS functionality.

## REFERENCES

- Armistead, C., Harrison, A. and Rowlands, P. (1995) Business process re-engineering: Lessons from operations management. *International Journal of Operations & Production Management*, 15(12), pp. 46-58.
- Banks, J., Carson, J. S., Nelson, B. L. and Nicol, D. M. (2000). *Discrete-event System Simulation*. 3rd ed. Upper Saddle River, NJ: Prentice-Hall.
- Childe, S. J., Bennett, J. and Maul, J. (1994) Frameworks for Understanding Business Process Re-engineering. *International Journal of Operations & Production Management*, 14(12), pp. 22-34.
- Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Boston, MA: Harvard Business School Press.
- Davenport, T. H. and Short, J. E. (1990) The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, 31(4), pp. 11-27.
- Davenport, T. H. and Stoddard, D. B. (1994) Reengineering: Business Change of Mythic Proportions? *MIS Quarterly*, 18(2), pp. 121-127.
- Eatock, J., Paul, R. J. and Serrano, A. (2001) A Study of the Impact of Information Technology on Business Processes Using Discrete Event Simulation: A Reprise. *International Journal of Simulation Systems, Science & Technology, Special Issue on: Business Process Modelling*, 2(2), pp. 30-40.
- Garvin, D. A. (1988). *Managing Quality*. New York: Free Press.
- Grover, V., Fielder, K. D. and Teng, J. T. C. (1994) Exploring the Success of Information Technology Enabled Business Process Reengineering. *IEEE Transactions on Engineering Management*, 41(3), pp. 276-284.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, NY: Harper Collins Publishers.
- Juran, J. (1964). *Managerial Breakthrough: A New Concept of the Manager's Job*. New York, NY: McGraw-Hill.
- Schein, E. (1969). *Process Consultation: Its Role in Organisation Development*. Reading, MA: Addison-Wesley.

## BIOGRAPHY



**ALAN SERRANO** is a Lecturer of IS at Brunel University, where he also received his PhD in Information Systems and a M.Sc. in Data Communication Systems. He has published in the areas of business process and computer network simulation. His research focuses on simulation and the evaluation of information infrastructure changes on business performance. He has a wealth of expertise gained from his work experiences in Mexico, ranging from distributed systems to computer network design. You can reach him by e-mail at [alan.edwin.serranorico@brunel.ac.uk](mailto:alan.edwin.serranorico@brunel.ac.uk) and his web address is [www.brunel.ac.uk/~csstaes](http://www.brunel.ac.uk/~csstaes)

# CAPTURING INFORMATION SYSTEM'S REQUIREMENT USING BUSINESS PROCESS SIMULATION

Alan Serrano  
Department of Information Systems and Computing  
Brunel University  
Uxbridge Middlesex  
UB8 3PH, London, UK

## KEYWORDS

Information Systems Requirements, Non-Functional Requirements, Discrete-event Simulation.

## ABSTRACT

Systems requirements can be divided into two major groups: functional requirements and non-functional requirements (NFR). Functional requirements describe what the system does or is expected to do. In other words, they describe the *functionality* of the system. Non-functional requirements are concerned with describing how well the system delivers the functional requirements. Despite the fact NFR play a very important role during the software development process these have been overlooked by researchers and are less understood than other factors in software development. One reason for this might be because NFR are difficult to represent in a measurable way, something that impedes their proper analysis. This paper explores the feasibility of using Business Process Simulation (BPS) models to provide IS analyst with alternative ways to capture non-functional requirements.

## INTRODUCTION

One aspect that is consistently addressed by IS development methodologies is to match organisational needs with the proposed IS. Despite this fact, however, there are still some issues that need to be overcome when capturing IS requirements. Traditional IS methodologies, such as the SDLC, reflected a view that user requirements did not change over time, and thus conceptual models could be used to represent the functionality of the system (Rolland and Prakash, 2000). This belief, however, has now been challenged. Information systems have to adapt to this environment, which in turn implies that requirements are not stable. In order to adapt to this environment, IS practitioners advocate that requirement validation must now be compared against organisational needs and not against the system functionality (Rolland and Prakash, 2000). Only in this way, they argue, can information systems adapt to the ever-changing organisational needs. To address these problems, requirements engineering extends the remit of the traditional IS modelling approach that answers the

question What does the system do? to an approach that answers the question Why is the system like this?

Capturing requirements is part of the software development process and is concerned with understanding the needs and wishes of the current and new system and finding mechanisms to portray these needs. Most of the IS methodologies argue that this task should be undertaken thoroughly otherwise it could cause user dissatisfaction. Requirements Engineering (RE) is the IS domain that studies how to develop systems that meet user requirements in the best possible way. Zave and Jackson (1997) define requirements engineering as the branch of software engineering that is concerned with the analysis and capture of organisational goals considering the constraints they impose on software systems. Systems requirements can also be divided into two major groups: functional requirements and non-functional requirements (Sommerville, 1997; Bennett et al., 1999). Functional requirements describe what the system does or is expected to do. In other words, they describe the *functionality* of the system. Functional requirements representations usually illustrate the way the system operates, details of the inputs and outputs of the system, and the relationships between the data that the system will hold. Non-functional requirements are concerned with describing how well the system delivers the functional requirements. Non-functional requirements are usually expressed as performance criteria, volumes of data that the system should hold, and security considerations.

The following section describes the advantages and limitations of IS modelling techniques to capture functional and non-functional requirements.

## MODELLING FUNCTIONAL REQUIREMENTS

Rich Pictures, Conceptual models, Data Flow Diagrams (DFD), Entity Relationship Diagrams (ERD), State-Transition Diagrams, IDEF1x, and Object-Oriented (OO), such as the Unified Modelling Language (UML), can be mentioned as the most dominant IS modelling techniques (Giaglis, 2001). Models in the IS domain can be used to represent many different aspects of the IS process. Consequently, the major problem when representing user requirements (functional or non-functional) is to identify which is the most appropriate

technique for the purposes the analyst wants to communicate.

Most of the IS modelling techniques specialise in different aspects, depending on the stage at which they are applied. For example they can be used to understand either the overall function of the system in question, to understand IS data structures, or to model the processes involved in the IS. Table 1 shows a classification of modelling techniques according to the stage that they can be applied and the aspect they address.

Table 1 Classification of Modelling Techniques (adapted from Avison and Fitzgerald, 2003)

Stage/Aspects addressed	Overall	Data	Process
Strategy	Rich Pictures		
Investigation & Analysis	Rich Pictures Objects Matrices Structure diagrams Use Cases	Entity Modelling Class Diagrams	Data Flow Diagrams Entity Life Cycle Decision Trees Decision Tables Action Diagrams Root Definitions Conceptual Models (UML)
Logical design	Objects Matrices Structure diagrams	Normalisation Entity Modelling Class Diagrams	Decision Trees Decision Tables Action Diagrams
Implementation	Objects Matrices Structure diagrams	Normalisation	Decision Trees Decision Tables Action Diagrams

Most of the techniques described in Table 1, though, are aimed to depict functional requirements. The following section describes current alternative techniques to model non-functional requirements (NFR).

### MODELLING NFR

Information systems can be determined by their functionality and also by properties of the whole system such as operational costs, performance, reliability, maintainability, portability, and many others. These constraints, also named goals, quality attributes, and *Non-functional Requirements (NFR)*, play a very important role during the software development process, since they usually work as the selection criteria among a variety of decisions in the development process. Despite these facts, non-functional requirements have been overlooked by researchers and are less understood than other factors in software development. One reason for this might be because NFR are difficult to represent in a measurable way, something that impedes their proper analysis. Mylopoulos et al.(1992), Nixon (1998), and Nuseibeh and Easterbrook (2000) have identified other major problems encountered when dealing with NFR:

1. There is not a formal definition or a complete list of NFR.
2. NFR usually interact with each other, a situation that can cause conflicts and tradeoffs with implementation techniques.
3. NFR are difficult to understand and represent since they have a global impact on the future system.
4. In order to produce a system that meets the NFR, it is important to consider the organisation's

characteristics. These, however, vary from one organisation to another.

5. In general, NFR represent properties of the whole system, therefore, it is almost impossible to verify them in terms of individual components.

Trying to address these problems, academics and practitioners have proposed many ways to model non-functional requirements. Mylopoulos et al., (1992) and Chung and Nixon (1995), for example, proposed a NFR framework to capture and relate non-functional requirements. The NFR framework uses a goal-oriented approach to capture NFR. After NFR are captured and analysed, the NFR framework finds links between them in order to determine the impact that a given decision would have on the requirements. Nixon (2000) extends this work and applies this framework to specify a particular group of NFR: *performance requirements*. Nixon adapts the NFR framework to integrate and catalogue a different number of knowledge of performance and information systems, including performance concepts, software performance engineering, and information systems development knowledge such as requirements, design, implementation and performance. Similar to this work, Cysneiros and do Prado Leite (1999) integrate non-functional requirements into traditional conceptual data models, namely Entity Relationships (ER). The objective is to represent NFR and understand their impact on database modelling design.

These approaches can be useful to elicit NFR and to match them against IS design in order to meet NFR. The approaches, however, do not provide the means to assess whether the proposed IS meets the NFR identified previously, neither are they useful to investigate how these requirements may affect the organisational performance.

The following sections describe the way Business Process Simulation (BPS) models can be used as a complementary technique to capture non-functional aspects of a proposed IS solution. To this end, the following section describes a case study that will be used as an example to provide evidence to support this theory and subsequent sections analyse the results of the BPS models and explains the way these results can be used to capture NFR.

### THE CASE STUDY

The case study presented here consists of two collaborating organisations in Greece. One company is a branch of a major multinational pharmaceuticals organisation (we will refer to this company as Org-A), while the other is a small-sized regional distributor of Org-A's products (we will refer to this company as 'Org-B'). The case study was carried out within a single business unit, which deals with hospital consumables. The business unit imports products from other Org-A production sites across Europe. The goods are stored in a

warehouse that operates as a central despatch point for all products, which are then distributed to the company's customers via a collaborating distributor, namely Org-B. Org-B responsibilities include:

- a) Maintaining an adequate inventory of products to fulfil the orders.
- b) Distributing the ordered products to customer premises.

Org-B has to operate within rigorous deadlines. The agreement between the companies, stipulates that each order has to be fulfilled within 24 hours for products delivered within the city of Thessaloniki, or within 48 hours for the rest of northern Greece. Org-A management noted, however, that these targets are rarely met in practice. A brief analysis by the companies seemed to attribute the problems to some inefficiencies within the ordering system as well as difficulties being experienced by Org-B in maintaining their inventory at an optimal level. The effects that these inefficiencies caused were seen as a major source of customer dissatisfaction, so an in-depth analysis of the problem was commissioned. The main objectives of this study were:

- a) To examine the existing business processes that were felt to be responsible for long lead times for order fulfilment.
- b) To determine the sources of problems and propose alternative solutions.
- c) To evaluate the potential of introducing appropriate IS to improve communication between the two companies.

### **USING BPS MODELS TO CAPTURE NFR**

This section describes the way BPS models can be used to capture NFR. To this end the simulation exercise was divided in two phases: One, the development of the as-is model and two, the development of different business process scenarios (to-be models). Because this paper aims to illustrate the use of simulation models to capture NFR, the design of the BPS models will be skipped, concentrating on the results provide by such models. For more information related to design of simulation models refer to Law and Kelton (2000).

The findings of the as-is business process confirmed the concerns of the companies that delivery times were much longer than the agreed targets. Even when no backorders were required, deliveries to Thessaloniki took 38 hours (target time was 24 hours), while deliveries to the rest of northern Greece took 60 hours (target time was 48 hours). Furthermore, when those orders that had items that were out of stock were included, the average time to deliver backorders rose to 82 hours. These figures suggested that the backorders were causing severe problems, so warranted further analysis.

Any order that required some out-of-stock products would effectively result in the order being divided into two separate orders; those products that were available, and those that were out-of-stock. The available products would be delivered as soon as possible, but the out-of-stock products would need to be ordered from Org-A, who would then add them to the next scheduled warehouse replenishment delivery, resulting in long delays from the order being submitted and the particular products being delivered. Hence, times were recorded for backorders. When this figure was analysed it was found that the time taken from the backorder being generated to delivery accounted for 168 hours for Thessaloniki and 190 hours for northern Greece. Consequently, a series of business process scenarios, including one using an IS solution, were designed trying to address the problems found in the as-is model. The following sub-section explains the experiments proposed.

### **Business Process Experimentation: To-be models**

The results rendered by the as-is model indicated that order and backorder processes had some limitations in terms of process design. Thus, before proposing an IS solution, it was decided to investigate different scenarios to improve these processes without using information technology. This would help to provide a better understanding of the way processes operate and to propose an IS solution that better fits the problems found in the business process model.

Three scenarios that would possibly alleviate the problems with backorders were considered. These are described in the following list.

1. Faxing backorders. Backorders were generated by Org-B and then held until the Friday evening, before being sent by post, which takes 2 days, to Org-A. For the purposes of analysis, the solution proposed was to fax the backorders to Org-A, instead of sending them by post. It was assumed that by reducing the time the backorders spent in the mail system would have a significant impact on the delivery times.
2. Org-B sends the backorder list twice a week. Instead of waiting until Friday to send the backorder list, this is sent Tuesday and Friday evenings.
3. A final scenario that included the use of an IS system was designed. The two companies share the same database to allow Org-A to have up-to-date information on stock levels at Org-B, and hence adjust replenishment shipments accordingly. The backorder list is generated automatically so Org-A knows at any moment the real-time stock levels in Org-B warehouse. It was thought that this would have an enormous impact on the delivery times, as a backorder that was generated on a Monday could now be transmitted immediately, rather than being delayed until the Friday before being forwarded.

### To-Be Business Process Results

The results obtained from running scenario 1 showed that although the time taken to send the back orders by mail was reduced from 2 days to a matter of minutes, the reduction in time taken to deliver the entire order was reduced by only 1 hour for orders and remained almost the same for backorders (times recorded for Thessaloniki). After a thorough analysis, it was found that this situation was due to two major reasons. Firstly, because of the organisation's policy, orders were retained until Friday afternoon to be faxed. Despite the fact orders arrived in a matter of minutes to Org-A, Org-A employees take about 8 hours to process an order list. Therefore, orders sent by fax on Friday afternoon did not have enough time to be processed on the same day and they have to wait until Monday. In the original scenario, orders took two days to arrive at its destination by normal mail. Considering that the mail works on Saturdays, the backorder list also arrived on Monday. Hence, both scenarios process the list almost at the same time. The second, and most important reason, is that there is only one employee working in the warehouse of Org-B, who, is busy nearly 97% of his working time. Consequently, an extra experiment was performed adding one more warehouse employee, resulting in an 11 hours decrease for orders to Thessaloniki. Backorders, however, remained the same.

The second scenario was to schedule the replenishment shipments to be sent twice a week instead of once. This resulted in a reduction in delivery times for backorders, but it was much smaller than anticipated (11 hours for back orders). This was due to the same problem identified in scenario 1 related to the warehouse employee. When the time was measured combining the scenarios of having two employees, faxing backorders twice a week, ordering times were recorded as 27 (11 hours reduction) and backorders as nearly 128 (40 hours reduction).

Scenario 3 addressed the only real IS-based solution, in which, both companies share a database. Sharing a database gives Org-A a better idea of the replenishment requirements of Org-B. The results did not show a noticeable reduction in the delivery times for the orders that had in-stock products, on the contrary, an increase of 29 hours was noticed for orders. The problem, as in the previous scenarios, was due to warehouse employee workload. It was reported that he was busy 99% of his working time. The increase of utilisation (more than 2%) was due to the fact he had to deal with a slightly higher number of backorders. Those products that required backorders, however, showed a substantial reduction of nearly 74 hours. This was mainly because the backorder list would no longer need to be created, as it would be generated in conjunction with the normal replenishment shipment. A final experiment was created which

combined the results from scenario 3 with an extra warehouse employee. The results were as expected, since there was a reduction of 10 hours for orders and 82 for backorders. The times reached in this scenario were the best in comparison to the as-is scenario, though, they were still distant from Org-A and Org-B targets.

The results for the average delivery times of the entire order, and the average delivery time of backorders for each of the scenarios described before are shown in Figure 1.

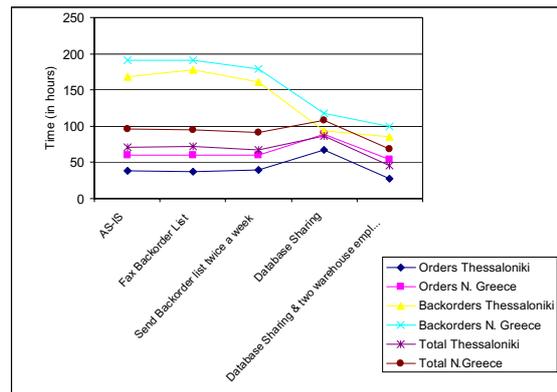


Figure 1 Delivery Times Scenarios

### CONCLUSIONS

The results obtained from the business process scenarios indicated that the majority of problems of Org-A and Org-B were related to business process design and not to information technology deficiencies. It was found that a major limitation of the way Org-B operates was related to the resources assigned to the different activities performed by Org-B. It was found that the only employee working in the warehouse of Org-B was busy 97% of his working time and that changes to the business process were constrained by this issue. Because this problem is related to the way processes operate, the insertion of an IS does not provide any improvement in this respect, and the contrary may be true for some cases. It was observed that the warehouse employee needed to work more because the information system produced more workload in the activities in which this resource is used (backorders). When another warehouse employee was used in the model, many problems related to performance in the activities he was involved in were alleviated.

In order to avoid these problems, the experience gained during the BPS exercise suggests that a thorough analysis of the BP is needed, and especially, it is necessary to identify:

- a) Business process limitations.
- b) Different business process alternatives.

- c) System bottlenecks where IS-based solutions can be implemented.

In order to identify these three issues, performance measurements of the BP are needed. To this end, the experiments proved that BPS is able to provide this information.

Perhaps, the most important outcome of the experimentation with the to-be business process models is the identification of a system bottleneck, which in turn, is the key to depict the impact of the IS on the business processes. Many business process methodologies suggest that applying IT with only the intention of automating business processes, and presumably improving performance, usually render disappointing results (Hammer, 1990; Kettinger and Groover, 1995). The results of the experiments performed on the IS-based scenario described, corroborate this statement. It was demonstrated that despite the fact that the use of technology was introduced (database sharing) the results obtained from the models were far below the expectations of the organisations. The experiments performed, however, helped to identify that the backorder sub-process was a major bottleneck of the ordering process. Initial assumptions were that the insertion of an IS would reduce the backordering processing time. Nevertheless, the results showed that this was not the case. Despite the fact that transmission times were substantially reduced, the performance of the system was still not satisfactory. To understand the reasons why the business process simulation exercise did not reflect what the IS should supposedly have delivered, the following paragraphs analyse the IS functionality in more detail.

The IS was thought to function in the following way. It was assumed that information between both companies would now be shared. Therefore, the system in Org-A would now be able to know when the stock level of any given product was below the organisation's policy. When a given stock level is below its set figures to fulfil demand, the system would set an alarm to Org-A so they could be aware that a replenishment cargo was needed. Disregarding some exceptional cases (i.e. unexpected demand in a single product) it was expected that this new way of monitoring product stock levels would reduce the number of backorders since Org-A would have information about which products needed to be replenished before they were out of stock. The business process model, however, did not represent this kind of behaviour. The business process model only reflects the time reduction due to the insertion of the IS. To reproduce the effects that this new information system would have on the business processes it is necessary to know, rather than the reduction time, the number of backorders that would be reduced due to the use of the new IS. Bearing in mind that the information system would more effectively

control the inventory replenishment policies, which in turn would reduce the number of backorders produced on the system, a final experiment was proposed. The experiment consisted of gradually decreasing the percentage of backorders produced in the business processes model from 30% to 5%. The results proved that the overall ordering processing time could be reduced significantly if the percentage of backorders is reduced. The experiments showed an average of 31 hours for the delivering of orders and backorders to Thessaloniki, and 50 hours to northern Greece. These figures are closer to the targets set by both organisations.

Summarising, the simulation exercise presented in this paper provided evidence that business process simulation can be used by IS practitioners to better understand the way the organisation operates and to identify systems requirements that could be overlooked when using traditional IS modelling techniques. For example, the BPS models used in the case study helped analyst to identify that a system that could help to improve Org-A and Org-B orders delivery time should be one that not only improves processing time, but more importantly, one that reduces the number of backorders produced by the system.

#### **FURTHER RESEARCH**

The to-be BPS model provided information that can lead to the identification and measurements of NFR. For example, it was detected that to predict the impact that the introduction of an IS may have on the business processes, it was necessary to identify the number of backorders that would be produced by the new information system. These figures can only be obtained by modelling the behaviour of the information system as it is used over time. Thus, in order to portray the benefits that the use of an information system brings to the business processes it is necessary to verify that the IS satisfy certain requirements, in particular non-functional requirements.

One reason non-functional requirements are difficult to analyse is due to the fact they have a global impact on the organisation and the system itself. Similarly, it is argued that to produce systems that meet non-functional requirements it is important to consider the organisation's characteristics.

This paper used BPS techniques to represent the behaviour of the IT considering the organisational context (business process) in which they operate. One of the problems using traditional BPS techniques is that it cannot be used to portray behavioural measurements of the way a given IS functionality may provide to the BP. These measurements, according to the analysis of the results derived from this exercise are likely to be IS non-functional requirements. This paper can be used as the basis to propose an approach to depict the way IS requirements may affect BP performance with the aid of

simulation techniques. Using this results and experience, a simulation framework to depict IS and BP interactions can be proposed. This framework can help to:

- a) Identify non-functional requirements. Non functional requirements are not always clearly defined by the users. For example, a non-functional requirement of the case study was to offer control of the inventory system in a more efficient manner. Further analysis using the BPS model indicated that the "real" requirement was to have as few backorders as possible. This was identified by experimenting with the model and isolating the parts of the system (processes) that were concerned with the backordering process.
- b) Verify that the IS satisfies non-functional requirements. Running a simulation model that depicts both IS and BP performance can be used to verify that the system satisfies user requirements.
- c) Identify the variables that affect IS and/or BP performance. Once a given non-functional requirement is identified, simulation can be used to isolate the process (together with the IS used to support this process) needed to satisfy this requirement. Experimentation can be used to investigate the variables (e.g. resources, entities, etc.) that may affect IS and/or BP performance.
- d) Identify conflicts and tradeoffs between non-functional requirements. Experimentation can be used to identify the effects that changes in one user requirement may have on the others.

## REFERENCES

- Bennett, S., McRobb, S. and Farmer, R. (1999). *Object-oriented Systems Analysis and Design Using UML*. London: McGraw-Hill.
- Chung, L. and Nixon, B. A. (1995) Dealing with Non-functional Requirements: Three Experimental Studies of a Process-oriented Approach. *Proceedings of the 17th International Conference on Software Engineering*, Seattle, WA, April 24-28. pp. 25-37.
- Cysneiros, L. M. and do Prado Leite, J. C. S. (1999) Integrating Non-Functional Requirements into Data Modelling. *Proceedings of the IEEE International Symposium on Requirements Engineering*, Limerick, Ireland, June 7-11. IEEE, pp. 162-171.
- Giaglis, G. M. (2001) A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques. *International Journal of Flexible Manufacturing Systems*, 13(2), pp. 209-228.
- Hammer, M. (1990) Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*, 68(4), pp. 104-112.
- Kettinger, W. J. and Groover, V. (1995) Toward a Theory of Business Process Change. *Journal of Management Information Systems*, 12(1), pp. 9-30.

- Law, A. M. and Kelton, D. W. (2000). *Simulation Modelling and Analysis*. 3rd ed. New York, NY: McGraw-Hill.
- Mylopoulos, J., Chung, L. and Nixon, B. A. (1992) Representing and Using Non-Functional Requirements: A Process Oriented Approach. *IEEE Transactions on Software Engineering; Special Issue on Knowledge Representation and Reasoning in Software Development*, 18(6), pp. 483-497.
- Nixon, B. A. (1998) Managing Performance Requirements for Information Systems. *Proceedings of the First International Workshop on Software and Performance*, Santa Fe, New Mexico, October 12-16. ACM Press, pp. 191-144.
- Nixon, B. A. (2000) Managing Performance Requirements for Information Systems. *IEEE Transactions on Software Engineering*, 26(12), pp. 1122-1146.
- Nuseibeh, B. and Easterbrook, S. (2000) Requirements Engineering. *Proceedings of the 22th International Conference on Software Engineering*, Limerick, Ireland, June 4-11. ACM Press, pp. 35-46.
- Rolland, C. and Prakash, N. (2000) From Conceptual Modelling to Requirements Engineering. *Annals of Software Engineering*, 10(1/4), pp. 151-176.
- Sommerville, I. (1997). *Software Engineering*. 5th ed. Wokingham: Addison-Wesley.
- Zave, P. and Jackson, M. (1997) Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 29(4), pp. 315-321.

## BIOGRAPHY



**ALAN SERRANO** is a Lecturer of IS at Brunel University, where he also received his PhD in Information Systems and a M.Sc. in Data Communication Systems. He has published in the areas of business process and computer network simulation. His research focuses on simulation and the evaluation of information infrastructure changes on business performance. He has a wealth of expertise gained from his work experiences in Mexico, ranging from distributed systems to computer network design. You can reach him by e-mail at [alan.edwin.serranorico@brunel.ac.uk](mailto:alan.edwin.serranorico@brunel.ac.uk) and his web address is [www.brunel.ac.uk/~csstaes](http://www.brunel.ac.uk/~csstaes)

# JOINT SIMULATION MODELING TO SUPPORT STRATEGIC DECISION-MAKING PROCESSES

Corné Versteegt, Sander Vermeulen, Eric van Duin  
Faculty of Technology, Policy and Management  
Systems Engineering Group  
Delft University of Technology  
P.O. Box 5015, 2600 GA Delft, The Netherlands  
E-mail: cornev@tbm.tudelft.nl

## KEYWORDS

Collaborative Business Engineering, Joint Simulation, Strategic Decision-Making, Airfreight Industry.

## ABSTRACT

The airfreight industry is highly dynamic. Airline companies need to adapt their processes continuously. This case study was carried out to support an airline company in designing a strategy for airfreight handling. The goal of our research was to explore the operational implications of strategic decisions on the new structure of airfreight handling processes. A Collaborative Business Engineering approach was followed in which simulation models were constructed jointly with the management of the airline company. Simulation models of the freight handling processes were built to provide insight in alternative designs of warehouses. Base models were constructed to save time during the joint modeling sessions. During the group sessions the base models were adapted and expanded jointly with the management. After the group sessions extensive experiments were conducted and the results were presented to the management. The CBE approach was applied successfully. The simulation models and results were valued highly by the management. The management had high levels of trust in the models because of the joint modeling and the 3D animations. In the end the management was able to study a 'richer' set of alternatives; more alternatives and more detailed insight in each alternative was gained.

## 1. INTRODUCTION

Our research was carried out for a large airline company in the Netherlands, which participates in one of the leading global airline consortiums as an independent European partner. The airline company transports more than 15,7 million passengers and 621,000 tons of cargo and mail. Their route network connects 145 cities in 67 countries. The cargo-flights use the home-airport as a hub, which is a central point for all flights. The passenger and cargo flows have increased strongly in the last years. The home airport developed plans to restructure and expand at the time of our research. This forced the management of the airline company to think about the future airfreight handling processes. This research was started to provide the management insight into future alternatives for cargo handling,

e.g. designs and locations of new warehouses, and new freight handling procedures.

The design of airfreight handling processes is a complex activity. The problem setting is characterized by a large design space. A large number of degrees of freedom on several axes exists; a lot of actors are involved, many technological questions have to be answered, and a lot of uncertainties have to be dealt with (Babeliowsky 1997). Among the actors are airport authorities, central and local government, freight handlers, airline consortium partners, and customs. Each actor has its own goals and objectives, which are frequently conflicting. This leads to a complex multi-actor setting and an intransitive problem setting, where it is impossible to select a single alternative that is preferred to all other alternatives (Dunn 1981). The actors make their own decisions that cannot be influenced by the airline company. Such decisions, however, influence the freight handling processes, e.g. regulations of the airport authorities. The problem is technical complex. The possibilities of automated airfreight handling have to be taken into account, due to the relative high labor costs in the Netherlands. The airline company has to deal with a large number of uncertainties in economic developments. The prices of freight transport and transported volumes are dynamic and influence the design of the warehouses. After September 11th 2001 the security regulations have been increased. This has severe consequences on the freight handling processes; freight is thoroughly checked before flights.

Due to the complex problem setting there is little consensus on the new structure of the airfreight handling processes within the management. The goal of our research is *to provide the management insight into possibilities and limitations of new structures for airfreight handling*, at current and alternative locations, given growing cargo flows and restructuring of the home airport. We followed a Collaborative Business Engineering approach supported by simulation.

After this introduction the Collaborative Business Engineering is discussed in section two. Our Collaborative Business Engineering approach,

abbreviated to CBE, supported by simulation is presented in section three. The case study in which the CBE approach was applied is presented in section four. Section five presents some lessons learned during the case study. This paper ends with a number of general conclusions.

## **2. COLLABORATIVE BUSINESS ENGINEERING**

Modern organizations face an almost constant need to evaluate their strategies, processes, and systems (Drucker 1988, Hammer 1990, Davenport 1994). Organizations continually have to adapt to changes in their environment, such as new legislations, new partners and changed market demands. Meanwhile organizations must satisfy continuous increasing internal demands with respect to operating more efficient and more effective. Organizations can apply the principles of Business Engineering in order to deal with these issues. Business Engineering is “organizational transformation focusing on integral design of information technology, organizational processes, and structures” (Hammer 1990). A number of different Business Engineering approaches have been presented during the last years (see for an overview Meel 1994). In general, BE approaches are not suited to facilitate multi-actor settings. Our problem area, the airfreight industry, is characterized by a multi-actor setting (Babeliowsky 1997). We extend Business Engineering to a Collaborative Business Engineering approach supported by simulation (Maghnouji & Versteegt 2003). Collaboration is the process in which two or more individuals with complementary skills interact to create a shared understanding that none had previously possessed or could have come to on their own (Schrage 1990).

We expect that by applying the CBE approach decision-making processes will be more efficient and effective. The lead-times will be decreased (efficient) and the outcomes of the decision-making process will be supported by all actors involved (effective). Actors in intra- or multi-organizational settings have conflicting objectives. All actors want to make sure their points of view are represented in the design and that the design satisfies their interests. Integral solutions are needed in this multi-actor setting.

To reach an integral solution actors need a common frame of reference, a shared space (Schrage 1990, Senge 1994). If the shared space is not created, actors will keep living in their “own world” and communication will be impaired making it very difficult to reach integral solutions. Within complex design processes humans are conflicted with bounded rationality (Simon 1969). There are practical limits to human rationality, which makes it hard or even impossible to find an optimal solution. By following the CBE approach we limit the effects of bounded rationality by combining knowledge and

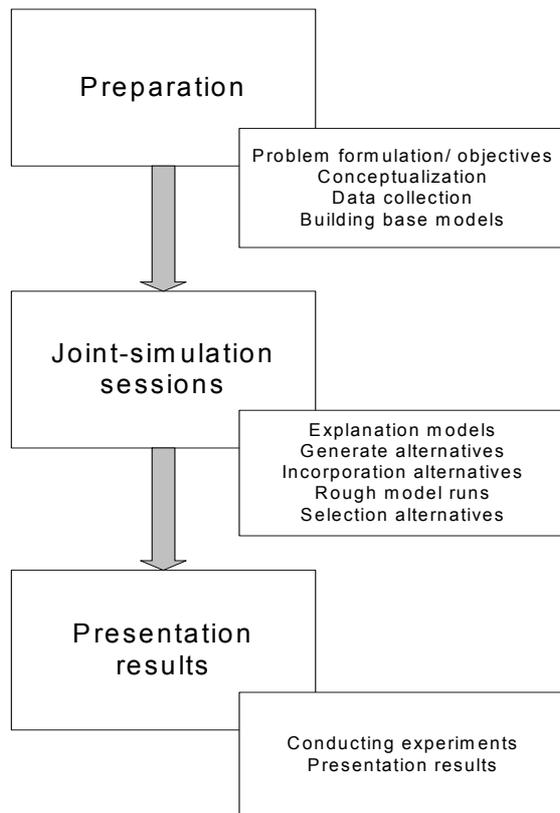
skills of actors from different disciplines. The CBE approach does not lead to an optimal solution, but to an integral solution; one that satisfies involved actors. Crucial aspect of the CBE approach is a shared space of understanding. *Within the CBE approach we develop a shared space of understanding by jointly constructing simulation models.* The models are projected on a central screen, visualizing the shared space of understanding. “Tacit” mental models of each participating actor are made explicit in simulation models to support discussions. Simulation offers other advantages. Simulation is used to study the operational aspects of a system. By working on the operational level design choices appear that would have been overlooked otherwise. The ability to work in a quantitative way on the operational level enhances the process of strategic decision-making. Simulation also offers possibilities to compare different designs of the airfreight handling processes. Traditional approaches to simulation studies are iterative processes that contain the following steps: problem formulation, setting of objectives, model conceptualization, data collection, building of the model, verification, validation, experimental design, model runs and analysis, documentation and reporting (Law & Kelton 1991, Banks 1998, Zeigler 2000). Traditional simulation approaches are not suited for joint simulation modeling. First, traditional simulation approaches have long throughput times. During joint simulation sessions there is only limited time available. Managers lack time and can only spend little time in group sessions. Second, in traditional approaches model builders construct the simulation model mostly on their own. In joint simulation modeling the model is build together with the problem owner. Traditionally problem owners are only little involved in constructing simulation models, they provide the boundaries, and questions that the models needs to answer. In joint simulation modeling the problem owners are participating actively in constructing simulation models. During joint modeling sessions laymen are involved in constructing simulation models. The problem owners are not familiar with simulation. A number of elementary simulation principles have to be explained to laymen. The problem owner has to make model assumptions and decisions about detailed aspects of each alternative during the group sessions. This enables the actors to gain shared understanding in all aspects of the alternatives. Third, in traditional simulation approaches there is a lot of time available for data collection. During group sessions data has to be available directly. Not all data can be obtained directly; assumptions have to be made. Fourth, traditional simulation requires large a number of model runs for obtaining reliable statistical output. During group sessions there is no time available for long simulation runs. It is more

important to gain rough insight into the alternatives. Short model runs are needed.

### 3. CBE APPROACH SUPPORTED BY SIMULATION

We develop a Collaborative Business Engineering approach in which simulation is used as a supporting tool. It consists of three phases; preparation, joint-simulation modeling sessions and presentation of the results, see figure 1.

Three different types of actors are involved in the CBE approach, facilitator, model builder and problem owner. The facilitator supervises the CBE process. The facilitator has knowledge of group design processes and guides the group in efficient decision-making. The facilitator structures the activities and secures the progress of each activity. The facilitator supports the decision-making process, without intervening in the actual content of the process. The model builders are responsible for constructing the simulation models. The problem owners define the problem definition, the model requirements and the evaluation criteria and provide input during the joint modeling sessions. This input ranges from small details, for instance the speed of a fork lift, to totally new designs.



**Figure 1. Collaborative Business Engineering Approach**

In the first phase, *preparations*, the preparations for joint-simulation modeling sessions are made. Good preparations are crucial, since the available time for the joint modeling sessions is limited. Managers have little time available and it is hard to bring them all together for a long time. Many time-consuming activities are performed before the joint modeling sessions, e.g. data collection. In the preparation phase the first steps of the traditional simulation approach are performed; the problem formulation, the clarification of the objectives, the conceptualization of the problem and the data collection of the basic processes. This requires time because different members of the problem owner may all have their own view on the design problem. These views must converge first before the joint modeling sessions can start. The facilitator is responsible for this process. The base models are constructed. Base models are simulation models that will be used as starting points in the joint-modeling sessions in the second phase. Base models are needed because it is impossible to construct a complete model from scratch within a joint-modeling session due to time limitations. One of the base models is the status quo model, which describes the current situation. This will be used to establish the shared space of understanding. Other base models should contain elements that are either expected not to change during the joint sessions (for example static objects such as buildings or railways) or that are impossible to model in a short time (for example control logic).

The goal of the second phase, *joint-simulation sessions*, is to jointly generate and evaluate alternatives. In the second phase the base models are explained in depth to the problem owner. The problem owners generate large numbers of alternatives by studying the base models. The facilitator encourages this creative process and structures it. The alternatives have to be incorporated in the base models. During group sessions the model builders and problem owner jointly create new simulation models. The problem owners generate alternatives, while the model builders construct models of these alternatives. This forces the problem owner to specify a lot of choices in each alternative and to make assumptions on how each alternative will be modeled. After a large number of alternatives have been generated fast model runs are performed to get a rough insight of the consequences of each alternative. This enables selecting the most promising alternatives and adapting or removing less promising alternatives. This procedure is repeated several times until only the most promising alternatives remain.

The final phase, *presentation of the results*, is conducted after the joint simulation sessions. Goal of this phase is to provide the problem owner with a thorough analysis of the chosen alternatives from the joint modeling sessions. Extended simulation models

of the most promising alternatives are constructed. These models are used to conduct numerous experiments to obtain statistically valid output of each alternative. This is not possible during the second phase, due to the lack of time. The results of the experiments are compared and presented to the problem owner. The responsibility for choosing one of the presented alternatives and eventually implementing this alternative lies at the problem owner. The CBE approach supports problem owners creating a number of promising alternatives, not in choosing one of the alternatives.

The CBE approach does not prescribe what simulation software to use. A number of criteria are given that the simulation software has to meet. The simulation software has to provide realistic animations. This is especially important when laymen are involved. Animations create a shared space of understanding. The modelers must be able to create models quickly. During groups sessions there is only limited time for model building. The software has to be flexible. Ideas from the problem owners must be implemented quickly, without major changes in the structure of the simulation models. The simulation software has to allow modeling at different aggregation levels. The level of detail of the model should be determined by the problem owners and not by restrictions of the software.

#### **4. CASE STUDY: AIRFREIGHT COMPANY**

##### *Phase 1: Preparation phase*

At the beginning of the preparation phase the airline company provided the problem definition and the research objectives. The problem definition and research objectives were too broad and ambiguous. The problem definition and objectives had to be sharpened in order to be able to construct simulation models. The second step was the construction of conceptual models that define the problem situations in broad terms. The conceptualization resulted in an overview of the different types of cargo and the current structure of the airfreight handling processes. Several types of conceptual models were constructed, mainly graphically oriented, like flow diagrams and layouts of airport and warehouses. The layouts and flow diagrams were combined in order to create an overall view of the freight handling processes. The conceptual models were created in such a way that they could be easily translated into empirical simulation models. The third step was to collect data as initial input for the simulation models. The management of the airline company expressed a wish to use real-world data rather than using stochastic distributions in the simulation model. The main reason to use real-world data was the lack of confidence in stochastic distributions by the management. The arrivals patterns of planes and cargo is capricious. This makes it difficult to fit stochastic distributions from the real-world data.

Collecting the real-world data led to several problems. The real-world data was retrieved from database systems containing all cargo information of the last years. The different database systems had to be merged to be able to retrieve input data for the simulation models. This was a time-consuming process, resulting in a huge database that was difficult to access and process. The final challenge was to link the database to the simulation models. In the database each line represents a single cargo load. In order to retrieve specific cargo information the simulation package has to search the database every time a cargo unit arrived in the model. Since the database was large, the search times were long, which led to performance losses of the simulation models.

The final step of the preparation phase was the building of base models. The base models were built in AutoMod version 10 (Banks 2000, Stanley 2001). AutoMod offers realistic automatically constructed three-dimensional animations, which is used for validation of the model. All infrastructures in AutoMod are built true-to-scale. AutoMod allows us to study the logistic processes at different levels of aggregation to study the operational aspects of freight handling. Finally, AutoMod is a package well suited for simulating logistic systems, with built-in features for Automated Guided Vehicles (AGVs), conveyors, and AS/RS (Automated Stacking and Retrieval System). All these characteristics of AutoMod are useful for simulating airfreight-handling processes.

The base models were constructed in cooperation with individual members of the management, middle management, and shop floor personnel. The different logistic systems were modeled in the base models, including layout of the airport, cargo buildings, handling procedures, infrastructure for forklifts, transporters, conveyors, trucks, planes and AS/RS. During the construction of the base models each individual component was validated directly in order to detect errors in an early stage. The validation of these components was conducted by the (middle)-management. This led to high levels of commitment and trust in the models.

##### *Phase 2: Joint Simulation Sessions*

In the second phase two joint modeling sessions took place. The goal of the first session was to create shared understanding and validate the entire base models. The goal of second session was to generate and alternative designs of the airfreight handling processes. During the group session the base models were presented to the management. Animations and results of short model runs were used to validate the models. Two types of validation were applied; structural and replicative (Sol 1982). Face validation, a form of structural validation, was applied, by animations to the management (Law & Kelton 1991). Replicative validation was applied by comparing the results of the simulation runs to real-

world data and expectations of the management. This resulted in a shared understanding of the problem situation and high levels of trust in the simulation models. A Group Decision Room (Vreede 1995) was used to identify and rank the most important performance indicators during the first joint simulation session. These performance indicators were incorporated in the simulation models after the session.

The goal of the second session was to generate and study alternative designs of the airfreight handling processes. The management generated a large number of alternatives. The management was forced to make explicit choices regarding the operational aspects for each of the alternatives. This was needed for the immediate incorporation of the alternatives into the base models. Examples of such choices are the location of the warehouse, detailed specifications of forklifts and loading procedures. By making these choices the management gained detailed insight into the consequences of each alternative. During the session short model runs were made and rough simulation output was collected. Based upon the output the management selected the most promising alternatives. In the third phase these alternatives were studied in more detail.

#### *Phase 3: Presentation results*

The base models of the most promising alternatives were extended after the joint modeling sessions. During the joint modeling session a number of assumptions had been made on the alternatives, since there was no time to collect all real-world data. For example, the exact position of a new terminal, the exact number of AGVs and processing times.

The validation of the extended models was carried out by involving domain experts in the validation process, e.g. different members of the management and airport officials. More experiments with the simulation models were conducted in order to obtain reliable statistical data. These included different scenarios such as future expectations about cargo volumes.

The results of the experiments were compared and presented to the management. The responsibility to choose the most satisfying alternative is no part of the CBE approach and was therefore left to the management.

Using the CBE approach the management was able to choose an integral strategy for airfreight handling processes for the next decades that was supported by all members of the management.

## **5. LESSONS LEARNED**

Simulation has a number of limitations for supporting Collaborative Business Engineering approaches. Constructing simulation models during sessions based on on-line input, conducting experiments, and use results directly for further exploration was not entirely possible. Constructing,

debugging, and validating simulation models takes too much time during group sessions.

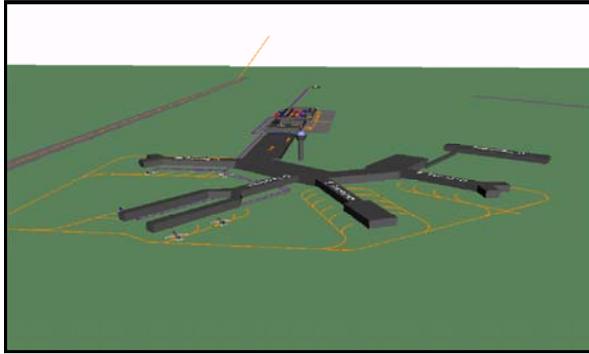
The preparation phase before the group sessions proved to be vital. During the preparation phase several base models were constructed to shorten the construction time during the group sessions. During group sessions changes to the base models were made and short experiments were conducted. After the group sessions the models were validated and more and longer experiments were conducted. The base models shortened the construction time during the group sessions, however, there are limitations to the flexibility that current simulation software offers. Not all changes could easily and quickly be incorporated in the base models. Changes to the infrastructure could easily be made. Changes to the control systems could not be incorporated during the sessions. The control systems consist of complex control algorithms that were difficult to understand for the management and difficult to validate during the sessions. For constructing base models it is important that clear problem definition and research objectives are available. This way the simulation model builders have a good idea on the type of base models they need to construct before the joint modeling sessions.

The simulation results during the group sessions formed a good starting point for discussions among the different members of the management. The simulation models forced members to make explicit choices and assumptions on the design of the airfreight handling structure. The members were forced to make their thoughts explicit and discussing such choices led to a shared understanding of the problem situation. Simulation supported the communication between members of management during these interactive sessions and smoothed and sped-up the design process. Changes to the base models were made while the management could look at the simulations models. This joint modeling sessions led to high levels of trust in simulation models and simulation results by the management.

The problem formulation and objectives changed during the project. The management came up with new ideas and wishes for the simulation models as they gained new insight during the project. This was difficult for the model builders, the base models had to be adapted frequently during the entire CBE approach.

Animations derived from the simulation models created a shared space of understanding within the management. Three-Dimensional true-to-scale animations were made of the warehouse and parts of the airport, as can be seen in Figure 2. Animations were used to validate the simulation models. The animations led to high levels of trust in the simulation models and simulation results. Animations were not used for decision-making during the group sessions. The management used

traditional figures and business graphs to make decisions on the new structure of airfreight handling.



**Figure 2. Animation model of airport**

The project of designing new structures of airfreight handling was continued by the airline company after this research was finished. Our research had two positive side effects for the airline company. First, the airline company started to make more use of simulation as a modeling tool. New simulation models were constructed of the airfreight handling processes in the next phase of the design project. The new models are based on the simulation models that were developed in the joint modeling sessions. Second, the airline company was forced to spend a lot of time on data collection. This resulted in taking a close look at all the available data of the airline company. Several separate databases were joined in one Management Information System, which improved data collection of the airfreight handling processes.

## 6. CONCLUSIONS

The CBE approach resulted in a 'richer' decision-making process. The management came up with alternatives that otherwise would not have been taken into account. The simulation models that were created during group sessions led to a shared space of understanding of the problem situation and alternatives. The CBE approach supported by simulation allowed the management to study more alternatives and to study each alternative in more detail. This provided the management the necessary insights to develop an integral strategy for airfreight handling processes for the next decades. The joint simulation sessions led to high levels of trust in the simulation models and simulation results. The alternative that was chosen in the end was supported by all members of the management.

## REFERENCES

Babeliowsky, M.N.F. (1997). *Designing interorganizational logistic networks: A simulation based interdisciplinary approach*,

Delft University of Technology, Delft, the Netherlands.

Banks, J. (2000). *Getting started with AutoMod*, AutoSimulations, Bountiful, Utah.

Banks, J. (1998). *Handbook of simulation; principles, methodology, advances, applications, and practice*, Wiley & Sons.

Davenport, T.H. (1994). Saving IT's soul: human-centered information management, *Harvard Business Review*, Vol.71, No.2.

Drucker, P.F. (1988). The coming of the new organization, *Harvard Business Review*, Vol. 66, No. 6, pp.45-53.

Dunn, W.N. (1981). *Public policy analysis; an introduction*, Prentice Hall, Englewood Cliffs, New Jersey.

Hammer, M. (1990). *Reengineering work: don't automate, obliterate*, Harvard Business Review.

Law, A.M., Kelton, W.D. (1991). *Simulation modeling and analysis*, McGraw-Hill, New York.

Maghnouji, R., Versteegt, C. (2003). AutoMod for supporting collaborative business engineering, *Brooks-PRI European Simulation Symposium*, Gent, Belgium.

Meel, J.W. van. (1994). *The dynamics of business engineering; reflections on two case studies within the Amsterdam municipal police force*, Delft University of technology, Delft, the Netherlands.

Schrage, M. (1990). *Shared minds: the new technologies of collaboration*, Random House, New York.

Senge, P.M. (1994). *The fifth discipline; the art & practice of the learning organization*, Breaaley, London.

Simon, H.A. (1969). *The sciences of the artificial*, MIT Press.

Sol, H.G. (1982). *Simulation in information systems development*, doctoral dissertation, University of Groningen, the Netherlands.

Stanley, B. (2001) The AutoMod product suite tutorial, in Peter, B.A. et al. (eds.), *Proceedings of the 2001 Winter Simulation Conference*.

Vreede, G.J. de (1995). *Facilitating Organizational Change; the participative application of dynamic modeling*, Delft University of Technology, Delft, the Netherlands.

Zeigler, B.P., Praehofer, H., Kim, T.G. (2000). *Theory of modelling and simulation: integrating discrete event and continuous complex dynamic systems*, Academic Press, 2000.

## **BIBIOGRAPHY**

*Corné Versteegt* is a researcher at Delft University of Technology, specializing in logistics and logistic control systems. Currently, he works on developing control systems for large-scale automated logistic systems. He teaches several courses on logistics and simulation.

*Sander Vermeulen* and *Eric van Duin* are students at the Faculty of Technology, Policy and Management of Delft University of Technology. Currently, they are specializing in simulation of logistic systems at airports and the role simulation can play within strategic decision-making processes.

# A FRAMEWORK FOR BUSINESS PROCESS SIMULATION: THE GRAB AND GLUE APPROACH

Tillal Eldabi  
Man Wai Lee  
Ray J. Paul

Centre for Applied Simulation Modelling  
Department of Information Systems and Computing  
Brunel University  
Uxbridge, Middx UB8 3PH, U. K.

## KEYWORDS

Discrete event simulation modelling framework, grab and glue simulation

## ABSTRACT

Simulation modelling is a powerful tool for problem understanding and problem solving. Constructing simulation models following the classical simulation modelling framework has disadvantage of being time consuming, hence making it expensive. Users can sometimes be reluctant to use simulation due to these reasons or implement simulation results. This paper proposes a new simulation approach that tackles the problem of time. For this purpose, this paper will start by reviewing a number of existing simulation modelling frameworks. From this analysis, we attempt to develop a simulation framework that deals with the question of time. The proposed simulation framework is supposed to enhance simulation results and reduce disadvantages related to cost and time.

## INTRODUCTION

Simulation modelling is defined as “an analysis and planning tool that captures real-world system variability and subsystem event interactions through time” (SDI 2001). Because of its ability to explore “What-If” questions, it is highly used as a predicting tool in forecasting the performance of the new systems under different sets of circumstances, or as a designing tool for analysing systems (Pidd 1998; Banks et al. 2001).

There is a number of simulation modelling frameworks which have already been developed by various researchers, for example Shannon (1998), Law and Kelton (2000), and Banks et al. (2001). The common problem when applying these frameworks is that they are time consuming. In this paper, we aim to find out why this problem exists and then attempt to tackle it.

This paper is structured as follows. In Section 2, we review the classical simulation modelling frameworks by Shannon (1998), Law and Kelton (2000), and Banks et al. (2001), shown in Figures 1, 2, and 3 respectively. Problems that may exist when applying these frameworks are discussed in Section 3. In Section 4, the

requirements to tackle the problem are presented, an alternative proposed framework is presented in Section 5. Section 6 is a discussion about the envisaged advantages and disadvantages of the proposed framework. Finally, Section 7 reports the conclusions.

## SIMULATION MODELLING FRAMEWORKS

Shannon (1998) stated that the process of constructing a simulation model should be problem definition; project planning; system definition; conceptual model formulation; preliminary experimental design; input data preparation; model translation; model verification and validation; final experimental design; experimentation; analysis and interpretation; and implementation and documentation (see Figure 1). Law and Kelton (2000) mentioned that a simulation model should be started by formulating the problem and planning the study; collecting data and defining the model. After that, the validity of the conceptual model will be checked. The data collection process will be restarted if there is any problem, and a new conceptual model will be defined again. If the conceptual model can pass the validity test, it will then be translated into a computer program. A pilot run will be executed, and the validity of the program model will be checked. If the testing of this model fails, the process of data collection will be restarted, otherwise a series of experiments will be designed and production runs will be executed. The output of the model will then be analysed, and finally, the model and the outputs will be presented (see Figure 2). Banks et al. (2001) mentioned that the steps followed in a simulation process are problem formulation; setting objectives and the overall project plan; constructing a conceptual model and checking the validity, and data collection at the same time. After that the model transformation and verification of that model can be tested. If the model fails at this step, the transformation of the model needs to be restarted, otherwise, the validation of the model will be checked. If it fails, the modeller needs to go back to the step for constructing the conceptual model. If the test is successful, the experimental design will be completed. Production runs and analysis can then be executed, documentation and reports can be written, and finally, the model can be implemented (see Figure 3).

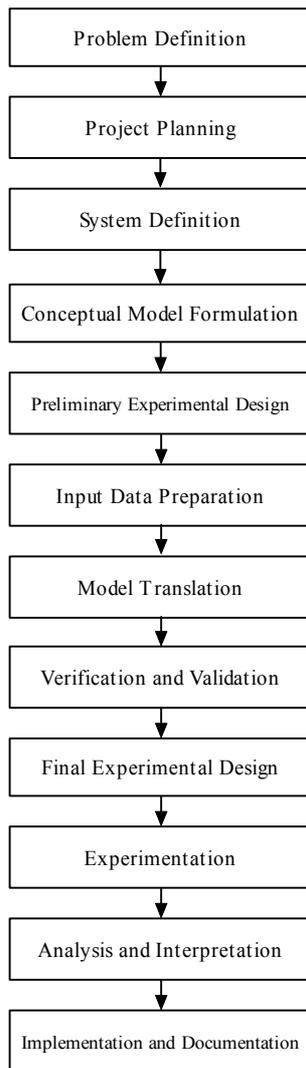


Figure 1: Simulation Modelling Framework (by Shannon, 1998)

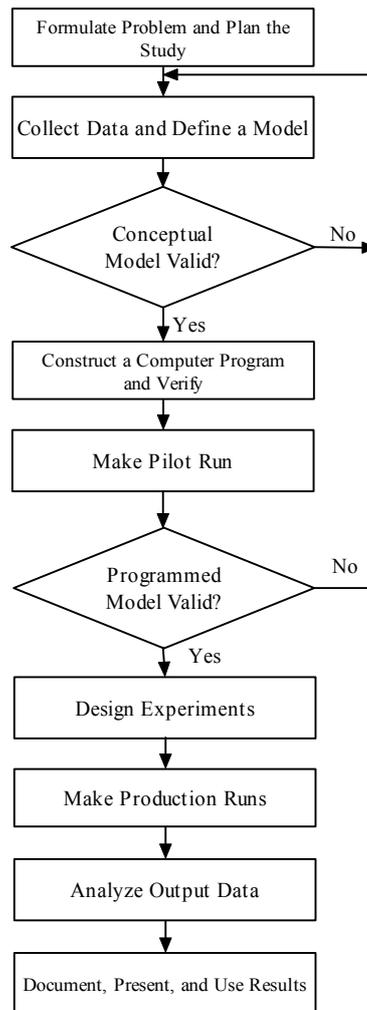


Figure 2: Simulation Modelling Framework (by Law and Kelton, 2000)

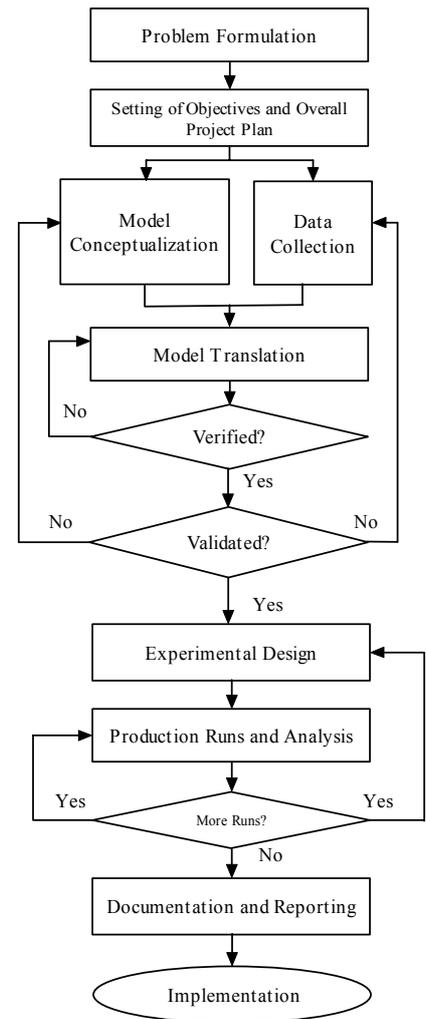


Figure 3: Simulation Modelling Framework (by Banks et al., 2001)

By comparing these three frameworks, it is obvious that the first step to build a simulation model is to formulate the problem. Formulating the problem is the most important step in the simulation modelling process (Eldabi 2000). If the user wants to find out the solution of a problem, he/she must know what the problem is. Without an understanding of the problem, the constructed model may not have the ability to solve the real problem that the problem owner has. After formulating the problem, all three researchers agreed that it is necessary to develop a project plan to ensure that the resources and support are enough to construct the model. Moreover, Shannon (1998) mentioned that it is necessary to determine the boundaries and restrictions of the system or progress as well as to investigate how it works. After that, a conceptual model and to validate that model is required. A conceptual model can be constructed either by graphics or pseudo code. Shannon (1998) mentioned that data can be collected after

deciding the required type and the required amount of data. However, Law and Kelton (2000) stated that data collection should be done immediately after the problem formulation process.

After collecting the data, all three frameworks agree that the simulation model can be built based on the conceptual model. Verification of the model is required after translating the model into computer codes or computer graphics. A pilot run of the model is necessary to validate the model. Finally, after validation, it is necessary to make the experiment design, product run and analysis, documentation and report, and the implementation of the model. Experimental design is used for making production runs. Depending on the system configuration of interest, experiment design can be the length of each run, the length of the time to warm up the model, or the number of independent simulation runs using different

random numbers. The objective of analysing the output results is either to determine the absolute value of the system configuration or to compare different system configurations under the relative condition. Finally, documentation or reports need to be provided so that the user can understand more about the model (Law and Kelton 2000).

In this section, three different simulation modelling frameworks have been studied, and the similarity and differences were investigated. In the next section, the disadvantage of applying these frameworks to construct a simulation model will be discussed.

### **SOME SIMULATION PROBLEMS**

A successful simulation model can help the problem owner to understand their problem. Applying the framework mentioned in the previous section can help to build the required model. However, constructing this model successfully is very time consuming and, hence, expensive, especially in the process of data collection and model analysis (Pidd 1998). One of the reasons is that data collected from a single person or document is insufficient for our complex world. Another reason is that people may provide inaccurate information. These two reasons show that data collection takes a long time and is hence high cost (Law and Kelton 2000).

Looking back to Figures 1, 2, and 3, Shannon (1998) stated that the process of data collection can be started after deciding the required type of data and the required amount of data, while Law and Kelton (2000) stated that this process should be done immediately after the problem formulation process. However, regardless of when we start the process of data collection, there is a possibility that the collected data will not be suitable for developing the model.

It could be argued that problem owners can discuss their problem with an expert before collecting the data. However, it is still difficult to guarantee that the recommendation provided by the expert is exactly what the problem owner wants. Thus, if the simulation modeller starts collecting the data without exactly understanding the problem, he/she may need to recollect the relevant data. Hence, the time and cost will increase significantly.

Because of this disadvantage, the usage of simulation is relatively low in the business area. According to a survey about the use of business process simulation (BPS) by different practitioners conducted by Melão and Pidd (2003), nearly 80% of respondents claimed that they did not use simulation in designing and improving business processes. There are several reasons why they refused to use simulation. In 82 replies, 71 said that they are not using simulation giving the following reasons: 34 think that is because of the nature of their current job; 17 think that it is the nature of the process/problem; 10 think that it is because of the

limitation of BPS; 7 think that it is because of the context of the organisation; and 3 think that it is because of their lack of expertise/awareness. From the limitations of BPS, 4 of them think it is too time and resource-consuming; 2 failed to find suitable software; 2 felt that simulation is too complex; 1 felt that it is too difficult to justify investment and 1 felt that it is not always appropriate.

As we can see from this survey, 4 out of 10 mentioned that applying simulation in their project was too time consuming. However, BPS projects are normally short, and the funding of the project is relatively less compared with other projects (Melão and Pidd 2003). Therefore, it is necessary to discover a method which can help simulation to apply to a BPS project.

In this section, it is justified that being time-consuming, resource-consuming, and difficult to justify investment, are the main disadvantages of classical simulation. As a result, a method to find out the requirements to overcome these disadvantages are discussed in the next section.

### **WHAT DOES THIS PROBLEM NEED?**

A fast and cheap way needs to be pursued to tackle the above problem. According to the frameworks by Shannon (1998) and Law and Kelton (2000), a model can start to be constructed after collection of the required data. Pidd (1998) mentioned that a model can be built by using computer programming code or a simulation software package. Using programming code has the advantage of high flexibility and low cost, but it is extremely time consuming whilst, construction of a model by using a package greatly reduces the time required due to the cost of packages being different, the amount of money required to be spent on package is variable.

According to a survey conducted by Hlupic (2000), over 60% of academic simulation users think that ease of modelling is necessary; while just 20% think that flexibility is important. Flexibility can be defined as easy to extend (Snowdon et al. 1998), coupled with the ability to link to external code (Hlupic 2000). When constructing models, modelling flexibility is the ability to model any system, no matter how much a system is complex or unique (Law and McComas 1997; Law and McComas 1998). Ease of use is the second criteria for selecting tools for problem solving. According to Snowdon et al. (1998), ease of use is defined as “user-friendly to multiple types of users: business process analysts, planners, and operations research experts throughout the organization”.

### **Criteria for Tackling the Problems**

Based on the results of the survey by Melão and Pidd (2003) and Hlupic (2000), a set of criteria in terms of *time* and *cost*, *ease of use* and *flexibility*, need to be

defined before finding a new way to construct a simulation model.

Time and cost are the first criteria that needs to be achieved. As discussed above, constructing simulation model is very time consuming and hence expensive. As a result, it is necessary to find a way which can construct a model in a faster time, hence, the amount of money required can be reduced.

The second criterion is that the tool used to construct a simulation model should be easy to use and high flexibility. Because most of the simulation packages are either too difficult to use or low flexibility, the problem owner may be reluctant to adopt the simulation model to help to solve their problem. As a result, it is also important that the introduced modelling method is easy to use and has high flexibility.

In this section, the disadvantages of classical simulation model have been discussed. Some criteria for constructing simulation which can reduce the problems of classical simulation model have been indicated. Based on these criteria, a new way to build simulation model will be introduced.

#### A PROPOSED SIMULATION MODELLING FRAMEWORK

One possible way to achieve the above-mentioned criteria is to assemble a model instead of constructing a model from scratch (Paul and Taylor 2002). The assembly of different existing objects together can save the time of building the objects from scratch, hence saving money. Moreover, if a tool can allow a simulation modeller to construct a model by putting different objects together, it will become easy to use and have high flexibility. When the user feels that something is not relevant, he/she can change it quickly and use some relevant objects instead. In addition, because the model is built by assembling different objects together the model can be used easily.

This paper proposes an assembly based framework based on the above idea. The proposed framework of assembly model is named the “Grab and Glue, Run, Reject and Retry” ( $G^2r^3$ ) simulation modelling approach as shown in Figure 4 (Paul, 2002). The concept is based on grabbing different objects from the web and gluing them together to form a model.

According to Yücesan et al. (2001), the web has experienced tremendous growth since the 1990s. The required objects for assembling a model can be easily grabbed from the Internet. After that, they can be glued together. If the model is satisfactory, life moves on and problem owners can continue their work. If it is unsatisfactory, the undesired parts of the model will be rejected. The grabbing process will be repeated and glued to the relevant position. This process will be iterated until the model is satisfactory.

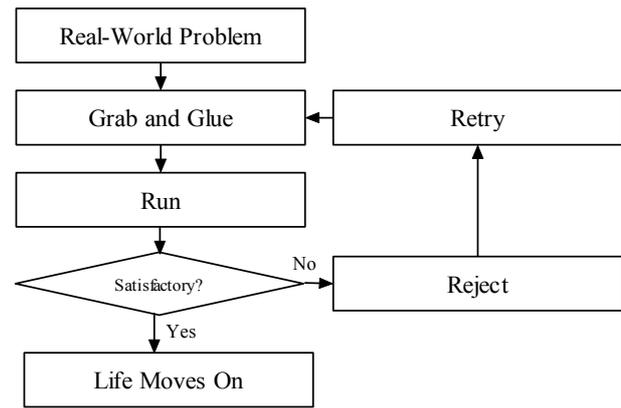


Figure 4: Framework for Grab and Glue Approach (Paul, 2002)

#### Can $G^2r^3$ Helps to Solve the Problems?

$G^2r^3$  attempts to reduce the time for model development. It does so by rethinking the concept of model development rather than merely reinventing a new tool that follows the procedures in Figures 1, 2, and 3. Since the aim of modelling is to understand the problem, data collection is deemed unnecessary, because the collected data is not usually the right data and the problem is not well understood.  $G^2r^3$  concentrates on enhancing the debate between the problem owners rather than producing mathematically precise models that bear no relation to the unknown problem.

$G^2r^3$  could also provide assistance on the second criteria, flexibility. The main principle of  $G^2r^3$  is to construct a model by assembling different objects. If the problem owners are not satisfied with the result of the model, irrelevant objects will be rejected and relevant ones will be glued easily. This process will be iterated until the problem owners are satisfied with the results from the constructed model. Models constructed by assembling techniques are easier to build and higher flexibility.

#### Object Reuse

$G^2r^3$  heavily depends on object reuse. The idea of object reuse has already been adopted for a few years in software engineering and simulation. Object reuse has the benefits of reducing software development time and costs, increasing software productivity, improving software system interoperability, reducing the number of people required to develop software, reducing the maintenance costs and producing better quality software (McClure, 1999). According to McClure (1995), from the organizational perspective, reuse can shorten development time, reduce costs and increase competitiveness; from a personnel perspective, the productivity can be increased; while from the customer perspective, a greater user satisfaction through the production of more flexible products can be achieved.

Pierre and Nouisser (2000) show that reusing graph theory algorithms, in terms of components reuse, can

increase the reliability of the software, and increase the maintainability by applying on a telecommunications network design. Bellettini et al. (2001) agree that reuse can increase product quality and decrease time-to-market, adding to the competitive edge of software development enterprises. Etzkorn et al. (2001) agree that software reuse can increase productivity, reduce costs, and improve quality. Ewing (2001) agrees that reuse can have significant effects on the cost and worth to use on simulation.

The idea of the grab and glue principle is not new, although its use in this domain is new. Mackulak et al. (1998) stated that reuse of existing generic models like simulators or software packages that contain pre-programmed models, can reduce model building time as well as increasing simulation accuracy. An automated material handling systems (AMHS) design project was used to investigate the effectiveness of simulation modelling reuse, and finally, to discover that both the model building and analysis time have been reduced from over six weeks to less than one week (Mackulak et al. 1998). Although most of the simulation packages such as Simul8 are now using the idea of grab and glue, objects are only reused within the same simulation package. However, in  $G^2r^3$ , what we are looking for is to find objects from anywhere on the Web instead of a specific simulator.

### A CRITIQUE AND LIMITATIONS OF $G^2r^3$

Because  $G^2r^3$  allows simulation modellers to build a model by assembling different objects together, it can greatly reduce the time required and, hence, the cost. Moreover, assembling a model is much easier than building a model using programming code. Thus, it is believed that this modelling method can help to reduce the disadvantages of classical simulation modelling methods. However, it could be argued that  $G^2r^3$  framework might lose the accuracy of the model. The accuracy of the model can be influenced by two factors: the accuracy of the selected objects or the accuracy of the result of the model. For the first factor, it is important to make sure that the collected objects are accurate enough before applying them to the assembly of the model. For the second problem, it is necessary to know that the purpose of the simulation model is to help the decision maker to make decisions, or to help the problem owner to gain an understanding of their problem (Paul and Taylor 2002). The grab and glue approach is only fit for the purpose of simulation modelling rather than being an elegant calculating machine. The underlying principle of  $G^2r^3$  is to enable modellers and problem owners to collaboratively develop better understanding through continuous modelling process using the  $G^2r^3$  approach. Because the design of a simulation model is not for finding out an exact solution, the interest of the numerical output becomes insignificant.

The framework of  $G^2r^3$  is much simpler than the framework in Figures 1, 2, and 3. This is because the whole concept is each problem has its own criteria. It is not possible to develop an overarching methodology that is so detailed. It may also be argued that it is difficult to motivate the simulation experts to public their simulation object or the Web. However, they will do so because they will benefit from each other by using the Web.

At the moment,  $G^2r^3$  is still at its infancy. It is difficult to conclude that which software tools or programming languages should be used for model construction. However, our aim is to use whatever is available to produce the model, object or web.

### CONCLUSIONS AND FUTURE WORKS

In this paper, different simulation modelling frameworks have been reviewed and analysed. The reasons for lack of usage of simulation models has been identified as time consuming and hence high costs are associated with the modelling process. It is always the case that models are either too complex to construct or there is not enough flexibility associated with modelling tools. In order to solve the problem,  $G^2r^3$  is introduced. However, it is not possible to conclude as yet whether constructing simulation following this approach can be successful or not, and this is due to the novelty of the approach. Different simulation software as well as programming tools will be studied to find out whether there are any existing tools suitable for assembling model by the idea of  $G^2r^3$ . After that, some realistic case studies will be conducted to analyse the performance of  $G^2r^3$ . If constructing models by assembling different objects is successful, it is believed that the disadvantages of classical simulation modelling could be significantly reduced.

### REFERENCES

- Banks, J.; J. S. Carson II; B. L. Nelson; and D. M. Nicol. 2001. *Discrete-Event System Simulation*. Prentice Hall International Series, London, 3<sup>rd</sup> ed.
- Bellettini, C.; E. Damiani; and M. G. Fugini. 2001. "Software Reuse in-the-small: Automating Group Rewarding." *Information and Software Technology* 43, 651-660
- Eldabi, T. 2000. *Simulation Modelling: Problem Understanding in Healthcare Management*. Unpublished PhD Thesis. Brunel University, London.
- Etzkorn, L. H.; W. E. Hughes Jr.; and C. G. Davis. 2001. "Automated Reusability Quality Analysis of OO Legacy Software." *Information and Software Technology* 43, 295-308
- Ewing, M. 2001. "The Economic Effects of Reusability on Distributed Simulations." In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters; J. S. Smith; D. J. Medeiros; and M. W. Rohrer. Association for Computing Machinery, New York, 812-817.
- Hlupic, V. 2000. "Simulation Software: An Operational Research Society Survey of Academic and Industrial Users." In *Proceedings of the 2000 Winter Simulation*

- Conference, ed. J. A. Joines; R. R. Barton; K. Kang; and P. A. Fishwick. Association for Computing Machinery, New York, 1676-1683
- Law, A. M. and W. D. Kelton. 2000. *Simulation Modelling and Analysis*. McGraw-Hill International Series, Singapore, 3<sup>rd</sup> ed
- Law, A. M. and M. G. McComas. 1998. "Simulation of Manufacturing Systems." In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros; E. F. Watson; J. S. Carson; and M. S. Manivannan. Association for Computing Machinery, New York, 49-52
- Law, A. M. and M. G. McComas. 1997. "Simulation of Manufacturing Systems." In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir; K. J. Healy; D. H. Withers; and B. L. Nelson. Association for Computing Machinery, New York, 86-89
- Mackulak, G. T.; F. P. Lawrence; and T. Colvin. 1998. "Effective Simulation Model Reuse: A Case Study for AMHS Modeling." In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros; E. F. Watson; J. S. Carson; and M. S. Manivannan. Association for Computing Machinery, New York, 979-984
- McClure, C. 1999. Reuse services --- Extending Software Development Methodologies to Support Reuse. Extended Intelligence, Inc. Available online via <<http://www.reusability.com/serv2.html>> [accessed April 20, 2002].
- McClure, C. 1995. Model-Driven Software Reuse: Practicing Reuse Information Engineering Style. Extended Intelligence, Inc. Available online via <<http://www.reusability.com/papers2.html>> [accessed April 20, 2002].
- Melão, N. and M. Pidd. 2003. "Use of Business Process Simulation: A Survey of Practitioners." *Journal of the Operational Research Society* 54, 2-10.
- Paul, R. J. 2002. "The Internet: An End to Classical Decision Modelling?" In *Internet Management Issues: A Global Perspective*, J. D. Haynes (Ed.). Idea Group Publishing and Information Science Publishing, Hershey, 209-219
- Paul, R. J. and S. J. E. Taylor. 2002. "What Use is Model Reuse: Is There a Crook at the End of the Rainbow?" In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan; C. H. Chen; J. L. Snowdon; and J. M. Charnes. Association for Computing Machinery, New York, 648-652. Available online via <<http://www.informs-cs.org/wsc02papers/083.pdf>> [assessed January 26, 2003].
- Pidd, M. 1998. *Computer Simulation in Management Science*. John Wiley & Sons, Chichester, 4th ed.
- Pierre, S. and N. Nouisser. 2000. "Reusing Software Components in Telecommunications Network Engineering." *Advances in Engineering Software* 31, 159-172.
- Shannon, R. E. 1998. "Introduction to the Art and Science of Simulation." In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros; E. F. Watson; J. S. Carson; and M. S. Manivannan. Association for Computing Machinery, New York, 7-14
- Simulation Dynamics, Inc., SDI 2001. Simulation dynamics. Available online via <<http://www.simulationdynamics.com/Simulation/SimulationDefined.htm>> [accessed January 17, 2002].
- Snowdon, J. L.; S. El-Taji; M. Montevecchi; E. MacNair; C. A. Callery; and S. Miller. 1998. "Avoiding the Blues for Airline Travelers." In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros; E. F. Watson;

- J. S. Carson; and M. S. Manivannan. Association for Computing Machinery, New York, 1105-1112
- Yücesan, E.; Y. C. Luo; C. H. Chen; and I. Lee. 2001. "Distributed Web-Based Simulation Experiments for Optimization." *Simulation Practice and Theory* 9, 73-90.

## AUTHOR BIOGRAPHIES

**TILLAL ELDABI** is a lecturer at the Department of Information Systems and Computing at Brunel University, UK. He received a BSc in Econometrics and Social Statistics from the University of Khartoum. He received his MSc in Simulation Modelling and his PhD from Brunel University. His research is in aspects of healthcare management and the intervention of simulation and his main research also concentrates on the economy of healthcare delivery. He is looking to exploit the means of simulation on the wider information systems management area to assist in problem understanding. Dr. Eldabi's email and web addresses are <[tillal.eldabi@brunel.ac.uk](mailto:tillal.eldabi@brunel.ac.uk)> and <[www.brunel.ac.uk/~cssrte](http://www.brunel.ac.uk/~cssrte)>, respectively.

**MAN WAI LEE** is a PhD student in the Centre for Applied Simulation Modelling and the VIVID Research Centre at the Department of Information Systems and Computing, Brunel University, U.K. He is now under the supervision of Professor Ray J. Paul and Dr. Tillal Eldabi. He received a B.Eng in Department of Mechanical Engineering in The University of Hong Kong and a M.Sc. (with distinction) in Building Services Engineering from Brunel University. His main research concentrates on fast simulation process, the new  $G^2r^3$  modelling technique. His email address is <[manwai.lee@brunel.ac.uk](mailto:manwai.lee@brunel.ac.uk)>.

**RAY J. PAUL** is a Professor of Simulation Modelling, Director of the Centre for Applied Simulation Modelling, creator of the Centre for Living Information Systems Thinking, and Dean of the Faculty of Technology and Information Systems, all at Brunel University, UK. He received a B.Sc. in Mathematics, and an M.Sc. and a Ph.D. in Operational Research from Hull University. He has published widely, in books, journals and conference papers, many in the area of simulation modelling and software development. He has acted as a consultant for a variety of United Kingdom government departments, software companies, and commercial companies in the tobacco and oil industries. He is the editor of the Springer-Verlag Practitioner book series. His research interests are in methods of automating the process of modelling, and the general applicability of such methods and their extensions to the wider arena of information systems. He is currently working on widely aspects of simulation, in particular in Web-Based Simulation and the new  $G^2r^3$  modelling technique. His email and web addresses are <[ray.paul@brunel.ac.uk](mailto:ray.paul@brunel.ac.uk)> and <[www.brunel.ac.uk/~csstrjp](http://www.brunel.ac.uk/~csstrjp)>.

# AN EXPERIENCE OF MODELING AND SIMULATION IN SUPPORT OF CMMI PROCESS

Yung-Hsin Wang  
Department of Information Management  
Tatung University  
40 Chungshan N. Rd., 3rd Sec, Taipei 104, Taiwan  
E-mail: ywang@ttu.edu.tw

Yuan-Fan Chen  
Department of Computer Science and Engineering  
Tatung University  
40 Chungshan N. Rd., 3rd Sec, Taipei 104, Taiwan  
E-mail: yuanfanc@hotmail.com

## KEYWORDS

Modeling and Simulation, Software Development, Process, CMMI.

## ABSTRACT

In this paper, we will present our experience of applying simulation technology to assist in the CMMI (Capability Model Maturity Integration) introduction through an industrial practical case. We use the SES/*workbench* software to model and simulate the proposed software development process model and compare the simulation results through different scenario plots of resource allocation, which provides managers with processes adjustment experiences and estimation basis during introduction. This demonstrated the feasibility and ability of modeling and simulation technique to support CMMI related process improvement for organizations.

## INTRODUCTION

Being essentially a non-industry that never existed forty years ago, software development and application in practice has now become an over several billion dollar business. However, with the involvement of software industry and other industries, the whole environment had become much more complicated and unpredictable. As can be seen, software has replaced hardware as the most accountable part for much of the functionality that systems provide and has become the brain of the systems we are using. Besides, with the increasing software complexity and the increasing customer demands that oriented to be "better, faster, and cheaper", software developers now are forced to seek any possible performance improvement. The industry had attempted several plots to work on this challenge to achieve their goals, in the form of various case tools, new computer languages, and more advanced and sophisticated machines. A key question is, "How can the tools, technologies and people work together in order to achieve these increasingly challenging goals?" It is noted that one potential answer to this question is through changes to the software development process or software organization. Thus, companies are addressing these issues with an emphasis on software process performance improvement and increased process maturity (Raffo 1999).

How can one be sure of the change plot that is going to be adopted and applied to be clearly correct? The answer is "You never know." Basically, the process behavior can be too complicated and the time span of the project can be too large so that potential problems couldn't be easily perceived. By the way, possible changes will require a significant amount of resources to implement and have significant implications on the firm. How can organizations gain insights into potential solutions within the processes to these problems and their likely impacts? One area of research that has attempted to address these questions and has had some success in predicting the impact of some proposed solutions is software process simulation.

In industrial practice, usually due to a shortage of time to meet market needs, processes are often changed at will without considering possible coming impacts the change could bring. This is most likely to be seen in fresh companies without much experience regarding processes, and even some mature companies without proper preservation of their past knowledge and experiences. If anyone wants to know the relationships between process changes and the possible outcome, it would need several dry runs under different conditions and examine the results to gain a better understanding of the process. Unfortunately, this is not feasible and practical since involving lots of staffs in experimental processes not producing marketable results is definitely out of the question. Hence, simulation comes to rescue for being a widely used popular method for studying complex system (Christie 1999; Kellner et al. 1999).

On the other hand, the SEI (Software Engineering Institute) Capability Maturity Model Integration (CMMI) was developed to help organizations improve their software engineering management practices (Paulk et al. 1993). It provides benchmarks evaluators can use to grade the ability of an acquisition or programming organization to produce reliable, maintainable software that meets customers' needs. Simulation and CMMI are both fundamentally process focused with common objectives to enhance process capabilities and performance. Some researches had applied simulation to the business practices to help achieve higher CMMI maturity level (Miller et al. 2002; Raffo et al. 1999).

The objective of this study is to model and simulate the software development process in order to support the introduction of CMMI in an enterprise. In the course of refining software development, the decision maker faces a huge amount of information, which may come from internal executive experiences or external user communication. It is hard to use the traditional linear programming or statistical analysis tools to handle them. With the dynamic analyzing ability, simulation technology can be used to assist in this kind of decision problem more powerfully. The enterprise can take advantage of adopting simulation results for the evaluation of new designed business processes without taking any risk of changing any physical part of a real business.

### CMMI INTRODUCTION PHASES AND EXPERIENCES

This study is based on the case of a major software development enterprise in Taiwan. Within the group there exists a lot of different business units and companies of various industries. We focus on the examination of internal activities and required process in this enterprise attempting to find out possible problems in its business process. Currently the business organization is conducting the CMMI introduction and expecting to pass level 2 in a year. Lab CMMI was established as the facilitator at the very beginning, where our study started. Firstly the role we play here was being an observer to the process improvement activities. Data was collected through attendance to training sessions, observation of requirements analysis and discussion sessions, administration of two questionnaires, inspection of relevant documents, as well as interviews with engineering development and management staffs.

CMMI was just beginning to become a springing concept and getting more attention in Taiwan recently. Now top-level managers in more and more companies are getting the idea that CMMI can be a solution to the chaotic software environment they are in. Generally speaking, companies wait too long before they attempt software process simulation and refinement. They then expect more from the process models than that may be reasonably provided. Software process simulation can help companies at early stages of process maturity. First, the graphical capabilities of software process simulation can help companies plan and define their processes and process changes such as that in CMMI Level 2 Activities. These models can be developed to a fine level of granularity and can provide a great deal of assistance in communicating and understanding a company's software development process.

The structure of the CMMI related attendants within or outside the case organization of this study contains the

SEPG (Software Engineering Process Group), three business units, one CMMI LAB and a consultant group. Basically, each business unit represents an independent company or simply one of several business units inside the group. These business units are selected from the running enterprise as a pilot to the CMMI introduction activity. The SEPG is the steering committee of the project assigned directly from high-level managers to be in charge of entire project execution and monitoring control. In addition, outside the organization an entity called CMMI LAB was established in support of the SEPG for the introduction project. It is important for being the third facilitator, the communicating bridge to the consultant group, and also the preserver of the introduction experience and knowledge for future knowledge management implementation.

Before the process started, Gap Analysis was firstly employed within the organization as a pre-evaluation of the CMMI assessment. The purpose was to find out how much difference there would be of the current processes from CMMI regulated processes. Figure 1 shows the CMMI adoption phases. A series of questions were issued based on the official CMMI articles, and a number of suitable objects were selected from the business unit including roles like senior managers, project managers and technicians who were in charge of the relevant processes.

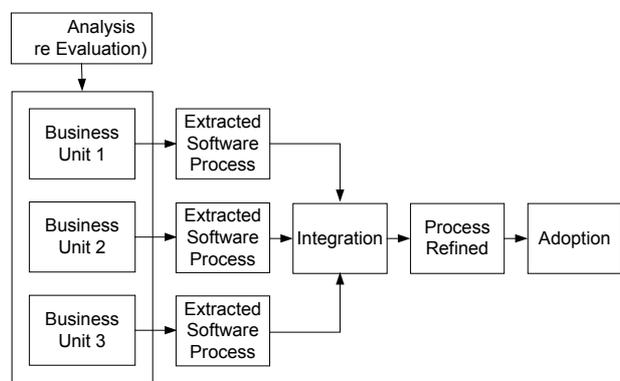


Figure 1: CMMI Process Refinement and Adoption

Before CMMI conduction, the organization needs a major integration of different processes fell in each sub unit. This is because each sub unit within the organization has their own business scope and style, and that even in the same industry, the same business practices may tend to be conducted in various ways, causing a great deal of redundancy. In order to meet the ultimate goal of process refinement and reengineering, these redundancies should be carefully eliminated. Thus, each different and unique software process must be extracted along with the relevant forms, applications and role definitions in preparation for the coming integration.

The processes extracted here would then be trimmed and nourished properly. With the unnecessary activities being cut down and required practices being replenished, integration can be successfully done. After certain adjustment and tailoring, the adoption can be then more smoothly conducted.

## SOFTWARE DEVELOPMENT PROCESS MODEL

We created the software development process model according to the real processes conducted within the company and with a reference to the typical software lifecycle models (Mead et al. 2000) as example. Then the whole model was divided into four different submodels, each representing a group of specific workflows and relevant processes representation (See Figure 2 for illustration).

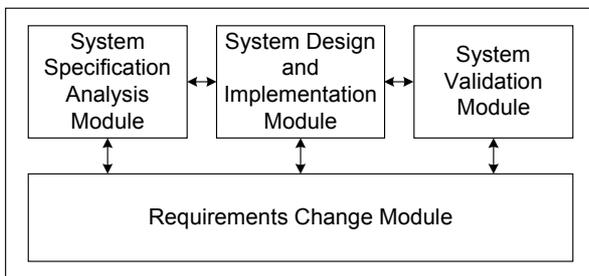


Figure 2: Proposed Model for Software Development Process

The entire model consists of four individual component modules: System Specification Analysis Module, System Design & Implementation Module, System Validation Module and Requirements Change Modules. Processes and information travel between all the sub-modules, while outside users, the clients or any other possible service objects outside the company, communicate with them through certain interfaces created. Within System Specification Analysis Module, all relevant information about the pre-requirement developing phase was gathered and analyzed, as a basis for further system design and implementation. System analysis, design and the actual coding phases were modeled through the creation of the System Design & Implementation Module, while necessary information were collected and transmitted throughout to other modules. The System Validation Module basically handles information of the final confirmation with outside users. During the lifecycle of the software development process, users' requirements change request can be received through the Requirements Change Module, and pass on to other possible sub-modules after detail analysis.

The model was simplified and established according to the software life cycle relations with process engineering areas as shown in Figure 3. As can be seen,

different projects contain different requirements, while they all have the same goal to develop the high quality product in order to meet the clients' needs within limited budget and time. Unfortunately, due to the perplexity of software development process, there is no such pattern or model suitable for all situations. Note that on CMMI level 2, there are seven KPAs (key process areas): Requirements Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management, Measurement and Analysis, Process and Product Quality Assurance and Configuration Management. In the figure we can see clearly the relationship between the software development lifecycle and the engineering areas. For example, technical solution covers two phases: Design, and Program Implementation and Testing. These issues should all be taken into consideration during the process modeling.

Due to space limitation, detailed workflow of the four sub-modules of our proposed software development lifecycle model is not depicted here. According to the problems that managers and the concerning participants stated, users' requirements change rapidly and are hard to track. Any request for a system requirement change during the later phases such as system integration and testing may cause serious effect and lead to an inability of delivery. Because a response to this request not only affect the around activities, sometimes also cause a specification redesign and lead to an unexpected and out of control situation (Lowry and Bebbington 1998).

Thus, in the modeling process, we tend to create the model in a more loosely fashion. While collecting and defining the processes, we tend to break these necessary tasks into smaller activities in order to break out the task dependency potentially lies within, which makes the model a little slightly compact and more loosely coupled. This is to eliminate the uncertainty and to increase the traceability throughout project management when receives users' requests for any change, also to make the affected area as tight and compact as possible.

According to the facts we found from the interview in the case company, the scope of the software project team varies from size to size due to the uncertainty of project scope in engagement. A small project sometimes takes only a few people while a mid-sized project takes up to a dozen of people. Normally the project team is formed not too early before the project is initiated, and team members are sometimes selected according to the needs and manager's experiences. Thus, management and the project quality can be very difficult to maintain. This is exactly why the rules of CMMI strongly address importance of the issue on the employment training. If all the employees were properly trained and skilled, once the project is initiated, managers can easily draw out needed persons and also enhance the project quality (Conwell et al. 2000; Car and Mikac 2002).

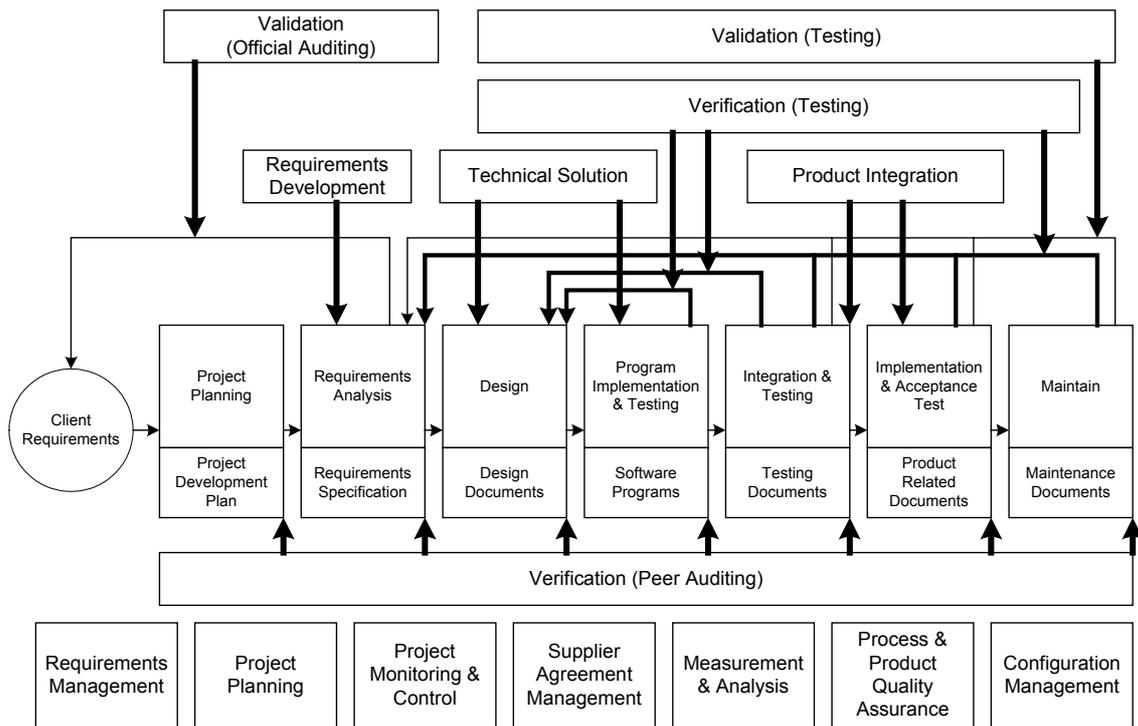


Figure 3: Software Life Cycle Relations with Process Engineering Areas

## SIMULATION MODELS & RESULTS ANALYSIS

### Software Development Lifecycle Model

The entire software development lifecycle model is designed and created in SES/*workbench* (SES, Inc. 1994) based on our discussion above for the case business enterprise. Instead of depicting the abundant submodels, we shall present some selected models in what follows.

Figure 4 shows one of the Design and Implementation Submodels. It represents the model activity processes from coding, code inspection to further function tests. In our case study, there is a testing group fully responsible for the entire code test, while in other cases we found most companies still use peer group test to ensure code product quality, which is not effective. For modeling convenience, here we simply put a node to represent it.

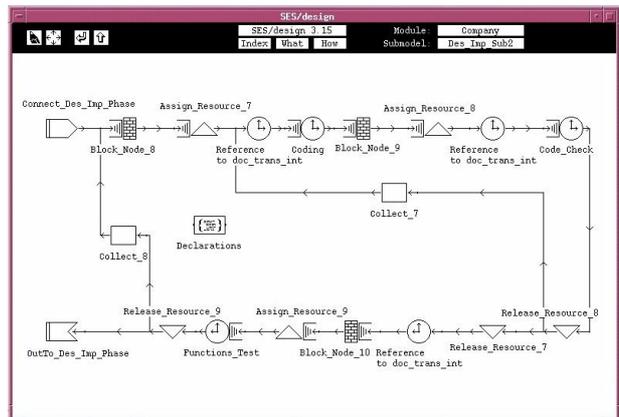


Figure 4: One of the System Design and Implementation Sub Models in SES/*Workbench*

Figure 5 shows the model of the requirements change. It handles the activity processes from requirements change receiving, analysis and validation to its final execution approval. In this study, the organizations usually use certain methods to decide which action should be taken during change analysis. Generally, changes requested from the clients are most likely of extra functions that only few programs need to be modified or created. This kind of requirements change does not affect the system drastically and normally can be solved easily. If the change effects a wide range of project extent, such as contradiction to the original specification or design, it should be taken as a new requirement and starts from the very beginning in avoidance of over time and cost.

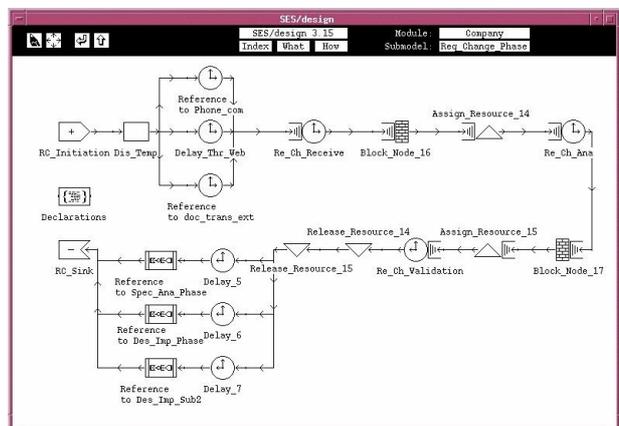


Figure 5: The Requirements Change Model

## Simulation and Analysis

In this case, we tend to find out the effects of different employment methods through some sketched scenarios with adjustment of resources allocation. A few scenarios were selected and conducted in the simulation in order to address the issues and seek the best, and relevant parameters were defined and set to distinguish the characteristics in each scenario.

In this study, we found that a small project may take only a few people while a mid-sized project may take up to a dozen of people. But mostly the size of the software development project is mid-large, which normally takes 10-20 (usually around 15) people and usually takes 3 to 6 months to complete. The project team members cover commonly from management level to downward technicians, including project manager, technical engineer, system analyst and testing technicians, etc. Some organizations may have their own project team structure according to their organizational policy or composition factors, and do a slight change. A project team is established while the project is still at the preliminary stage in contacting the clients, not long before the project initiates. The actual members of the project team were selected under considerations of their workload and technical factors. Sometimes the decision can be very hard for the managers to make.

In our simulation study, we try to find out an evaluation way and provide an analysis basis of statistical results for managers to take and see if the project plan needs to be changed. If the project were taken, would the project team be able to accomplish it efficiently? Should more technical engineers be assigned to share the unbearable workload while still keeping in budget? These are all serious issues that managers seek to answer. We use two different scenarios of resource allocation examples for the simulation and then compare their results. Table 1 and Table 2 describe the general simulation information for both scenarios. The total simulation time is set to 9140 hours, i.e., more than 4 working years.

Table 1: Simulation Settings

Name	Content
Simulation Time	9140 hours
Project Scope	28 mid-large sized projects
Project Team	10-15 people
Project Duration	3-6 months

Table 2: Resource Allocation Scenarios

Project Team Structure	Scenario 1	Scenario 2
Project Manager	1	1
System Analyst	3	2
Technical Engineer	6	8
Testing Group	3	2

In scenario 1 the parameters and the relevant settings were set as the original plot we discovered from within the company. A usual mid-large project normally takes one manager, 3 system analysts, 6 technical engineers and 3 testing technicians. Simulation results (Table 3 and Table 4) show that there exists possible working overload for the technical engineers, which managers should take into consideration to adjust team structure or the task assignment. Should the project include more people to digest unconsumed workload within budget, or does the inclusion here bring only extra expense without effective contribution to the project progress? These are dangling and unsolved if not being practiced. Accordingly, scenario 2 was developed where a system analyst and a testing technician were assigned to support technical engineers to share the unbearable overload. The results in Table 5 show that the utilization of technical engineer was cut down clearly in scenario 2. Compare Tables 4 and 6, the average cycle work hour and average rework time have about 11% decreases. This significance is due to the improvement of some possible lock knot process. These example results of simulation provide a good basis for managers to control and monitor the project progress more aggressively instead of passively waiting for possible crisis to come.

Table 3: Staff Utilization of Scenario 1

Team Member	Utilization
Project Manager	0.8900
System Analyst	0.8788
Technical Engineer	0.9661
Testing Group	0.6410

Table 4: Other Simulation Results of Scenario 1

Performance index	Value
Average Cycle Work Hour	5980.79 hrs
Average Rework Time	2318.24 hrs
Average Requirements Change Time	1783.36 hrs
Number of Requirement Change	82

Table 5: Staff Utilization of Scenario 2

Team Member	Utilization
Project Manager	0.8803
System Analyst	0.9217
Technical Engineer	0.8412
Testing Group	0.7221

Table 6: Other Simulation Results of Scenario 2

Performance index	Value
Average Cycle Work Hour	5321.32 hrs
Average Rework Time	2072.25 hrs
Average Requirements Change Time	1802.19 hrs
Number of Requirement Change	83

## CONCLUSIONS AND FUTURE WORK

### Conclusion

The process of developing software is dynamic and rapidly evolving due to the competitive pressures to deliver software products of high quality, on time and at reasonable cost. But implementing process changes along with, it is expensive, risky and unrealistic for the requirements change constantly. Thus, simulation provides a viable and inexpensive approach for organizations to gain insights into potential solutions within the process, predicting the impact of some of these proposed solutions by assessing and reducing this risk through the quantitative analysis of process performance. The software process model we proposed here was constructed from the current processes conducted within the case enterprise with the aid of the SES/*workbench* simulation tool. Before conduction, the standard process was extracted from several companies. Simulation here is used as an estimation approach for better project planning. From the research results, simulation proves to be an effective and valuable way to support both initial planning and re-planning activities. In initial planning, resource need and cost estimates must be established. And if the result doesn't go well, pitfalls should be notified and erased in the re-planning activities. In the way a better solution is made, CMMI can be introduced more smoothly with less resistance and difficulty, thus saving its cost.

### Future Work

A business organization should keep on pursuing higher software capability maturity not only for the underlying benefits it can bring, but also for being able to deal with the rapidly changing era. According to the CMMI definition, any level-5 organization should have the ability of self-adjusting to the optimal state. Thus, we will keep on the work on using simulation to help achieve higher software development maturity, and smoothen the CMMI introduction. In our research, since we are now focusing on an about-to-be level 2 company, only fairly experience-based data associated with costs and schedules are likely to be available. Some information was collected from the widespread survey or interview. But these issues can be improved while moving onto higher level of CMMI. On level 3, a library of processes that can be reused should be created and maintained. On pursuit of level 3, information like requirements changes, design-defects and code-defects can also be acquired as an input to the simulation model. Therefore, detailed performance measures like software or service quality and development time to adding or changing functionality can all be elaborately calculated and estimated. In the practical case, the question of how can we conduct CMMI standard without sacrificing past efforts is a serious issue to the organization.

## REFERENCES

- Car, Z. and B. Mikac. 2002. "A method for modeling and evaluating software maintenance process performances," In *Proceedings of Sixth European Conference on Software Maintenance and Reengineering*, 15-23.
- Christie, A.M. 1999. "Simulation in Support of CMM-based Process Improvement." *Journal of Systems and Software* 46, No.2/3 (Apr), 107-112.
- Conwell, C.L.; R. Enright; and M.A. Stutzman. 2000. "Capability Maturity Models support of modeling and simulation verification, validation, and accreditation." In *Proceedings of Winter Simulation Conference*, Vol.1, 819-828.
- Kellner, M.I.; R.J. Madachy; and D.M. Raffo. 1999. "Software Process Simulation Modeling: Why? What? How." *Journal of Systems and Software* 46, No.2/3 (Apr), 91-105.
- Lowry, G.R. and P.R. Bebbington. 1998. "Towards enterprise transition to object technologies: a CMM-based methodology." In *Proceedings of International Conference on Software Engineering: Education and Practice*, 78-87.
- Mead, N.R.; R. Ellison; R.C. Linger; H.F. Lipson; and J. McHugh. 2000. "Life-Cycle Models for Survivable Systems." In *Proceedings of the Third Information Survivability Workshop* (Boston, MA, Oct. 24-26).
- Miller, M.J.; D.M. Ferrin; and F. Pulgar-Vidal. 2002. "Achieving Higher Levels of CMMI Maturity Using Simulation." In *Proceedings of the Winter Simulation Conference*, Vol.2, 1473-1478.
- Paulk, M.C.; C.V. Weber; S.M. Garcia; M.B. Chrissis; and M. Bush. 1993. "Key Practices of the Capability Maturity Model for Software, Version 1.1." Technical Report SEI-93-TR-25. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (Feb).
- Raffo, D.M. 1999. "Getting the Benefits from Software Process Simulation." In *Proceedings of 1999 Conference on Software Engineering and Knowledge Engineering* (Kaiserlautern, Germany, June).
- Raffo, D.M.; J.V. Vandeville; and R.H. Martin. 1999. "Software Process Simulation to Achieve Higher CMM Levels." *Journal of Systems and Software* 46, No.2/3 (Apr), 163-172.
- SES, Inc. 1994. *SES/workbench Creating Models*, Release 3.1. Scientific and Engineering Software, Inc., Austin, TX (Feb).

## AUTHOR BIOGRAPHIES

**YUNG-HSIN WANG** is an associate professor of the Department of Information Management at Tatung University, Taipei, Taiwan. He received his MS and PhD degree in Electrical and Computer Engineering from the University of Arizona in 1987 and 1992, respectively. His research interests include modeling and simulation methodology, web-based simulation, simulation-based intelligent systems, decision support systems, and e-Business related area.

**YUAN-FAN CHEN** received his MS degree from the Department of Computer Science and Engineering at Tatung University in August 2003. His research interests are in the area of modeling, simulation, and CMMI related software process improvement.

# VALUING REAL OPTIONS WITH SIMULATION SOFTWARE

Eyler Coates  
University of Southern Mississippi  
P.O. Box 5137  
Hattiesburg, Mississippi 39406 USA  
E-mail: Eyler.Coates@usm.edu

Jon Juneau  
Fulcrum and Lever  
920 Sioux Lane  
Hattiesburg, MS 39402 USA  
Email: jonjuneau@fulcrum-lever.com

Rita Schweickert Endt  
University of Southern Mississippi  
P.O. Box 850  
Gautier, Mississippi 39553 USA  
Email: Rita.Endt@usm.edu

## KEYWORDS

Real options, financial risk analysis.

## ABSTRACT

Traditional engineering economic analysis concerns itself mainly with deterministic inputs, even though deterministic data seldom occur in business. Additionally, traditional net present value methods used to evaluate potential projects make no allowance for flexibility by management and assume a static environment. Practitioners often assume that risk analysis and real options are too complicated to include in their analyses. The simplified approach is less accurate and managers often intuitively adjust the results to reflect their understanding of the risks and potential rewards. This paper demonstrates the ease that engineering economic analysis with risk analysis and real options can be valued by simulation software that is readily available to owners of personal computers. This novel approach to modeling real options may also encourage more sophisticated and realistic engineering economic analysis.

## INTRODUCTION

In order to deal with the variability issues of real business projects, risk analysis is necessary. Unfortunately, the risk analysis approach is one aspect of economic analyses that is commonly ignored during project evaluations. Ristroph points out that errors in estimates of cash flows are the rule rather than the exception. He also states that the primary question involving most cash flows is not whether they will be correct, but rather by how much will they be incorrect (Ristroph 2000). Including risk analyses in engineering economy solutions is an important step for acquiring more information to make better management decisions. Ho and Pike report that "proponents of risk analysis argue that increased risk information improves management's understanding of the nature of risks, helps identify the major threats to project profitability and reduces forecasting errors." (Ho and Pike 1998) They also report "the risk analysis approach provides useful insights into the project, improves decision quality and increases decision confidence."

There are several approaches used to handle economic risk. One approach is scenario analysis. However, as

Park has pointed out, the worst-case and best-case scenarios are not easy to interpret and do not provide probabilities of occurrence of those possibilities. He continues to point out that these scenarios normally do not provide additional information such as the probability of losing money on a project or the probability of other possibilities (Park 2002). However, multi-scenario analysis, used in conjunction with probability descriptions of input variables, forecasts the relative impact and interaction of all the uncertain factors simultaneously. The result of the study is a distribution of the desired answer. The distribution of the possible outcomes in itself provides a clear picture of the range and variability of the possible outcomes of the project. The use of simulation software to conduct multi-scenario analyses makes the approach practical. As Ristroph states, simulation provides insight into the variability of a project's potential performance and hence its risk, so that an informed, albeit subjective, decision can be made (Ristroph 2000).

Given that there is imperfect information when we make a financial decision in the planning and beginning of a project, we need to determine what happens when the information becomes certain as the project progresses. Is there an option to bail out of a bad deal or is the organization committed to finishing a project? Are there other alternatives at different stages of the project that would yield better results? If the organization can "cut its losses" and perhaps recoup some losses by recovering some of its investment in the event that a project goes sour, then the organization has some flexibility in regards to the execution of the project. In other words, there is an "option" for alternative project direction once the project is underway. Evaluating these 'options' ahead of project execution or during project execution gives management more information to make better decisions about a project.

## INTRODUCTION TO REAL OPTIONS

By definition, "a real option is the right, but not the obligation, to take an action (e.g., deferring, expanding, contracting, or abandoning) at a predetermined cost called the exercise price, for a predetermined period of time – the life of the option." (Copeland and Antikarov 2001) The ability to adjust a project gives an organization real options. Recently, the evaluation of

real options in project analysis includes the methods developed for evaluating financial options (Herath and Park 1999) (Herath and Park 2000) (Nembhard et al. 2000). For example, deciding to prematurely abandon a project can be modeled in a similar manner as modeling a put option on a stock.

A financial stock option is a contract between the option writer, who sells the option, and the option owner, who buys the option. If the option owner decides to exercise the option, the option writer must execute the transaction specified in the contract. Since the option contract can be bought and sold, they have a market value. The factors that determine the value of an option include the price of the publicly traded stock and the specifics of the option.

To introduce the ideas of stock options, assume that XYZ Corporation has publicly traded stock and stock options. An option on XYZ Corporation's stock would specify an exercise price (or strike price) and an expiration date. The exercise price is the cost to the option owner to exercise the contract. The expiration date is the last date that the owner can exercise the option.

Calls and puts are the two types of stock options. A call option on XYZ stock might have an exercise price of \$100 and an expiration date of July 20. This option would allow the owner to buy XYZ stock from the option writer for 100 dollars per share on or before July 20. (Note: European options can only be exercised on the expiration date and American options can be exercised on or before the expiration date. We assume here that the option is an American option.) The owner of an American put option on XYZ stock with an exercise price of 100 and an expiration date of July 20 can sell XYZ stock for 100 dollars to the option writer on or before July 20. Stock options have an intrinsic value. For example, if the market price of the stock grew to 125 dollars, the call option owner can buy the stock for 100 dollars (option exercise price) instead of the market price of 125 dollars and the intrinsic value of the option would be 25 dollars. If the stock of XYZ corporation stock has a market price less than 100 dollars, the option to buy XYZ stock for 100 dollars becomes worthless. Likewise, the put option owner can sell the stock to the option writer for \$100 dollars. Thus, if the market price of XYZ stock is 85 dollars, the owner could buy the stock for 85 dollars and then exercise the option to receive 100 dollars making an instant profit of 15 dollars. Therefore, the intrinsic value of this put option would be 15 dollars. If the market price of XYZ stock rises above 100 dollars, then the put option will become worthless.

Because of the possibility for large profits, a great deal of work has been done to learn methods to value options. On the expiration date, an option's value will

equal its intrinsic value, with no uncertainty. The European option is the easiest to evaluate because it can only be exercised on the expiration date. The value of the European option before expiration involves adjustments for both the time value of money and the uncertainty in the final stock price. An analysis on a binomial lattice is used when the situation is modeled with a discrete time periods and discrete changes in price. The Black-Scholes formula for the European call option is the continuum limit, with the time periods and the price changes becoming very small, of a binomial lattice. Brealy and Meyers hint at how to adjust the value of the European call to determine the value of other options (Brealy and Myers 2000). The value for real options can be approximated when the methods of evaluating financial options are applied to the choices an organization makes in a project.

### **SIMPLE REAL OPTIONS**

If an organization can show that their choices, or the real options, are similar to financial options, then they can use the financial option evaluation methods to evaluate the real options. In this paper, the authors examine an organization's real option to prematurely abandon a project when expectations are not being met.

The option to abandon is often modeled like a put option. When the organization decides to abandon a project, they might sell their capital equipment investment. The cash flow from this sale could be modeled as the exercise price of the put option. (Note: If expectations are better than expected, the organization might also have the real option to expand their production capability. This could be analyzed like a call option with the exercise price being the investment amount required for the expansion.) Only the option to abandon, the put option, will be modeled in this paper.

The next section presents the financial option approach to modeling the premature abandonment of a project. Then the simulation approach is demonstrated for the same project. Since real options are often more complicated than the European call option, simulation is a powerful tool for evaluating these real options.

### **EXAMPLE PROBLEM**

The problem presented in this paper is a three-phase project with incoming cash flows in each phase and an option of abandoning the project at the end of the first two phases. This type of problem is becoming more prevalent in today's business because projects are becoming more and more complex, longer in duration and more costly. Most companies cannot afford to financially back a long-term project without receiving some feedback (cash flows) at several stages throughout the duration of the project.

Below is the description of the problem presented in this paper:

*A company has a project that requires a \$30,000 investment. The \$30,000 investment is the best estimate but there is an uncertainty about the amount. The company's minimum attractive rate of return (MARR) can be calculated based on the capital structure of the firm using the weighted-average cost of capital (Park 2002). The degree of project variability is captured by the probability distributions of the cash flows and the timing. So, the discount rate can be set to the risk-free rate. For our example, we are using  $i = 15\%$ . Each phase of the project could have a duration between 1 and 2 years and incoming cash flows for each phase is estimated to be \$15,000. From past experience, the company knows that the estimated cash flows are somewhat variable and that they are also correlated to each other (a typical time series). That is, if the first cash flow is low, then the remaining cash flows are also likely but not necessarily low. Could the company lose money doing this project?*

The traditional engineering economy analysis solution to this problem, using end-of-year cash flow conventions, yields a risky answer because it is not designed to handle variable progress payments and the other uncertain parts of the problem. The method presented in this paper provides an easy way to solve this more complex problem using simulation software. Simulation software can estimate the distribution of the possible net present values (NPV) for the project. Arena, a readily available PC based simulation software, was used to solve this problem. The following steps should be used when using simulation software to solve this type of economic decision problem:

1. Determine the input variables
2. Create a flowchart for a single scenario
3. Identify the entity for the simulation
4. Determine the appropriate attributes and global variables that are required for the final result
5. Design and enter the network diagram to generate multiple scenarios according to the flowchart
6. Verify the model
7. Run the simulation
8. Analyze the output

Each of the above steps is described in the following paragraphs. The given data for this problem are the distributions for the initial investment, discounting rate, the cash flows at the completion of each phase, the duration of each phase of the project and the late penalty for missing the deadline.

### **Step 1 - Determine Input Variables**

The simulation solution requires that we input the initial investment, the minimum attractive rate of return (i.e., the discounting rate), the amounts of the individual

progress payments, the timing of these progress payments, the project deadline and the late penalty for missing the deadline. The values of all of these inputs can be deterministic or stochastic according to the requirements of the problem. For this problem, the initial investment by the company to begin the project is assumed to be normally distributed with a mean of \$30,000 and a standard deviation of \$333, denoted in this paper as NORM(30000, 333). Given that plus or minus three standard deviations in a normal distribution tend to encompass 99.73% of the possible data points, essentially it is assumed that the investment will be likely be around \$30,000 but could even be as low as \$29,000 and as high as \$31,000. Likewise, the cash flow at the completion of the first phase is NORM(15000, 2000) dollars and the duration of each phase of the project is estimated to be NORM(1.5, 0.2) years. The time duration is given in years to match the time units of the discount rate, which is 15% per year. Note that unique cash flows and time durations could easily be assumed for each phase. The details on how to determine what distributions to use for a variable are covered in Coates and Endt (Coates and Endt 2000).

Note that the cash flows for the last 2 phases follow a first-order auto-correlated series based upon the first cash flow. For this example we will assume an 80% correlation coefficient although this figure should be based on past experience. So, this means that if the first cash flow is low, then the remaining cash flows are likely (but not necessarily) to be low also and the management might decide to abandon the project.

### **Step 2 – Create Flowchart**

A good intermediate step to solving the overall problem is to generate a flowchart for calculating an individual scenario. The flowchart for this problem is shown in Figure 1. Once the basic steps to generate a single scenario are worked out, the transition to a simulation flowchart is rather simple.

### **Step 3 – Identify Entity**

Each entity, in the simulation run, will represent a single replication (scenario) of the net present value calculation. After an entity is created, the stochastic variables are selected via sampling from the appropriate distribution.

### **Step 4 – Determine Attributes and Global Variables**

Several variables are needed in order to calculate the net present value for an individual scenario. These would be assigned as attributes since they would be unique to a scenario (entity). From the flowchart in Figure 1, these attributes would be Interest, Investment, Timing, Payment and NPV. The auto-correlation of 80% could also have been a global variable.

### Step 5 – Design And Enter The Network Diagram

There is almost a one-to-one relationship between the Arena simulation network diagram and the flowchart given in Figure 1. The simulation network is rather elegant but well within the grasp of the average student.

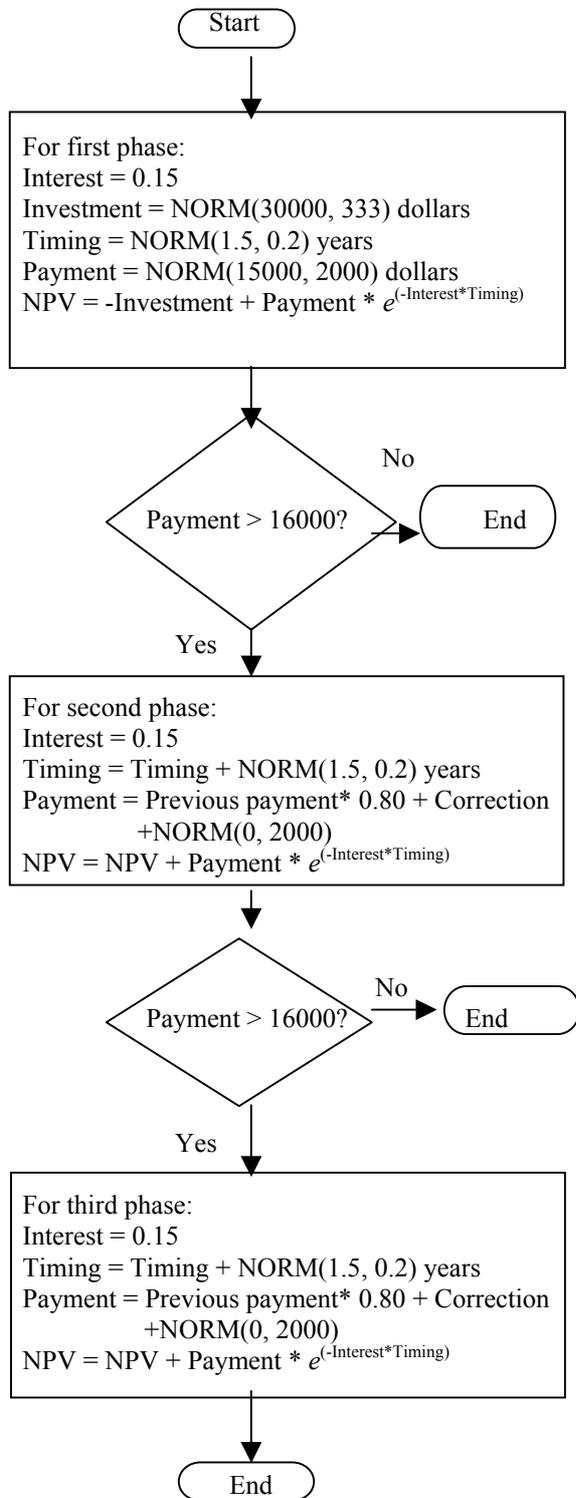


Figure 1: Individual Scenario Flowchart

The Actions Modules perform the iterative calculations to generate the value of the NPV attribute. The first phase Action Module calculates the NPV up to the point of the first progress payment. The details for that module are given in Figure 3. The details of the other action modules for the remaining progress payments are very similar to Figure 3.

The Arrive module, labeled New Scenario, allows for the creation of entities. Each entity represents one scenario. Detail of the Arrive Module is given in Figure 2.

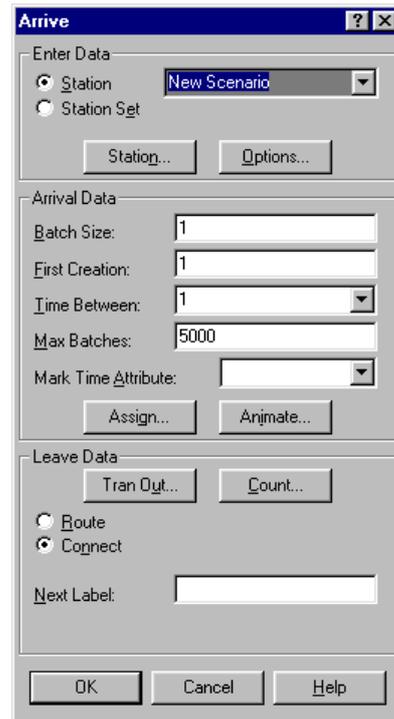


Figure 2: Detail of Arrive Module

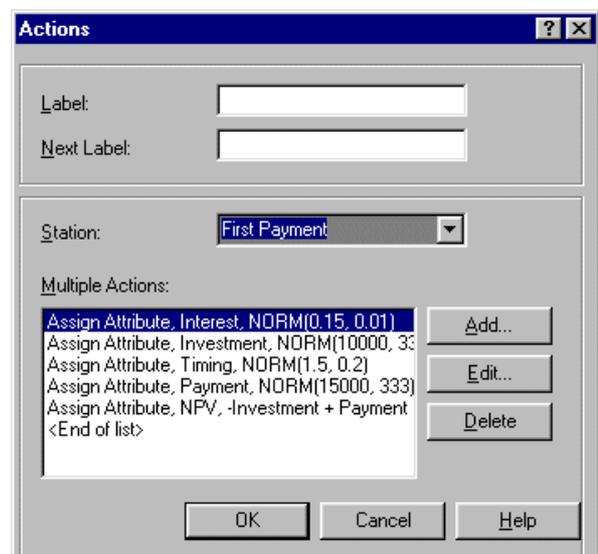


Figure 3: Detail of Action Module for First Phase

The Depart module allows the attribute NPV to be tallied before the entity is destroyed. The Arena network requires two additional modules. The Simulate module is required only to enter the title of the report and the name of the analyst. The Statistics module is required to tell the simulation software to save the output data, namely NPV, into a file for later analysis. There are several methods for stopping the simulation. The method chosen was to limit the number of created entities in the Arrive module. For this example, the number of entities (scenarios) was 5000. See Figure 2.

### Step 6 – Verify The Model

There are a couple of ways to verify this simulation model. A relatively easy method is to replace all the stochastic variables with constants. Then run the simulation program (only one scenario is needed). Compare the results with a manual calculation. For this problem, the simulation, using all deterministic inputs such as the means of the distributions yields and allowing no flexibility to terminate the project, yields a point estimate of -\$821. This is verified by the manual NPV calculation, which yields a point estimate of a negative \$820.49.

### Step 7 – Run The Simulation

The NPV associated with the particular scenario (entity) is calculated from the cash flows from all 3 phases of the project. This NPV is stored as an attribute of the entity. Before the entity leaves the system, the NPV is collected as a statistic and tabulated by the simulation software. Five thousand scenarios are run by the simulation program. It only takes 0.07 minutes to run the simulation. Thus, the number of scenarios can be increased greatly with little strain on a personal computer's resources. For this paper, the simulation was run with no option to abandon and then again with the option to abandon.

### Step 8 – Analyze The Output

An excerpt of the text output of the simulation program is given in Figure 4. When there is no option to abandon, then the project has a greater variability. One can see that the net present value of the 5000 scenarios ranges from \$-15,040 to \$14477. Also, the average NPV for the project with no option is \$-860.60. By comparison, the net present value of the 5000 scenarios of the project with an option to abandon ranges from \$-2822 to \$16761 and the average NPV is \$3067.1

For additional comparisons, histograms of the outputs, when there is no option to abandon and when there is an option to abandon are given in Figures 5 and 6. Based on the histograms, the analyst can determine any prediction (tolerance) interval. For example, on the project with no option, the middle 90% of the

observations fall between \$-8000 and \$6000. By contrast, when there is an option to abandon, it appears that the middle 90% of the observations fall between \$0 and \$7,000. Therefore, the option to abandon has removed most of the risk. Incidentally, if we had used deterministic estimates of the input values and no allowance for flexibility to cancel the project (a real option), the NPV calculation would yield a point estimate of -\$860.49 with no indication whatsoever of the probable range of the risk associated with this project.

### CONCLUSION

This paper demonstrated that commonly available simulation software that runs on personal computers can:

- easily be used in engineering economy analysis to explain the risk associated with a project
- give the present worth distribution of a project when there is uncertainty about the future timing of cash flows, particularly in those projects that have progress payments which involves the uncertainty of the amount of the cash flows, an uncertain interest rate as well as uncertain timing.
- easily incorporate a discontinuous function such as a decision to cancel a poor performing project (a real option)
- handle problems with a number of replications that that can not be handled by spreadsheet software

```

ARENA Simulation Results
Project: 3-phase project with option to
abandon
Replication ended at time      : 5000.0

Identifier      Avg      Half Width      Obs.
-----
NPV w/option    3067.1    40.368          5000
NPV no option   -860.603  131.16          5000

Simulation run time: 0.07 minutes.
Simulation run complete.

```

Figure 4: Text Output of the Simulation Program

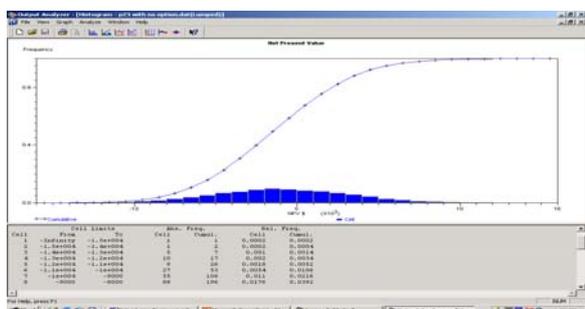


Figure 5: Project NPV with No Option to Abandon

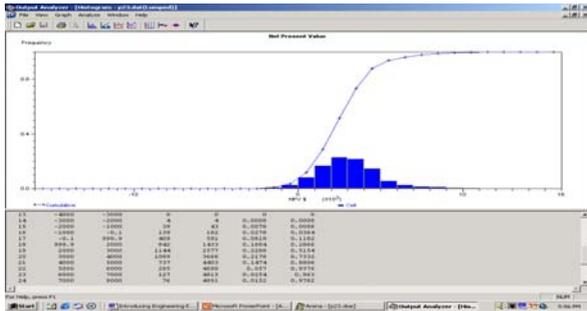


Figure 6: Project NPV with Option to Abandon

## REFERENCES

- Brealy, R.A. and Myers, S.C., [2000], *Principles of Corporate Finance*, 6th Edition, Irwin McGraw-Hill.
- Coates, E.R., and Endt, R.L., Making Engineering Economy 'Real' With Simulation Software, *Proceedings of the 2000 ASEE Annual Conference* – St. Louis, MO, June, 2000.
- Copeland, T.E., and Antikarov, V., [2001], *Real Options, A Practitioner's Guide*, TEXERE Publishing Limited, New York, New York.
- Goyal, A.K., J.M. Tien, and P.A. Voss, [1997], Integrating Uncertainty Considerations In Learning Engineering Economy, *The Engineering Economist*, 42(3), 249-257.
- Herath, S.B.H., and C.S. Park, [1999], Economic Analysis of R&D Projects: An Options Approach, *The Engineering Economist*, 44(1), 1-35.
- Herath, S.B.H., and C.S. Park, [2000], Exploiting Uncertainty: Investment Opportunities as Real Options: A New Way of Thinking in Engineering Economics, *The Engineering Economist*, 45(1), 1-36.
- Ho, S.S.M., and R.H. Pike, [1998], Organizational Characteristics Influencing The Use Of Risk Analysis In Strategic Capital Investments, *The Engineering Economist*, 43(3), 247-268.
- Nembhard, H.B., Shi, L., and C.S. Park, [2000], Real Option Models for Managing Manufacturing System Changes in the New Economy, *The Engineering Economist*, 45(3), 232-258.
- Park, C.S., [2002], *Contemporary Engineering Economics*, 3rd Edition, Prentice Hall, Upper Saddle River, New Jersey.
- Ristroph, J.H., Economic Simulations for Risk Analysis, *Proceedings of the 2000 ASEE Annual Conference* – St. Louis, MO, June, 2000.

## AUTHOR BIOGRAPHIES



**EYLER COATES** was born in Baton Rouge, Louisiana, USA. He is an Associate Professor of Engineering Technology at The University of Southern Mississippi in Hattiesburg. He has 12 years of industrial work experience with manufacturers performing industrial engineering functions. He received a B.S. degree in Industrial Engineering (1979), a M.S. degree in Engineering Science (1996), and a Ph.D. in Engineering Science (1998) all from Louisiana State University in Baton

Rouge. His Web-page can be found at <http://www.set.usm.edu/bcoates>.



**JON JUNEAU** was born in Dallas Texas, USA. He is a management consultant in Hattiesburg, Mississippi. He has 13 years electric utility experience in nuclear fuel management. He has a B.S. degree in Nuclear Engineering (1978), a M.S. degree in Nuclear Engineering (1980) and a M.S. degree in Mathematics (1981) from Texas A&M University. He has a Ph.D. in Engineering Science (1996) from Louisiana State University. He is a licensed professional engineer and certified in production and inventory management. His email is [JonJuneau@fulcrum-lever.com](mailto:JonJuneau@fulcrum-lever.com)



**RITA SCHWEICKERT ENDT** was born in Detroit, Michigan, USA. She is an Assistant Professor of Engineering Technology at The University of Southern Mississippi in Gautier. She has 27 years of work experience with several large corporations and the US Navy. Her research interests lie in life-cycle costing. She has a B.S. degree in Industrial Engineering (1977), and an M.S. degree in Industrial Engineering (1979) from Wayne State University. She is currently completing her Ph.D. in Industrial Engineering at Mississippi State University. Her email is [Rita.Endt@usm.edu](mailto:Rita.Endt@usm.edu)

# Discrete Event Simulation in a Virtual Enterprise Environment: a Case Study Reflection of Multiple Developers

Joacim Johnsson  
Björn Johansson

Department of Product and Production Development  
Chalmers University of Technology  
SE-412 96, Gothenburg, Sweden

E-mail: joacim.johnsson@me.chalmers.se, bjorn.johansson@me.chalmers.se

## KEYWORDS

Discrete Event Simulation, Virtual Enterprise, Multiple Developers.

## ABSTRACT

Today, making decisions in a distributed production system, like a Virtual Enterprise [VE], have few support tools. This paper discusses how Discrete Event Simulation [DES] can be used in a VE environment. A DES model was built involving multiple developers who were individually responsible for one part of the model. In this case study, incremental development methodology has been used together with a methodology for conducting DES projects. The paper presents reflections from the developers and gives recommendations for applying DES on a VE. The most important reflections were to formulate a well defined goal for the project as a whole, to start with integration as early as possible, and to have tangible goals.

## INTRODUCTION

In recent years there has been a growing attention concerning competitiveness for small and medium-sized enterprises [SME]. More efforts are being exerted on planning and managing flexible and efficient organisation and collaboration network between companies (Porter 1998). Collaborating in production networks has, by researchers, been given many names. Names such as Dynamic Network (Miles and Snow 1986), Intelligent Enterprise (Quinn 1992), Virtual Organisation (Venkantraman and Henderson 1998), and Agile Virtual Enterprise (Goranson 1999) are some examples describing similar concepts.

The Virtual Enterprise[VE] is based on a temporary cooperation with the capability of fulfilling a specific customer order. The order, often classified as a short business opportunity, is divided between the collaborating companies, each adding their specific competence to the divided value chain. A structure with different actors connected to the collaborating companies has been developed for an efficient handling of activities within the VE. Analysis of the VEs lifecycle has also been investigated, resulting in a Virtual Enterprise Reference Architecture and Methodology [VERAM]. Reid et al (1996) and GLOBALMAN21 (2002) describes the phases a VE

goes through in its lifecycle. However, there is still a lack in the VE research concerning supporting tools for the decision making. In this paper the use of DES in a VE environment will be discussed. A case study in this environment, done by 25 MSc students at Chalmers University of Technology, will be presented and reflections from this will be discussed.

## THE VIRTUAL ENTERPRISE CONCEPT

The research area of Virtual Enterprise is a growing and multidisciplinary one, which requires precise definition of the concept. Afsarmanesh et al (1997) describe VE in the following way:

*“VE is a network of enterprises that constitute a temporary alliance, in order to share their costs, skills, and resources, in supporting the necessary activities towards the exploitation of fast-changing opportunities for product or service requests and competitiveness in a global market.”*

Companies, which share common interests, form a network which works as a platform towards customers. When the network receives an opportunity, the firms that are suited to manufacture the order are joined together in a VE where the whole production system is distributed between the firms, Figure 1. The network have to be efficient in forming different VE constellations to handle different orders.

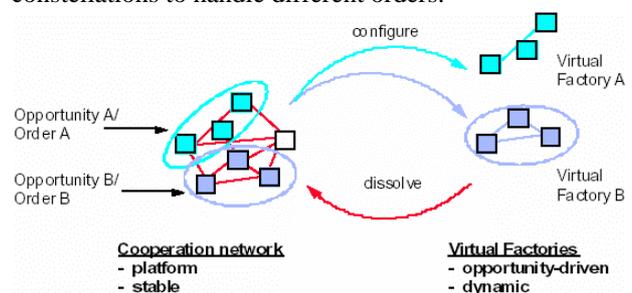


Figure 1: Establishment and management of cooperation (Virtuelle Fabrik 2002)

Advantages in collaborating in a Virtual Enterprise are, for example increased response speed to the customer, flexibility, reduced costs, and the fact that each company can focus on its core competence (Afsarmanesh et al 1997; Goranson 1999). Flexibility

can be achieved by using overcapacity within the collaborating companies. Therefore no, or very small, investments have to be made. A manufacturer that has all operations in-house has better control over the complete production system. But is not as good as a VE to manage the flexibility in volume and mix that comes with handling many different orders.

## DECISION MAKING IN VIRTUAL ENTERPRISES

A VE is characterised by its flexibility of synchronising companies in different combinations due to different orders, figure 1. In this environment there is a need for a supporting tool that helps decision maker in the offering, planning and execution phase. These decisions will secure the delivery time, give an opportunity to plan production and affect the priority within companies.

During a VEs lifecycle there are a number of decisions that have to be made. In early phases possible constellations for managing the production is investigated. This is done by dividing the product into operations and tasks that can be summed up in a value chain. The VE constellation is based on the collaborated companies capacities and capability of conducting the operations. The next phase is to schedule these operations so that lead times can be secured both within the VE and toward customer. In this phase obligations from the VE should be synchronised with each of the companies obligation in it self. Since companies have different obligations towards other customers and maybe also to other VEs, a priority list have to be followed. In a VE the relations have to be trustful to achieve a long-time benefit.

VE works in an environment where the configuration between companies can change fast, depending on the customer order size. This complexity makes it hard to optimise the distributed production system, however to be competitive there is still a need to increase the efficiency of the production system as a whole.

## DES IN VIRTUAL ENTERPRISE

Building a DES model of a VE, involves both the attended companies and the value chain that the product is divided into, figure 2. The approach used in this section is from a start-up view of using DES to develop a complete new VE.

While building a VE simulation model there is a need for each company to have a model representing their production which can be added to the VE model. Since Virtual Enterprise is a conflicting and changing environment, updating models plays a vital role.

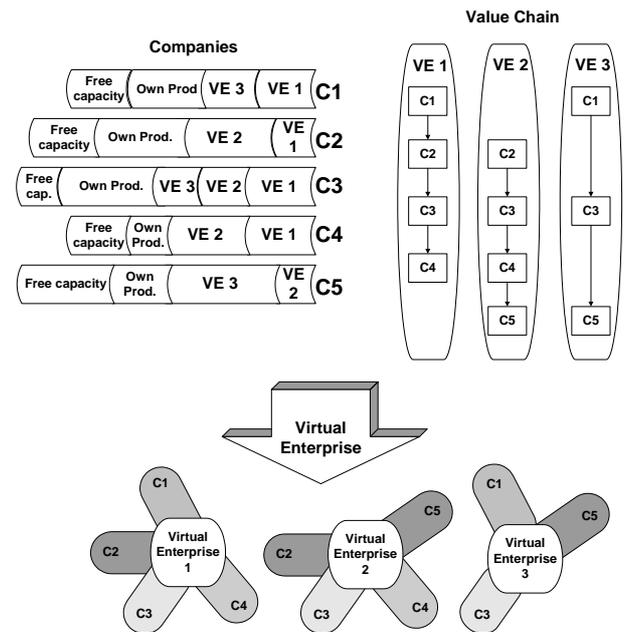


Figure 2: Configuration of collaborating companies in different VEs

Being successful with this kind of complex project is a difficult task due to the many different developers that are involved during the project, and the fact that the process of building a simulation model is classified as an art, according to Pegden et al (1995):

*“Model building requires special training. It is an art that is learned over time and through experience. Furthermore, if two models of the same system are constructed by two competent individuals, they may have similarities, but it is highly unlikely that they will be the same.”*

On the other hand, if two or more developers would build sub-models representing one large system together, how would this model consisting of many connected systems act? To answer this question a case study was prepared that would involve multiple developers, where each developer was building a production unit of the complete model.

## Multiple Developers

Building a simulation model is a time consuming task and by adding multiple developers working with the model it is expected that the development lead-time would be reduced. But with more developers involved the communication complexity is increasing as figure 2 shows. The developers need to build their models with the same level of abstraction, enabling the aim of the model to be met. A name convention is also needed, which secure transparency in communication between sub-models. Since the aim is to develop a model representing the VE, open communication can reduce sub-optimisation which reduces the lead time in the model building phase. The complexity of figure 3 calls for a methodology to support model building with multiple developers. Primarily to make sure that

conflicts, where for example entities using the same information, is avoided, or secured.

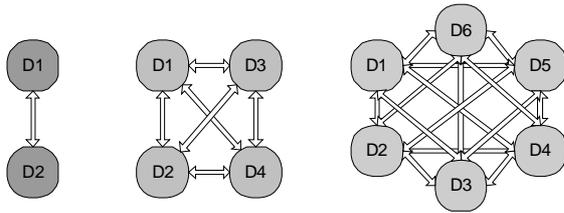


Figure 3: The increasing number of communication channels with multiple developers (Babich 1986)

There is a need to work in a structured manner to be able to verify the complexity represented by the VE. Incremental development is one way of structuring the development of a complete model.

### Incremental Development

One common recommendation when building a simulation model is to Keep-It-Small-and-Simple, KISS, which the approach of incremental development also support (Randell et al 1999). Figure 4 shows how incremental development can be used in a VE environment. In three stages that have the same goal, represented by the background arrow, but differ in the level of abstraction; VE, Factory and Machine level. In the first stage, VE level, the model is built up by “black boxes” representing different areas within the company. Building this model is swift due to the low level of detail, which also makes it poor in supporting the decision making in an overview analysis of the system.

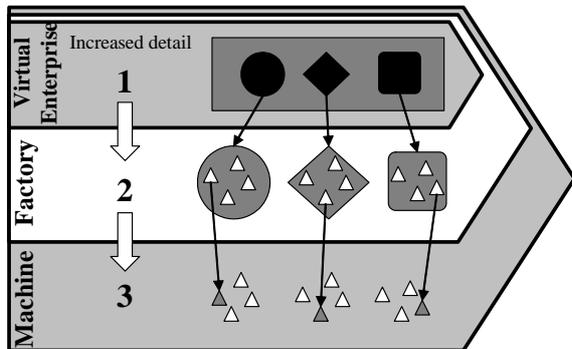


Figure 4: Incremental development of a simulation model

In the next stage, Factory level, the boxes open up and more details are added to them. With this model the level of detail is increased and with this also the possibilities for analysis. In the last stage, Machine level, the abstraction is down on the lowest level where details for each small entity are added.

Modularisation as shown in Figure 4 reduces the complexity of the model building phase of a simulation study. Although complexity between different entities within the model itself is reduced by modularisation, the complexity grows by the number of entities in the model.

In a Virtual Enterprise all developers in Figure 3 represent a company on the highest level of abstraction. The goal for working efficiently with DES in a Virtual Enterprise environment would be to have every company’s model worked as a “plug-and-play” model that could be added to the VEs distributed production system. To do this the level of abstraction has to be decomposed and well defined both within and between each node in the VE.

### Success factors

Since DES is a tool that have been used for many years, factors for succeeding in this projects are well documented. Still we find cases studies that fail in the most fundamental areas (Johnsson and Johansson, 2003). Following are some of the most well known success factors from literature (Banks et al 2001; Shannon 1998; Williams 1996).

- Have clearly defined goals.
- Have adequate resource available to successfully complete the project on time.
- Have management’s support and have it known to those who supplies us with information and data.
- Assure that the necessary skills required available for the duration of the project.
- Be sure that there are adequate communication channels to the sponsor and end users.
- Have a clear understanding with the sponsor and end users as to the scope and goal of the project as well as schedules.
- Have good documentation of all planning and modelling efforts.

### CASE STUDY: USING MULTIPLE DEVELOPER IN DES

#### Introduction

The case study, carried out by students as a part of a project course at Chalmers University of Technology, was conducted on a company with traditional manufacturing, including both machining and assembly. The company’s main interest in this analysis was to find opportunities to reduce the lead time and improve the accuracy in the delivery process. Recently the company changed their manufacturing layout from focusing on manufacturing process to a more flow orientation one and made lead time a prioritised area. One big problem for the company is the customised product variance which is effecting the production planning. To cope with the huge variance the company produces batches of all the different components and stores them in modular assembly units to reduce the lead time from order to delivery.

#### Method

The students were divided into eight groups. Seven of these groups were responsible for one production unit

each and the last group was responsible for the complete factory model. Figure 5 shows how operations in the production flow were divided into seven areas, each representing a company in the VE environment.

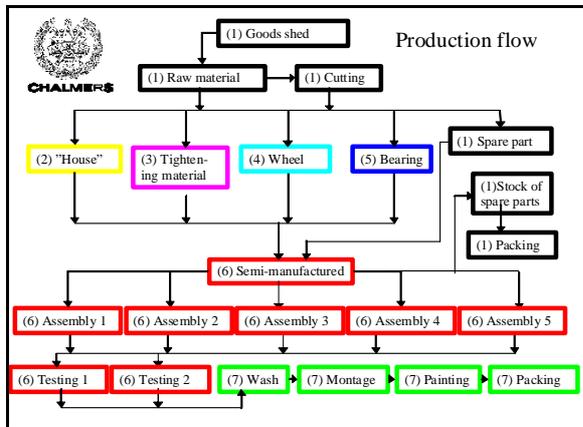


Figure 5: The production flow divided into seven production units

The project was coordinated in a activity organisation with eight groups and nine activities see Table 1. The overall group where responsible for handling the project plan and maintain the communication channels between the seven production unit groups.

Table 1: Project organisation map

	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6	GROUP 7	OVERALL GROUP
<b>Problem definition</b>								
<b>Data collection</b>								
<b>Modelling</b>								
<b>Integration</b>								
<b>Verification</b>								
<b>Visualisation</b>								
<b>Validation</b>								
<b>Optimisation</b>								
<b>Planning and controlling</b>								

The activities in Table 1 were modified from the methodology concerning steps in a simulation study described by Banks et al (2001).

From each group, at least one student was attached to each activity within the project map. Communication channels were opened both within the group and between all responsible students in each activity.

## Results

Reflections from the activities were made during the project, which are highlighted and summarised below.

**Problem definition:** It is important to not only look upon the group's definition of the problem but also on the problem definition as a whole. This will ensure that all the parts (i.e. sub-models) of the model have the same level of detail and can work together to a wide extent. It is also important to understand what parameters to measure in the beginning, enabling preparation for the future merging of the sub-models.

**Data Collection:** Collecting data has in many projects been the missing link to a successful simulation project. The data is often in the wrong format or not updated and sometimes even estimated. If the right data can not be gathered initially, the model has to be checked carefully in the validation phase and also be analysed for its sensitivity to other input data.

**Modelling:** In this phase it is very important to have good communication between the developers to reduce sub-optimisation, and to solve complex modelling tasks. Building up a network between the developers will also improve creativity within the problem solving phase, where solutions can be used by all developers. Modelling takes time, but there were groups that got more attached to their model and adding more details than needed to solve the problem.

**Verification:** Since the main model was updated by each group at short intervals the verification was not expressed as a problem.

**Visualisation:** Machines and factory layout were given 3D-life in a low level of detail. This phase was done mainly to make the personnel at the company understand the model more easily, and find acceptance for the model as a replication of their company.

**Validation:** This phase of the project was left out to be conducted by the company after a takeover of the model.

**Optimisation:** This phase was also left out from the project course.

**Planning and controlling:** Many of the groups addressed problems with integrating the sub-models as the most difficult one. This phase should have been started earlier in the project. All groups had an own project room situated close to each other in a laboratory which enabled good communication environment. Even so, there was still a lack of communication between certain groups. The closeness between the groups that the project room created made all groups work nearly the same amount of time in the project, which was appreciated by the students.

## CASE STUDY REFLECTIONS TO VE ENVIRONMENT

The case study was conducted at one single company, which in other words is not a VE. In a VE the production system is more complex, with a less hierarchical structure, compared to a single company. Still the case study was handled in a manner that would suit a structure of a simulation study on a VE environment with production units representing small companies. Each student group represents one company in the VE, focusing on one area, unaware of the

companies hierarchical structure. By using incremental development method the multiple developers worked in a way which made it representative to a VE environment. From the case study only reflections are made, due to the complexity that VE models have which will make conclusions hard to draw. The DES software used in the case had not, in this version, a support for handling hierarchical models which would have simplified the case.

## DISCUSSION AND CONCLUSION

This paper gives rise to reflections of important issues when carrying out a Discrete Event Simulation project with multiple developers. The reflections after the case study were:

- It is of vital importance to start with integration as early as possible
- Many meetings with divided goals and result follow-up are needed
- Clear project goals for the model building as a whole are important
- Communication in general has to be extremely frequent to achieve success

Communication was not seen as a major problem which, according to the groups, was due to the nearness of the groups during the project, and the fact that the group members already know each other. However there were some problems with communication when it came to knowledge about the model and the real system, which caused misunderstandings and rework. Also sub-optimisations indicates that communication between the groups in each activity was not sufficient. Additionally communication is of importance due to the distance between companies in a real VE, which was not reflected in this study. Transferring knowledge between developers is a research area in it self (Nonaka, 1994) and communication is the way of sharing this. To secure communication during the project a clear organisation and methodology has to be applied, like the activity organisation used in the case study.

The groups' found it hard to schedule the activities during the project which indicates the importance of making all participants understanding and accepting the objectives of the model. Once again communication is an important issue.

Building simulation models is an art and when multiple developers are involved there is an increasing need of starting in a small and simple way (Keep It Small and Simple). This will simplify the verification, which in complex models like VE environment is very hard. Integration is the hardest phase in these kinds of simulations and therefore it is important to let the model grow as a whole from the beginning of the project, and not as islands.

Working efficiently with DES in a VE environment, compared to a traditional company environment, has to

be more focused on synchronising and standardising the model building. This would make it possible to "plug and play" models together when a new VE constellation is to be analysed.

A Virtual Enterprise that competes with larger companies does not have the same amount of supporting tools for making improvements within the production process. When working with Discrete Event Simulation as a decision tool, a Virtual Enterprise has larger potential in improving the competitiveness of the divided production system.

## Future of VE simulation

Building simulation models that is valid to the real production is hard work not least because the lack of accurate data (Johnsson and Johansson, 2003). This lack may soon occur to be a problem of the past due to the increased number of computers that are attached to machines nowadays (Taylor et al, 2002). Taylor et al also states that the tremendous potential that distributed simulation has can fall on the willingness to share sensitive/critical data.

Simulation software have become more and more object oriented with an hierarchical thinking that supports a easier handling of VE models. Models will not be merged in the future, which will make naming convention an issue of the past. Pegden predict that future software could handle pre-built models or model component that can be plugged together to form a model of our system (Diamond et al, 2002). This future looks bright.

## REFERENCE

- Afsarmanesh H., Garita C., Hertzberger L.O., Santos-Silva V., 1997, Management of distributed information in virtual enterprises – the prodnet approach, 4th International Conference on Concurrent Enterprising
- Babich, W. A. 1986. Software Configuration Management. Addison-Wesley, Reading, Massachusetts
- Banks, Jerry, John S. Carson, II, Barry Nelson, and David M. Nicol. 2001. Discrete-Event System Simulation, 3rd edition. Englewood Cliffs, New Jersey: Prentice-Hall
- Diamond R. et al., 2002, The Current and Future Status of Simulation Software (Panel), Proceedings of the 2002 Winter Simulation Conference, San Diego, USA.
- GLOBEMAN21, 2002, Global Manufacturing in the 21st Century, Final Report, [http://www.ims.org/projects/project\\_info/globeman.html](http://www.ims.org/projects/project_info/globeman.html), Accessed Dec
- Goranson, H T., 1999, The Agile Virtual Enterprise. London: Quorum Books
- Miles R., Snow C., 1986, Organizations: New Concepts for New Forms. CMR, Vol. 28, Nr.3,S.62-73.
- Nonaka, I., 1994, A Dynamic Theory of Organizational Knowledge Creation, In Organization Scene, Vol 5 ,No 1, February
- Porter, M. 1998 Clusters and the new economics of competition. Harvard Business Review, November-December, 77- 90.
- Pegden C D., Shannon R E, and Sadowski R P., 1995, "Introduction to simulation using SIMAN", McGraw-Hill
- Quinn, J.B., 1992, The Intelligent Enterprise. New York: Free Press

- Randell, L. Holst, L. Bolmsjö, G., 1999, Incremental system development of large discrete-event simulation models, Winter Simulation Conference, Phoenix, Arizona, USA
- Reid R. L., Rogers K. J., Johnson M. and Liles D., 1996, "Engineering the Virtual Enterprise." 5th Industrial Engineering Research Conference. Minneapolis, MN, . pp. 485-490.
- Shannon R. E., 1998, Introduction to the art and science of simulation, Winter Simulation Conference, Washington ,USA
- Venkatraman N, Henderson C., 1998, "Real Strategies for Virtual Organising."Sloan Management Review 40.1:33-8
- Virtuelle Fabrik Bodensee regionen, 2002, <http://www.virtuelle-fabrik.org>, Accessed June
- Williams, Edward J. 1996. Making Simulation a Corporate Norm. In Proceedings of the 1996 Summer Computer Simulation Conference, eds. V. Wayne Ingalls, Joseph Cynamon, and Annie V. Saylor, 627-632.

## AUTHOR BIOGRAPHIES



**JOACIM JOHANSSON** was born in Stenungsund, Sweden 1973. He attended Luleå University of Technology at Mechanical Engineering, where he obtained his M.Sc. degree in 1999. He is now working as a PhD student in the field of Discrete Event Simulation and Low Volume Production, at the Department of Product and Production Development, Chalmers University of Technology, Sweden. His email address is: [<Joacim.Johnsson@me.chalmers.se>](mailto:Joacim.Johnsson@me.chalmers.se)



**BJÖRN JOHANSSON** was born in Gothenburg, Sweden, 1975. He attended Chalmers University of Technology at Mechanical Engineering, where he obtained his M.Sc. degree in Production Engineering in 2000, and his Licentiate Degree in 2002. He is now working as a PhD student in the field of Discrete Event Simulation and Productivity Improvements in Manufacturing Systems at the Department of Product and Production Development, Chalmers University of Technology, Sweden. His email address is [<Bjorn.Johansson@me.chalmers.se>](mailto:Bjorn.Johansson@me.chalmers.se).

# USING WEB SERVICES AND ARTIFICIAL INTELLIGENCE TECHNIQUES TO DEVELOP SIMULATION MODELS OF BUSINESS NETWORKS

Tamrat W. Teweldeberhan  
Alexander Verbraeck

Systems Engineering Department  
Faculty of Technology, Policy and Management  
Delft University of Technology  
Jaffalaan 5  
Delft, 2628BX, THE NETHERLANDS

## ABSTRACT

The research described in this paper introduces a new approach for developing simulation models using web services and artificial intelligence (AI) techniques so as to address the current challenges in modeling business processes within business networks. The approach is based on the Web service business process specification standards for identification and extraction of structural elements of business processes and also it is based on AI neural network techniques for modeling behavioral aspects of each of activities within the business process. The paper first formulates the current challenges in modeling business processes of business networks. Next, some related works on this area are discussed. After that, the approach for developing business process simulation models of business networks is explained step wise. The paper concludes by stating the expected benefits from using this new methodology.

## INTRODUCTION

Just as in evolution theory, where the fittest organisms are those that are the most adaptable, organizations that are capable of adapting quickly to the rapidly changing environment are the most likely to survive and thrive. In the current era, organizations are using interorganizational relationships i.e. business networks, as a means of adapting and increasing their performance level (Hengst and Sol, 2001) because the current era is characterized by high competition, high information flow, high demand for timeliness and accurateness of information, change of business needs, and change of customer needs. According to Hengst and Sol (2001) trends such as deregulation of markets, increase use of information and communication technology, outsourcing, and globalization, indicate this fact. By using these business networks, organizations bring together their core competencies to create “best of all” products or services.

Because there is so much interdependence between the nodes (organizations) within the business network, each organization needs to have a sufficient insight into the operational business processes of partner organizations to make appropriate assessments to increase performance level. Simulation models of the operational business processes of organizations in business networks looks like a very good candidate to give sufficient insight to decision makers in an organization when assessing the functioning of the other organizations (see Figure 1).

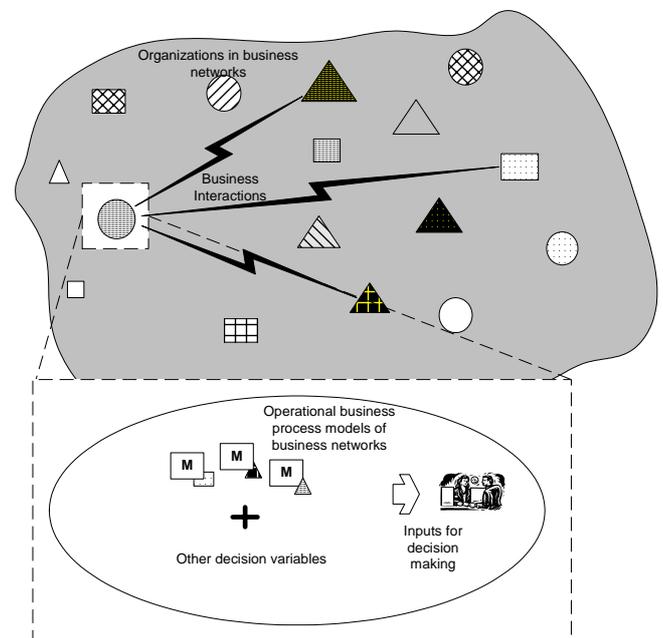


Figure 1: Business processes of organizations

Simulation models of these business processes can give insight into the dynamic behavior of organizational business processes (Aguilar, M. et al, 1999). However, getting or creating such models is challenging given the fact that organizations in the business network are auto-

mous and not transparent. Even though collaboration in model development between the organizations in the network is another option for constructing interorganizational simulation model, it usually takes time to build interorganizational operational model of all the partners involved in business networks, each organization might be interested in different aspect of the interorganizational interaction, and finally, the organizations might not be willing to share such information with other organizations.

The emerging web services technologies promise to facilitate collaboration among business partners by helping potential partners to find one another and integrate their business processes. It also enables organizations to specify and model their processes as services. Web services allow organizations to describe and discover processes (Benatallah et al, 2003). Once the processes are describe using web service standards, the abstract (public) operational business processes becomes visible to other organizations.

Artificial intelligence techniques on the other hand provide ways to model behavior of a system, e.g. business process, that appear as a black box. Neural networks are one type of artificial intelligence techniques that are used for modeling behavior by monitoring the input and output parameters of the system. In theory, these neural network models can also be used to describe/model the dynamic behavior of the processes of organizations in a business networks when the inputs and outputs are known.

This paper shows an approach for developing simulation models of operational process models of business networks by using and integrating the Web Service business process choreography standard and a neural network. The rest of the paper is organized as follows: Section 2 presents some of the related research in this area, Section 3 describe the proposed approach, Section 5 states the implementation planned, and finally Section 6 discusses the expected findings.

## RELATED RESEARCH

There are several research works available on the use of simulation and neural networks for different purpose. Panayiotou et al. (2000) used a neural network model in order to overcome one of the limitation of simulation: low computational speed. In their approach, neural network model is used as a “surrogate” model of the original system capturing the relationships between input and output, but computationally more efficient than simulation.

Kilmer et al (1994) used supervised neural networks as a metamodeling technique for discrete-event, stochastic simulation. In their study, neural network estimates are used to form confidence intervals, which are compared for coverage to those formed directly by simulation.

Chandrasekaran et al (2002) investigated the synergy between web service technology and simulation. According to this research, simulation can be used to understand and design composition of web service in order to get better performance. By using the JSIM package, they indicated that users can do “what-if” scenarios and visualize the Web process in action before enactment.

In the area of operational business processes integration, there are several works that make use of the current web service technologies. There are several standards proposed by different industry sector such as RosettaNet<sup>1</sup>, ebXML<sup>2</sup> and OpenXchange<sup>3</sup>. Wombacher et al (2003), for instance, work in automating matchmaking of business processes of potentially compatible partners by using web service standard WSCL and UDDI business process tModels. Benatallah et al (2003) proposed a way to conceptually model web service conversation among different partner organizations. By building on top of Web standards they proposed a framework for defining extensible conversation meta-model to enable description of generic abstraction such as temporal constraints and implications of service conversation.

## APPROACH

The approach that we are going to explain uses two technologies: the web services business process choreography standards and the artificial intelligence technique neural networks. The reason for using the web services business process choreography standards is that they describes structural elements of business processes of organizations by specifying the possible sequence of interaction with the web service of organizations. In order to construct a simulation model of business processes, structural elements (building blocks) of the process are not enough. Behavior of each of the blocks must be modeled as well (e.g. service time distributions). However, modeling behavior of each blocks of organizational business processes is not as straightforward because the internal mechanism of these processes is not transparent to other organizations in the business network. Only the input and output of these processes can be monitored. For this purpose, we use neural networks for modeling the behavior of the building blocks. We are aware there are several AI techniques besides neural networks for modeling black box systems and we chose to use neural networks for the reason that it is possible to develop models from data without an initial model and it can handle noise and irregularities in inputs. Before we describe the approach, we will discuss something about both technologies.

---

<sup>1</sup> <http://www.rosettaNet.org>

<sup>2</sup> <http://www.ebXML.org>

<sup>3</sup> <http://www.openXchange.org>

## Web services

Web services are being considered as a contemporary paradigm for the development of distributed, Internet-based and platform-agnostic business applications (Yang et al, 2000). The main appeal to the business community is the fact that they can facilitate interaction between complex, heterogeneous and highly distributed, enterprise information systems using standards for virtually all interoperability aspects (Heuvel et al, 2003). Web services are composed of different technologies and standards: SOAP (a standard based on XML that is used for exchanging messages within the web service infrastructure), WSDL (an XML based description for services), UDDI (a standard for publishing and discovering services), and some other standards for choreography and aggregation. A visualization of the technologies is presented in Figure 2.

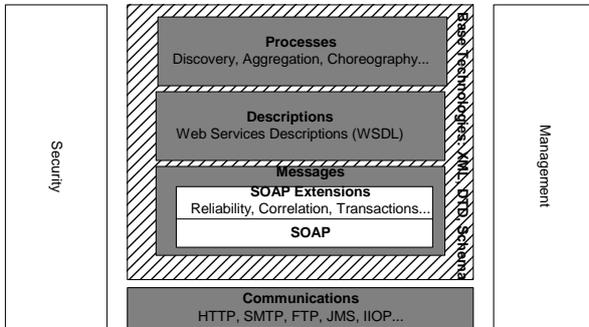


Figure 2: Web service technologies

For enabling and designing business transactions, there are several proposed web services standards: Business Process Execution Language for Web Services - BPEL, Business Process Modeling Language - BPML, Web Service Choreography Interface - WSCI, Web Service Conversation Language - WSCL. These choreographic languages are used for modeling, representing, and describing internal and external parts of their operational business processes so that they can be recognized as web services. The public part of the business processes are used for collaboration with external business partners while the internal part is used for modeling and understanding those parts of the business processes that are private (confidential).

Conceptually, services can be comprised of three levels: messages, abstract processes, and execution processes. The message level describes the message that are exchanged and the syntax involved. WSDL and EDIFACT are examples. Abstract processes describe the sequences in which messages are exchanged. Example of standards used at this level are WSCL, the abstract part of BPEL, cpXML and ebXML BPSS. Execution processes are used to implement abstract processes for execution within an organization. The executable part of BPEL is an example of a standard addressing this level. Abstract processes are considered as public business processes that can be revealed to other organizations while execu-

tion processes are usually internal and confidential. In the ebXML framework, for instance, potential business partners register their profiles (including their abstract business processes) within public registries (see Figure 3).

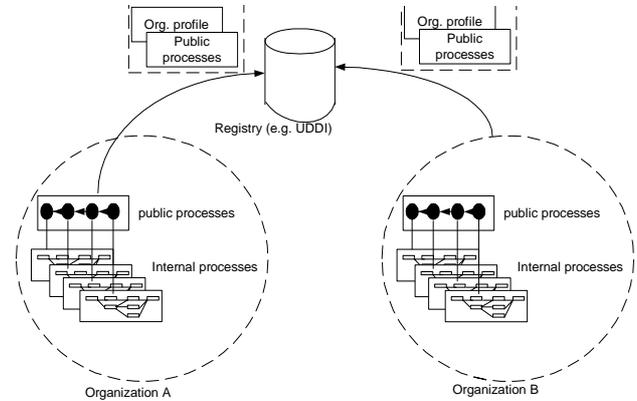


Figure 3: Public and internal processes of organizations

In this paper, the focus is on abstract processes that describe the sequence of message exchange between trading partners. For illustration purpose, we chose to use the WSCL standard for specifying and modeling abstract processes.

## Neural networks

Neural networks are one of a group of artificial intelligence technologies for data analysis. They differ from other classical analysis techniques by learning about the chosen subject from input and output data, rather than being programmed in a traditional sense. Neural networks discover patterns by detecting patterns and relationships in the data, learning from relationships and adapting to change. Compared to other data mining methods, neural networks are powerful for behavior modeling because they can successfully deal with non-linearities. Noise and irregularities in input can be handled by the neural network models and the models can also be updated easily and quickly. The two main advantages of neural network techniques are processing speed and facilitation of model development in situations where there is limited theory describing the cause-effect relationship between the independent and dependent variables (Chao et al, 1994, Flood, 1990). The picture below shows a type of neural network model called Multilayered Feedforward Neural Network (MFNN) (see Figure 4).

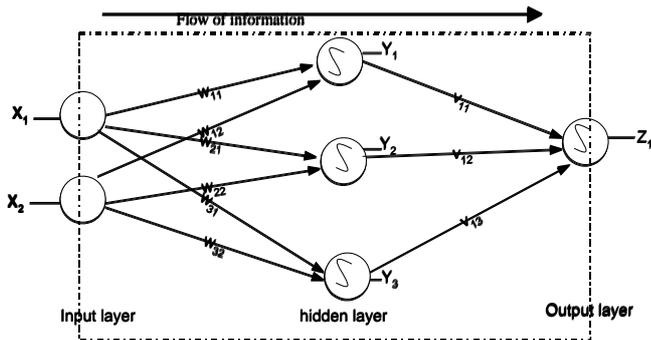


Figure 4: Multilayered Feedforward neural network

These type of models can mimic the behavior of a black box system by extracting relationships between the input data and output data. Once the model is trained, it responds with the same behavior as the black box system given a set of inputs.

There are different classes of neural network models, depending on the problem type (e.g. prediction problem, classification problem, or clustering problem), the structure of the model and the model building algorithms.

There are several other artificial intelligence techniques: Expert systems/Knowledge based system, Genetics algorithm, and Intelligent agents. However, the application areas of these AI techniques is different:

- Expert System: for diagnostic or prescriptive problem type which is based on strategies of experts. It uses expert's know-how
- Genetics algorithms: for situation that require optimal solutions. It is based on biological evolution. It uses set of possible solutions
- Intelligent agents: for specific and repetitive tasks

Neural network on the other hand is suitable for identification, classification, and prediction problem types. It make use of patterns.

## DEVELOPMENT OF SIMULATION MODEL

After covering the Web service technology and AI techniques, we describe here an approach to use them together to construct simulation model of operational business processes in business networks. As mentioned earlier, the business process choreography component of the web service provides a way to specify high level structural elements of operational business processes, where as neural networks help in modeling behavior of each process blocks within the business processes.

Sol (Sol, 1988) proposed a framework for presenting approaches in terms of way of thinking, way of working, way of modeling, and way of controlling. The way of thinking refers to the philosophy that is used in the approach. The way of working specifies the steps that are to be taken in order to realize the approach. The way of controlling specifies the guidelines and set of directives

(e.g. management of time, means and quality aspects) that are to be followed while using the approach. Finally, the way of modeling defines the modeling concepts that are used in order to use the approach. In this paper, we will focus on the way of modeling, i.e. we describe the modeling concepts that are going to be used in order to use the approach in solving the problem, which is modeling dynamic behavior of operational business processes of a business network.

To start with, the purpose of a model is to reduce the complexity of understanding or interacting with a phenomenon by eliminating the detail that does not influence its relevant behavior (Curtis et al., 1992). In order to model dynamic behavior of operational processes in business networks, we adopt the problem solving cycle (Mitroff et al, 1974, Sol, 1982). Our concern here is on the first two problem solving activities: conceptualization (a way to define the problem structure) and specification (a way to define an empirical model that gives detail specification of the situation).

In order to develop a conceptual model of operational processes of organizations, structural elements (activities, sequence flow, and decision rules) are needed. Given the fact that organizations in business networks are autonomous and their operational processes are not transparent, it is challenging to get detailed structural elements of the processes. However, as mentioned in Section 3.1, the abstract part of the processes are visible and extractable. One of the ways to retrieve these information is by referring to business processes choreography specified in the many of the web services deployed (or in UDDI registries). The processes are deployed in several specification standards. Here, we show an example using one of the many proposed standard: Web Service Conversation Language (WSCL). Figure 5 shows a UML diagram of a simple purchase activity between two trading partners. Its representation in WSCL XML is shown in Figure 6.

WSCL XML schema uses four main specification elements to describe conversations between trading partners: *Document type descriptions* (specify the types i.e. schemas, of XML documents the service can accept and transmit in the course of a conversation), *Interactions* (model the actions of the conversation as document exchange between two participants), *Transitions* (specify the ordering relationships between interactions), and *Conversation* (list all the interactions and transitions that make up the conversation). Within the bounds of a *conversation* element, the two important elements for constructing conceptual model are the *Transitions* and *Interactions* elements. *Interactions* represent the abstract process building blocks/nodes of the processes and *Transitions* specify the sequence between the nodes in order to execute a business process. Therefore, from WSCL specification, it is possible to identify structural elements of the abstract part of an operational business process. Other choreography standards that have formal represen-

tation in XML or other similar types can also be used as well. The important thing is to map the business process specification standard's metamodel to simulation model specification metamodel. Once the metamodel is mapped, the next step is to create simulation software specific constructs by using the existing building blocks of the simulation software. Afterwards, a parser is created that map an instance of business process (specified using the standards) to simulation model. By this way, a conceptual model of the business processes is created.

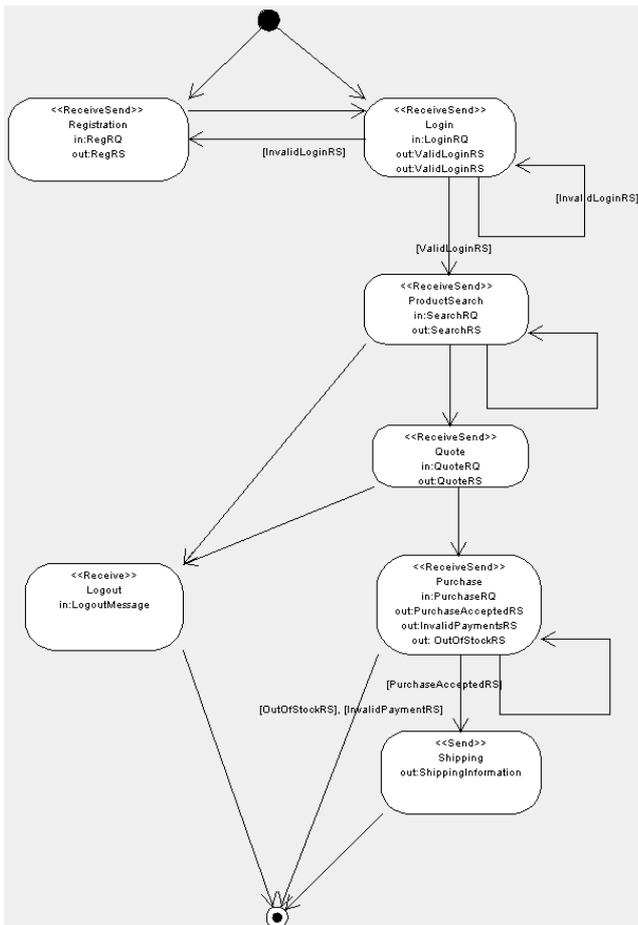


Figure 5: UML activity diagram of a purchase business process (extracted from W3.org site)

```

<?xml version="1.0" encoding="UTF-8"?>
<Conversation name="StoreFrontServiceConversation"
  xmlns="http://www.w3.org/2002/02/wscl10"
  initialInteraction="Start" finalInteraction="End" >
<ConversationInteractions>
<Interaction interactionType="ReceiveSend" id="Login">
  <InboundXMLDocument hrefSchema="http://conv123.org/LoginRQ.xsd"
    id="LoginRQ"/>
  <OutboundXMLDocument hrefSchema="http://conv123.org/ValidLoginRS.xsd"
    id="ValidLoginRS"/>
  <OutboundXMLDocument hrefSchema="http://conv123.org/InvalidLoginRS.xsd"
    id="InvalidLoginRS" />
</Interaction>
<Interaction interactionType="ReceiveSend" id="Registration">
  <InboundXMLDocument hrefSchema="http://conv123.org/RegistrationRQ.xsd"
    id="RegistrationRQ"/>
  <OutboundXMLDocument hrefSchema="http://conv123.org/RegistrationRS.xsd"
    id="RegistrationRS"/>
</Interaction>
...
</ConversationInteractions>

<ConversationTransitions>
<Transition>
  <SourceInteraction href="Start"/>
  <DestinationInteraction href="Login"/>
</Transition>
<Transition>
  <SourceInteraction href="Start"/>
  <DestinationInteraction href="Registration"/>
</Transition>
<Transition>
  <SourceInteraction href="Registration"/>
  <DestinationInteraction href="Login"/>
</Transition>
...
<Transition>
  <SourceInteraction href="ProductSearch"/>
  <DestinationInteraction href="ProductSearch"/>
</Transition>
<Transition>
  <SourceInteraction href="ProductSearch"/>
  <DestinationInteraction href="Quote"/>
</Transition>
...
</ConversationTransitions>
</Conversation>

```

Figure 6: WSCL XML specification of a purchase business process

The next step is specification of the details of each processes i.e. behavior of each processes. The internal state and structure of these abstract business process is hidden is not transparent from outside. Therefore, the usage of neural networks is appropriate to mimic the behavior of each of the building blocks by monitoring actual business transactions and conversations and creating a relationship between input values/requests and output values/responses.

The final step is to combine the neural network models of each of the simulation building blocks with the building blocks extracted from the WSCL specification. The overall activity is depicted in Figure 7.

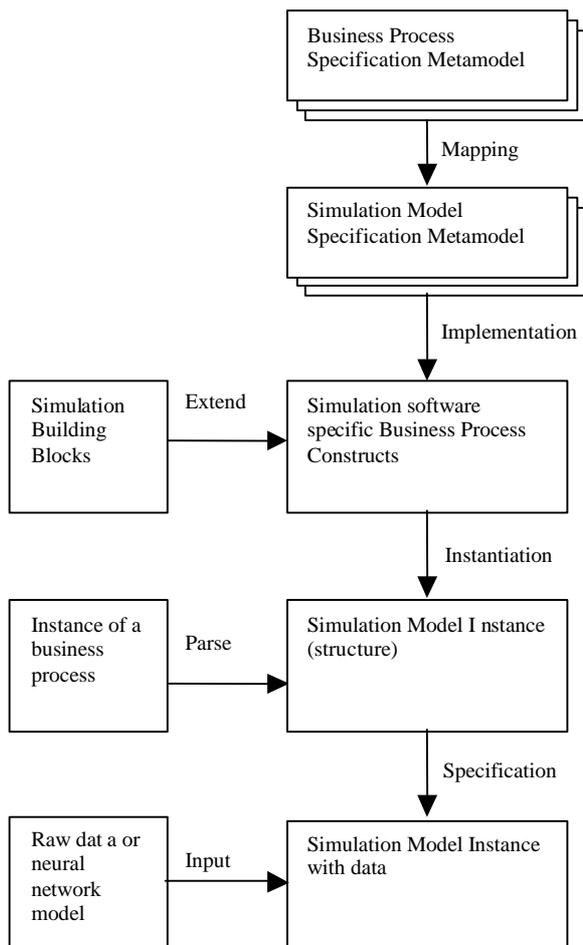


Figure 7: Steps taken to develop simulation models

## FEASIBILITY TESTS AND CONCLUSIONS

Experiments have been conducted to evaluate the described approach for modeling operational processes of organizations in a business network by using a case study that consists of two supply chain partners in interaction.

The case is as follows:

*“A kitchen equipment store in New York frequently orders espresso machines from a manufacturer in Italy. The store uses make-to-stock ordering policy. The store is known for its high quality service. The store manager wants to keep the service quality of the store and at the same time wants to lower costs by avoiding high stock quantity of espresso machines. An out of stock situation for the store is not an option, since it affects the service quality. Therefore, the store manager wants to model, among other things, the order processing delay of the espresso-making machine manufacturer”*

In the experiment, we assume that both the supply chain partners are using electronic means to conduct business transactions. Moreover, we adopt the emerging paradigm of business interaction/exchange as a service. The

supply chain partners exchange by opening their services to outside using web service technology. Interfaces of the available services are opened and descriptions of each of the interface (including the data types) are made visible. A service is also considered as consisting of sub services that address different tasks (layering of services). The Web services paradigm allows organizations to describe external structure of their processes.

To construct operational business process models of a supply chain partner, an identification of the structure and behavior of the process blocks are necessary. We follow two approaches to construct the structure and behavior of the business processes blocks. The structure of the business processes has in this case been constructed from the interface description of the different web service blocks in WSDL (Web Service Description Language) that constitute the business process. The behavior of each of the process blocks is modeling using a neural network. By monitoring the input and output of the service blocks i.e. monitoring the different business transactions (requests/responses), the behavior of a particular process block has been modeled. In this case a Multilayered Feedforward neural network architecture was used, together with a Java based neural network engine, in order to implement the approach. The supply chain case has been programmed in Java and the interaction between the supply chain partners was simulated using the Java based discrete-event simulation engine D-SOL (Jacobs, et al., 2002). The neural network was loosely integrated with the supply chain simulation model. A simple operational process with one input and one output was used to model the behavior of the business processes. Thereby, a model could be made in spite of the fact that the underlying real business process of the partner remained hidden. The aim was to see whether the supply chain actor could model the processing delay time of its partner by monitoring its order request and response delay. In the experiments, the neural network was trained in real-time from the start of the simulation and provided predictions for every next interaction with the partner. After less than 100 business interactions, the average predicted value produced by the neural network was significantly close (error of less than 5%) to the actual value that was unknown to the model. This indicates that the neural network can indeed be used for modeling the behavior of other partners, at least in cases where we reduce the business processes to one input-output neural network model. However, further detailed studies need to be conducted to see whether it can be applied to more complex operational processes that depend on time and other factors.

The next step will be to identify and extract the structural elements of business processes from Web services business process specification standards such as WSCL and BPEL. The moment these models are automatically generated, they can be easily combined with the neural network behavioral models that have been discussed in this paper.

## REFERENCES

- Aguilar, M., Rautert, T., Pater, L., Business Process Simulation: A fundamental step supporting process centered engineering, In Proceedings of Winter simulation conference, 1999
- Banerji, A., Bartolini, C., Beringer, D., Chopella, V., Govindarajan, K., Karp, A., Kuno, H., Lemon, M., Pogossiants, G., Sharma, S., Williams, S., Web service conversation language (wscl) 1.0 w3c note, <http://www.w3.org/TR/wscl10/>, 2002
- Benatallah, B., Casati, F., Toumani, F., and Hamadi, R., Conceptual Modeling of Web Service Conversations, In Proceedings of 15<sup>th</sup> International Conference of Advanced Information Systems Engineering, (Elder, J. and Missikoff, M., eds), Springer, pp. 449-467, 2003
- Chandrasekaran, S., Silver, G., Miller, J.A., Cardoso, J., Sheth, A.P., Web service technologies and their synergy with simulation, In Proceedings of the 2002 Winter Simulation Conference, 2002
- Chao, L.C., Skibniewski, M.J., Estimating construction productivity: neural network-based approach, Journal of Computation in Civil Engineering, ASCE, 8(2), 234-251, 1994
- Curtis, W., Kellner, M.I., and Over, J., Process Modeling. In: Communications of the ACM, 35(9), pp. 75-90
- Flood, I., Simulating the construction process using neural networks, In Proceeding of 7<sup>th</sup> International Symposium on Automation and Robotics in Construction, Bristol, UK, June, 1990
- Hengst, M., Sol, H.G., The impact of information and communication technology on interorganizational coordination, In Proceedings: Hawaii International Conference in System Sciences, Island of Maui, Hawaii, 2001
- Jacobs, P., Lang, N., Verbraeck, A., D-SOL:A distributed java based discrete event simulation architecture, In Proceedings: Winter Simulation Conference 2002, San Diego, 2002
- Heuvel, Van den W., Weigand, H., Coordinating web-service enabled business transactions with contracts, in Proceedings of CaiSE 03 conference, J.Eder and M. Missikoff(Eds.), pp. 568-583, 2003
- Kilmer, R.A., Smith, A. E., Shuman, L.J., Neural networks as a metamodeling technique for discrete event stochastic simulation, Intelligent Engineering Systems Through Artificial Neural Networks, Volume 4, (Dagli, C.H., Fernandez, B.R., Kumara, R.T.S., editors), ASME Press, pp. 1141-1146, 1994
- Mitroff, I.I., Betz, F., Pondy, L.R., and Sagasti, F. On Managing Science in the System Age: Two Schemas for the Study of Science as a Whole Systems Phenomenon, Interfaces, Vol. 4, No. 3, pp. 46-58
- Panayiotou, C.G., Cassandras, C.G., and Gong, W.B., Model Abstraction for Discrete Event Systems Using Neural Networks and Sensitivity Information, in: J.A. Joines et al., Proceedings of the 2000 Winter Simulation Conference, pp. 335-341, 2000.
- Seligmann, P.S., Wijers, G.M., Sol, H.G., Analyzing the structure of IS methodologies: An alternative approach, in:Proceedings of First Dutch Conference on Information Systems, Amersfoort, Netherlands, 1989
- Sol, H.G. Simulation in Information Systems Development. Dissertation, University of Groningen, 1982.
- Sol, H.G., Information systems development: A problem solving approach, in:Proceedings of 1988 INTEC symposium on Systems Analysis and Design, Atlanta, Georgia, 1988
- Wijers, G.M., Modeling support in information systems development, Doctoral Dissertation, Delft University of Technology, Delft, Netherlands, 1991
- Wombacher, A., Mahleko, B., Risse, T., Classification of Ad hoc Multi-lateral collaborations based on workflow models, Symposium on Applied Computing (ACM-SAC'03), Melbourne, Florida, 2003
- Yang, J., Papazoglou, M.P., Interoperation support for electronic business, Communications of the ACM, Vol. 43, No. 6, pp.39-47, 2000

## AUTHOR BIOGRAPHIES

**ALEXANDER VERBRAECK** is an associate professor in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology, and a part-time full professor in supply chain management at the R.H. Smith School of Business of the University of Maryland. He is a specialist in discrete event simulation for real-time control of complex transportation systems and for modeling business systems. His current research focus is on development of generic libraries of object oriented simulation building blocks in C++ and Java. Contact information:

[<a.verbraeck@tbm.tudelft.nl>](mailto:a.verbraeck@tbm.tudelft.nl)

[www.tbm.tudelft.nl/webstaf/alexandv](http://www.tbm.tudelft.nl/webstaf/alexandv)

**TAMRAT WOLDU TEWOLDEBERHAN** is a junior researcher in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. He participates in the BETADE research program on developing new concepts for designing and using building blocks in software engineering, simulation, and organizational modeling. Contact information: [<tamratt@tbm.tudelft.nl>](mailto:tamratt@tbm.tudelft.nl)

[www.tbm.tudelft.nl/webstaf/tamratt](http://www.tbm.tudelft.nl/webstaf/tamratt)



# **SIMULATION IN MANUFACTURING, PRODUCTION AND LOGISTICS**



# JOBS SEQUENCING IN INDUSTRIAL PLANTS BY MULTI-OBJECTIVE OPTIMIZATION BASED ON A SYSTEM OF AUTONOMOUS GENETIC AGENTS

Roberto Mosca, Filippo Queirolo and Flavio Tonelli  
Department of Production Engineering – Savona  
University of Genoa  
Via Cadorna, I-17100  
Savona, Italy  
E-mail: [filippo.queirolo@dip.unige.it](mailto:filippo.queirolo@dip.unige.it)

## KEYWORDS

Job sequencing, simulation optimisation, parallel genetic algorithms, evolutionary multi-objective optimisation, autonomous genetic agents

## ABSTRACT

Simulation Optimisation is one of the hottest topics in the M&S area. Evolutionary Computation has been shown to have great synergy with simulation both for fitness assessment and constraints description.

In this paper, the Authors discuss a general architecture for job sequencing in a case of multi objective simulation optimisation by agent-directed simulation. Autonomous genetic agents have been used.

## INTRODUCTION

This paper addresses a real case involving a manufacturing company producing and commercialising mineral water and soft drinks.

The firm is interested in the enhancement of the:

- weekly production rate, measured in terms of the number of bottles produced over a certain period;
- service level, evaluated by the delay with respect to the due dates.

Notice that the maximisation of the weekly production rate immediately implies to minimise the total set-up time and therefore to determine the optimal production sequence.

The firm plan to achieve these goals by a progressive process of re-organisation, whose first phase has been successfully completed. In this paper the Authors focus on the improvement of the planning, scheduling and sequencing process. This is a real world case of innovation based on information technology and processes re-engineering applied to production management in an industrial plant.

In a previous work [Mosca et al. 2002], the Authors presented the results achieved during the first phase of this re-organisation process; now they discuss the extension of the architecture of the scheduling system and show preliminary results.

In the next section, the Authors briefly present the production process (three production lines alternatively working) and detail the assumptions and the process modelling stages: it is shown that the proposed case belongs

to the class of independent job sequencing problems. A short argumentation is then reported in order to formulate it as a Simulation Optimisation (SO) problem over a non-parametric input domain and to show that Evolutionary Computation (EC) represents the technique most commonly agreed to be suitable for this kind of SO problem. The relevance of the proposed problem has been addressed both in the area of scheduling and sequencing [Pinedo 2002] and SO [Jacobson and Schruben 1989].

The various stages followed during the development of the simulation model are then summarised. This process has been structured according to Williams and Narayanaswamy [Williams and Narayanaswamy 1997], who further refer to [Banks and Gibson 1996], following the key tasks needed for a successful simulation analysis.

The Authors then state that optimisation of the set-up time and earliness-tardiness is required, to be estimated by simulation on the basis of the job sequencing input. This is a Multi Objective Optimisation (MOO) problem.

In order to justify the proposed architecture, with respect to previous literary works, an extreme synthesis of the scientific work developed in the area of MOO by using EC is provided and a special focus (paragraph 5) is dedicated to MOO by Parallel Genetic Algorithms (PGA), in particular to Coarse-Grained Genetic Algorithms (CGGA).

The Authors focus on the algorithm selection process (a sort of CGGA have been adopted), the software architecture (autonomous agents communicating by blackboard protocol) and to the modification of previous PGA paradigms (introduction of an agent for the subpopulations merging and selection of the new individuals). The paper ends with some preliminary results and considerations about the next stages of the projects.

## DESCRIPTION OF THE PROBLEM ENTITY

The proposed industrial case refers to a manufacturing process for mineral water and soft drink bottling, with three semi-automated production lines producing twenty-one different product categories. The equipment is connected by rolling tapes and this is a flowshop process. Resources are requested on each line for producing and the number of operators required varies: 8 employees for line 1 and line 2; 10 employees for line 3. The production-aimed resources pool counts 10 persons: simultaneous production by two lines is unfeasible. Equipment allows to produce different

sizes of products (i.e. 500ml, 750ml and 1000ml), but long setup times are required (2-6 hours) for each format change. Production is organised according with a make to stock policy. Nevertheless due dates should be considered, especially in the case of large batches requested by major customers on the basis of an open contract and to the requirement (i.e. product code, date and quantity) of the Planning Department stated in the master production schedule [Aloi et al. 2002]. As a result, from the production manager point of view each batch has a proper product specification, required quantity (i.e. number of pallet) and due date (i.e. day), either derived by direct customer needs or Planning Department requirements. Producing by line 1 and 2, two persons would remain inactive; thus preforms insuffling or setup activities on another production line can be assigned to them. Notice that performing the set-up of one of the inactive lines allows reducing the total completion time, by partial or full hiding the setup times,.

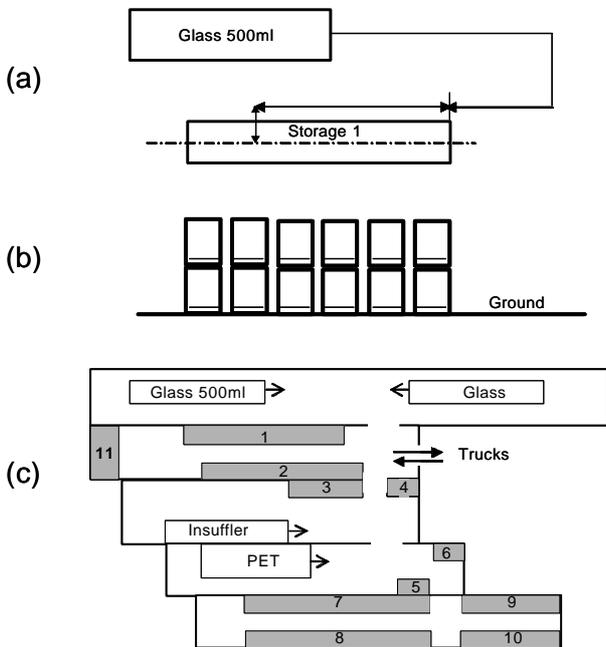


Figure 1: (a) Material Handling from a Production Line to the Pre-assigned Storage Area. (b) Schematic View of a Storing Block. (c) Production Plant.

Lot sizing is performed according to the shift length: a job always starts at the beginning of a shift and ends at the end of (that or a subsequent) shift. In this way, stops due to setup operations are reduced. Finished goods are then stored in a pre-assigned area before the pick-up for distribution and commercialization. As usual in several SME involved in the large consume market, storage represents a critical issue, especially when trucks arrivals and departures are affected by high variability. The saturation of the storing area should be taken into account during the scheduling process.

### ASSUMPTIONS AND MODELING PHASE

Because of the features of the production processes, the scheduling problem crystallises in a job sequencing

problem. Indeed, allocating a product on a line means also to define the starting and completion times. It means that the goal is determining the sequence that optimizes the performance indices. One can indeed think of each line as a single block, which transforms raw materials in finished goods by collapsing the whole bottling process in a single black box. In this prospective, the parameters of each line can be estimated on the basis of the production time series of that specific line.

Sequencing is an integer optimization problem on bounded domain (i.e. an integer constrained optimisation problem), that has been shown to be NP-hard [Du and Leung 1990], even in case of total tardiness minimization only.

For these reasons (i.e. complexity and pivotal role in scheduling heuristics) and because the proposed industrial case belong to real world domain, the Authors consider it relevant from a scientific point of view.

### IS THIS A SIMULATION OPTIMISATION PROBLEM?

With the great incidence of simulation modelling over a huge number of areas, it has been essential to extend the scope of traditional optimisation to include simulation domain. Simulation Optimisation (SO) is the scientific field that provides a structured approach to determine the optimal values for the input (operating) parameters of a simulation model, according to a performance measure. Simulation models can indeed be used as the objective function and/or constraints functions in optimising stochastic complex systems.

Azadivar [Azadivar 1999] (first formulation pag. 94) and Joshi et al. [Joshi et al. 1996] define SO as the problem of finding an input vector  $\mathbf{X}$  that minimises:

$$f(\mathbf{X}) = E[r(\mathbf{X})] \quad (1)$$

subject to the following sets of stochastic and deterministic constraints:

$$sc(\mathbf{X}) = E[s(\mathbf{X})]$$

$$dc(\mathbf{X}) < 0$$

where  $f$  and  $sc$  respectively are the unknown expected values of the objective function(s) and the set of stochastic constraints evaluated on the basis of the random vectors  $r$  and  $s$ , and  $dc$  is a set of deterministic constraints.

Jacobson and Schruben [Jacobson and Schruben 1989] shown that this problem is hard to solve.

Objective function(s) evaluation by simulation leads to enormous advantages, especially if does not exist any analytical expression of the goal function and/or the objectives function(s) and the constraints are stochastic functions of the deterministic decision variables. Moreover simulation allows reaching relevant improvements in the description and characterisation of the problem (e.g. accurate representation of the physical and logical constraints).

On the other side, using simulation as an aid for optimising presents specific challenges, such as those related to the optimisation of complex and highly non-linear functions. Further, the efficiency of the optimisation algorithm is more crucial, since objective function evaluation is performed by simulation run instead of calculation of an analytical expression.

SO techniques are generally classified on the basis of the nature (continuous/discrete/non parametric) and structure (quantitative/qualitative) of the input space. Table 1 shows a common classification of SO techniques; Table 2 focuses on the commonly used techniques in each one of the four classes previously highlighted. Notice that SO techniques have been inserted in the most suited class: e.g. Response Surface Methodology or Nelder-Mead Method have been largely used also in discrete domain since 80's [Mosca and Giribone 1985; Mosca et al. 1986a and 1986b].

Table 1: Short Classification of SO Techniques

Input Parameter	Input Structure	Reference of Surveys
Continuous	Quantitative	[SWISHER2000]
Discrete Small	Quantitative	[CARSON1997]
Discrete Large	Quantitative	[CARSON1997]
Non Parametric	Quali/quantitative	[Azadivar1999]

In this mind, when any matter of sequencing and scheduling problems [Pinedo 2002] arise in complex systems and reaching an accurate description of all (relevant) production constraint is required, a SO problem over a non-parametric input domain can be stated. According with Table 1, these problems can be successfully faced by evolutionary computation. With particular attention to the real case described in this paper the problem formulation reported below is appropriated. Notice that it refers to a complex production system and requires that all the relevant production constraints are fully considered.

It is required to determine the weekly schedule (i.e.  $X$ , the non-parametric input of the simulation) that minimises the expectation of the multi goal function  $f$  over a set of replicated simulation runs:

$$f(X) = (E[f_1(X)], E[f_2(X)]) = (E[r_1(X)], E[r_2(X)]) \quad (2)$$

subject to:

- the availability of storing positions, that is definitively due to the stochastic arrivals and departures of the trucks. They are  $sc$  in Equation (1).
- the physical constraints of the production plant and the resource pool. They are  $dc$  in Equation (1).

The goal function  $f$  can be informally defined as follow:

- $f_1$  is the total set-up time, observed over a week.
- $f_2$  is the service level, evaluated by the total earliness and tardiness penalty (ETP) [Baker1995]:

$$ETP = \sum_{j=1}^{N \text{ of Items in Advance}} E(j) \cdot Qt(j) \cdot EW(j) + \sum_{k=1}^{N \text{ of Items in Delay}} T(k) \cdot Qt(k) \cdot TW(k) \quad (3)$$

where  $E(T)$  is the earliness (tardiness) of the  $j$ -th batch in advance (in delay), measured by time units;  $Qt$  is the required quantity of product for a batch;  $EW(TW)$  is the penalty due for a unit of product stored in the assigned position in advance (in delay) of a unit of time regarding to its due date;

On the basis of the argumentations and references provided in this section, this problem can be formulated as a multi

objective SO over a non-parametric domain. The Authors faced it by Evolutionary Multi Objective Optimisation (EMOO).

Table 2: SO Techniques

Input	Mostly Adopted Techniques
Continuous	Gradient Approaches; Response Surface Methodology; Stochastic Approximation; Nelder-Mead Method; Hooke-Jeeves Method
Discrete Small	Importance Sampling; Ranking and Selection; Multiple Comparison
Discrete Large	Evolutionary Computation; Evolutionary Strategies; Simulated Annealing; Tabu Search;
Non-Param.	Evolutionary Computation

## EVOLUTIONARY MULTI OBJECTIVE OPTIMISATION

EC mainly refers to Genetic Algorithms (GAs). In this paper, the Authors avoid any description both of the basic principles of EC and evolutionary mechanisms (i.e. selection, crossover or mating, mutation of the individuals of the population). See [Goldberg 1989] for a general review about GAs and [Mosca et al. 2002] for a detailed explanation of the specific crossover and mutation mechanism adopted for the evolution of both the populations in this application.

Since the proposed case belongs to the class of multi objective SO problems over a non-parametric domain, EMOO is the appropriate paradigm to be adopted as optimisation technique.

MOO requires determining a set of Pareto Optimal solutions instead of a single optimal configuration. "A solution is said Pareto optimal, or non-dominated, if starting from that point in the design space, the value of any of the objective function cannot be improved without deteriorating at least one of the others" [Cardon et al. 1999].

EC seems particularly suitable to solve MOO problems because evolutionary algorithms simultaneously deal with a set of possible solutions, which allows finding an entire set of Pareto optimal individuals in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques [Coello 1999].

Even if no reference has been published, explicitly addressing the problem of Pareto Optimality in SO, large literature is available in the field of EMOO with respect to this problem. Regardless to the approaches that leads to a degenerate MOO two main concepts can be identified in EMOO, such as aggregating functions [Fonseca and Fleming 1997] or Target Vector Approaches [Coello 1999]. A first research direction aims to determine optimal solutions by directly minimising the vector of objective functions in a Pareto sense (see figure 2). This is the case of MOGA [Fonseca and Fleming 1993], NSGA [Srinivas and Deb 1993], NPGA [Horn and Nafpliotis 1993].

Figure 3 shows the expected behaviour of the proposed approach: by selecting the sub-populations recombination interval (i.e. the migration interval in the Coarse-Grained Parallel Genetic Algorithms literature [Cantù-Paz 1998]) it is possible to drive the optimisation process through several

parallel directions in order to determine the searched set of Pareto optimal solutions. Notice that GAs are inherently parallel.

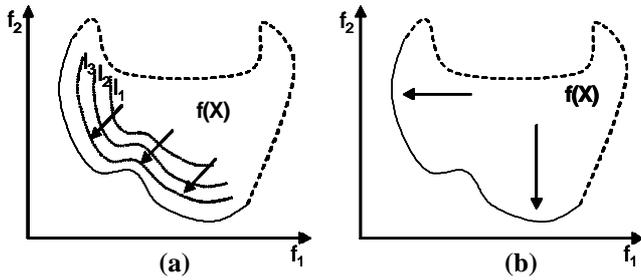


Figure 2 : (a) Pareto Optimality Increases from  $I_1$  to  $I_3$ . (b) Pareto Optimality is Reached by the Minimisation of each Single Objective Function

The second direction, which at the basis of the proposed approach, is largely discussed in [Grefenstette 1984] and in [Shaffer 1985], presenting Shaffer's VEGA. The idea behind VEGA is based on the optimisation of each single objective function by the division of the initial population in two (or more depending on the number of functions) sub-populations. Each sub-populations is generated by selecting individuals according with one of the objective functions; the sub-populations are then shuffled together in order to obtain a population to be mated and mutated in a single GA. See Figure 3.

A main problem of VEGA is speciation that is the excellence of some individuals on specific aspects of performance. This leads to the evolution of "species" within the population due to the mono-directionality of the selection mechanism. A crucial point is that "middling" (i.e. individuals with whole acceptable performance, but not outstanding for any objective functions) can be prevaricated by specialised individuals, avoiding the essential evolution of compromising solutions.

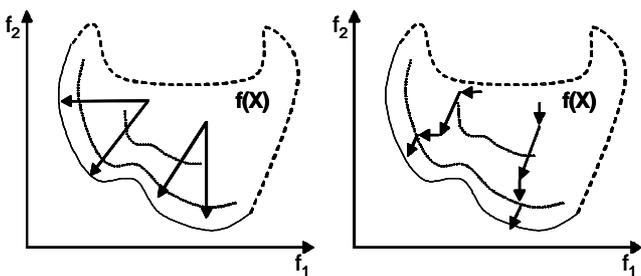


Figure 3 : Expected Behaviour of the Proposed Approach

In this paper VEGA approach to EMOO is extended to the case of multiple parallel genetic algorithms and a specific mechanism for sub-populations recombination has been implemented. Figure 4 shows a schematic representation of the proposed concept. Starting from a random population, two genetic algorithms perform the evolution of the individuals. Within each GA, individuals are selected according with a pre-assigned objective function or with a mixture of some of them. In this way it would be also possible to implement different crossover and mutation

mechanisms in different GA, according with the specific issues of the objective function to be optimised by the GA. Even if this approach could present some advantages, it is still affected by the some behaviour observed in the Schaffer's VEGA. In this mind, a periodic recombination of the sub-populations has been implemented in order to allow the selection of the individuals according to a Pareto ranking criterion.

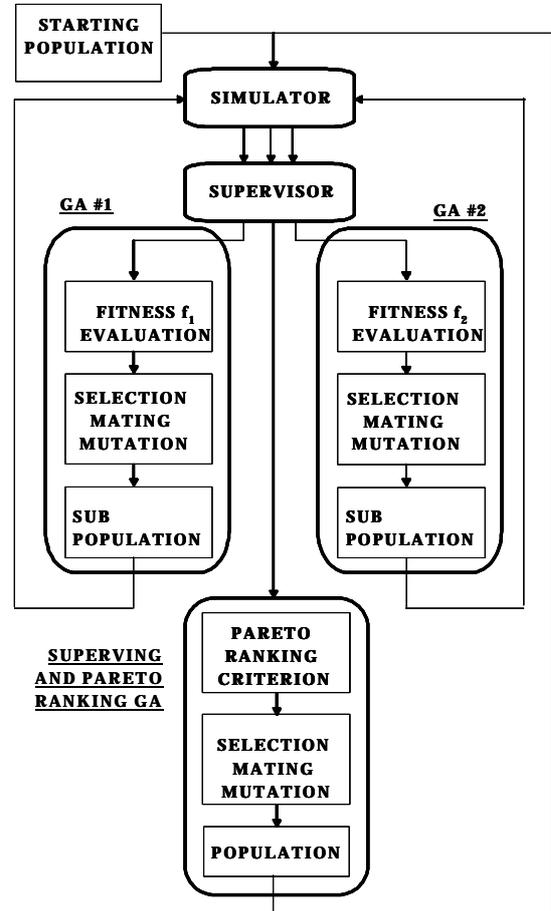


Figure 4 : the Proposed EMOO Architecture

Moreover a really high degree of scalability and flexibility have been reached:

- if  $n$  objective functions should be optimised, to add a number of GAs is sufficient that enables to reach the desired pool of problem solvers;
- if one of the function to be minimised would requires highly customised mechanism of optimisation, that mechanism should be implemented in the GA, assigned to that function.

### MIGRATION RATE AND RECOMBINATION

The problem of periodic recombination of two or more sub-populations is typical of Parallel Genetic Algorithms (PGAs). PGAs have been largely studied since their efficiency in solving combinatorial problems: by dividing the population, the evolution can be speed up, with regard to the communication among the GAs. The determination of the correct migration rate is still an open problem and several recent papers are mainly referred to formally

investigate the relation between the selection pressure and the migration rate [Cantù-Paz 1999].

In this work, the Authors implemented a supervising entity (later defined as an agents' supervisor) that is responsible for the migration of the individuals of each sub-population in the large populations: the interval migration rate is defined by a setting parameter that specifies the number of generation between two subsequent migration.

Selection is performed by MOGA's criterion of Pareto optimality: each individual are ranked on the basis of the number of chromosomes by which it is dominated. This choice is justified by the critical comparison provided in [Coello 1999] and by the argumentation of Goldberg and Deb [Goldberg and Deb 1991] about the rapid converge (even premature) of the algorithm. The Authors consider this property has a favourable one since the proposed framework involves the alternation of one-dimensional and Pareto ranking EMOOs.

## DEVELOPMENT OF THE SIMULATION MODEL

The *scope of the simulation model* contains: i) the bottling processes considered as single operations each one performed inside a block; ii) the human resources; iii) the line stops due to machine breakdowns; iv) the material handling and finished goods transportation inside the plant; v) the storing operations and constraints due to the real volume of the storing areas. On the basis of a cost effective evaluation (for similar approaches see [Williams and Narayanaswamy 1997] or [Jayaraman and Agarwal 1996], the Authors didn't model the availability of raw materials, the supplying and distribution chain (i.e. the logistic, transportation and supplying issues), human resources behaviour and they didn't distinguish among the different breakdowns or production interruptions.

This model was derived by the integration of two (already validated) conceptual models, [Mosca et al. 2002] and [Nan et al. 2002]. The Authors coded it by a programming language in order to have a discrete event simulator [Banks et al.1995]. Indeed, since an EMOO approach had been selected, they avoided using a commercial simulation tool, such as Automod and Autosched (optimised by AutoStat) or Micro Saint (optimised by OptQuest), even if they had been already successfully tested in the case of SO since 90's (respectively [Carson 1996] and [Drury and Laughery 1996]). In this way the complete control was reached both on the simulation model and its interfacing with the optimiser; it was crucial for the further development of the scheduling system and to preserve future extension and scalability.

The *choice of a programming language* has been driven by a need for completely manageable software allowing deep integration with the optimisation algorithm (see corresponding paragraph). Considering some specific requirements of the firm (i.e. platform independency, and remote usability and controlling of the software) and because of the suitability for the implementation of agent frameworks and architectures (Bigus framework [Bigus and Bigus 2001], Madkit [Gutknecht and Ferber 2000], Zeus or Jade platforms), the Authors considered Java has the most

suitable programming language for the proposed application.

The *objectives* of the proposed simulation model have been preliminary states in a previous paragraph by formulating the optimisation problem: by the simulator the Authors want to assess the performance of a scheduling (i.e. the job sequence provided as input to the simulator), respecting the production constraints. The *output data* measured in order to evaluate the performance of the scheduling provided in input to the simulator are the earliness and tardiness penalties and the total set-up time. Indeed, they are representative of the production rate of the plant (by the total set-up time), the cost effectiveness of the schedule (by time advance with respect to the due dates and total set-up time), and service level (by time delay).

According to [Dileepan 1993], the earliness and tardiness penalties have been calculated by (1):

$$ETP = \sum_{d-C_i>0} b_i(d-C_i) + \sum_{C_i-d>0} b_i(C_i-d) \quad (2)$$

where the subscript  $i$  refers to the  $i$ -th job and  $C_i$  is the completion time;  $d$  is the assigned due date;  $b_i$  is the earliness weight; and  $a_i$  is the tardiness weight. Notice that generally the earliness and tardiness weights are cost evaluators difficult to be estimated, even with the support of expert. Indeed it is strongly dependent on various voices of the balance sheet and on the organisation of the firm.

*Data collecting* has been performed in strict collaboration with practitioners, being known that it is a critical task in simulation [Amico et al. 2000]. Data validation has been performed during all the collecting process and initially unavailable data has been estimated on the basis of the standards (adopted by the production department) and the experience of the production personnel.

No warm-up period is here calculated since this is the case of terminated simulation [Banks et al.1995] over a time  $[0; T_w]$ , where  $T_w$  equals a working week. The Authors searched for a trade off between the confidence interval (estimated according with the MSp in [Mosca et al. 1982]) and the number of replications to be performed. Since the goal function  $\mathbf{f}$  is a vector of objective functions, the analysis of the number of iterations has been performed by the total weekly production (i.e. number of pallets). Figure 5 shows the amplitude of the interval of confidence versus the number of simulation runs. For each value of simulation runs, 10 replications have been performed, using different random stream. As a results, the average amplitude (solid line) seems to stabilise when 14 simulation runs are performed. Notice that the vertical lines progressively reduce until experiments are performed over 11 simulation runs; it provides an estimation of the standard deviation of the amplitude of confidence interval. According with the accuracy required for the specific testing case, the Authors estimate each value of the goal function by running 6 simulation runs. In this way the experimental error can be reasonably considered lower than 1.1% (basis the weekly production).

Since the Authors previously developed and tested the conceptual models now integrated in the described simulator, the Verification and Validation (V%V) phase especially focused on the testing and of the computerized

model by comparing the simulation model with the validated ones. Moreover, a full validation of the system has been performed by the following V&V techniques [Sargent 1999]:

- structured walkthrough of the model logic;
- Operational Graphic;
- Predictive Validation, quantitative statistical comparison of the estimated working rates of the equipment (i.e. the utilization rates) with the corresponding values observed during the simulation;
- Deep analysis of the simulation trace;
- Turing Tests.

Last two stage of the V&V process have been performed in strict collaboration with the production personnel.

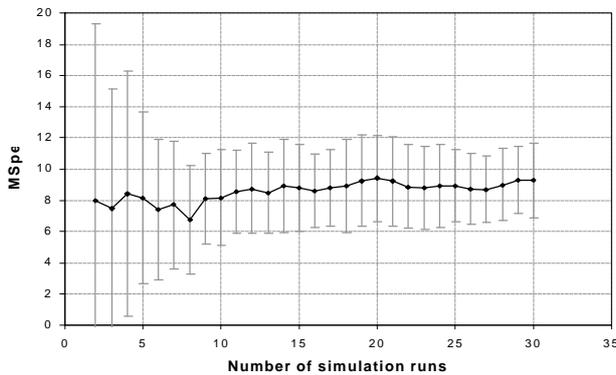


Figure 5: Amplitude of the Interval of Confidence versus the Number of Simulation Runs. For each Number of Simulation Runs, 10 Replications with different random stream have been performed

### WHY AN AUTONOMOUS GENETIC AGENTS ARCHITECTURE?

Referring to Prof. Ören's Invited Paper at the St. Petersburg Workshop on Simulation [Ören 2001c], the Authors implemented the proposed concept in an agent software architecture. As an evolution of the application of artificial computation in statistics, software agent is emerging as a key research area. Especially considering simulation, software agents sounds as a promising paradigm for agent-support in simulation, which is design of experiments, simulation-based optimisation, and analysis of simulation results [Ören 2001a and 2001b].

Genetic Agents represents hence an important paradigm since they are suitable for optimising complex system over a non parametric domain (one of the hottest research area [Law and McComas 2000]) and have cognitive abilities such as autonomy, goal processing and input evaluation.

Notice that according the whole argumentation provided in this paper, these exciting features always require coupling each genetic agent with a validated simulation model.

The agent architecture presented in this paper uses a blackboard communication protocol. Coherently with its primary aim [Erman et al. 1980], it is used to share indirectly data by a common knowledge exchange place.

An agents' supervisor monitors the evolution of the genetic agents and decide for the recombination of the sub-

population according with the framework previously discussed.

### PRELIMINARY RESULTS

The proposed architecture has been preliminary tested on the real problem presented at the beginning of this paper. Some results have been obtained. They showed a certain instability of the average value of the fitness function, probably due to the re-combination of multiple research approaches. Nevertheless the definitive performance results in a significantly improvements of the production schedule with respect to the previous scheduling system: the total set-up time has been reduced by 4% and the earliness/tardiness function records savings higher than 5.8%. The computational effort is significantly higher than in [Mosca et al. 2002]: even if the number of simulation runs to be performed in this case (i.e. 5) is lower than the correspondent value adopted in the previous simulations (i.e. 11 runs), more than twice the time needed for standard GA is now required.

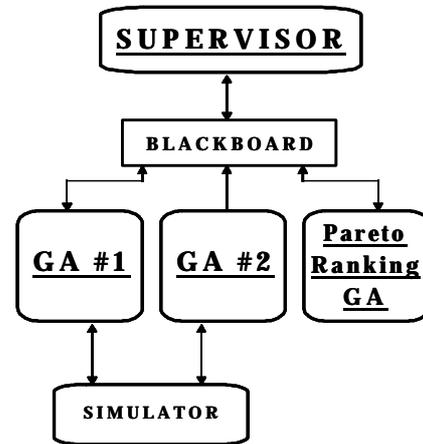


Figure 6: the Proposed Agents Architecture

### CONCLUSIONS AND FUTURE WORK

In this paper a general architecture has been introduced starting from a real industrial problem. The proposed logical scheme uses EMOO in a case of SO over a non-parametric domain. Two genetic agents and a supervisor communicate by a blackboard protocol in order to implement collaborative problem solving based on evolutionary algorithm. Simulation has been shown to be crucial in EC both for fitness assessment and constraints description. As a result, agent-directed simulation optimisation has been discussed and architecture proposed. Novelty refers to the extension of Shaffer's VEGA approach, in order to mitigate the "middling" effect, and to the structured formulation of a scheduling problem as a multi objectives SO case.

Future work will be mainly devoted to investigate the relationship among the different parameters and to implement an algorithm for novelty detection in the sub-populations in order to adaptively decide whether combine (i.e. perform migration of) the individuals.

## REFERENCES

- Aloi U; R. Mosca; F. Queirolo; and F. Tonelli. 2002. "Strategic Planning and Production Control: Deriving an Useful Master Production Schedule from Sales Forecasts". *Proceedings of Summer Computer Simulation Conference*. (SCSC'02). San Diego (CA)
- Amico V.; R. Guha; and A.G. Bruzzone. 2000. "Critical Issues in Simulation", *Proceedings of Summer Computer Simulation Conference*, Vancouver, July
- Azadivar F.. 1999. "Simulation optimization methodologies". *Proceedings of the 1999 Winter Simulation Conference*, 93-100.
- Baker K.R.. 1995. "Elements of Sequencing and Scheduling". Amos Tuck School of Business Administration. Dartmouth College, Hanover, New Hampshire.
- Banks J.; J.S. Carson; and B.L. Nelson. 1995. *Discrete Event System Simulation. Second Edition*. Prentice Hall, Upper Saddle River, New Jersey
- Banks J. and R.R. Gibson. 1996. "Getting started in simulation modelling. Industrial Engineering Solutions". 28(11):34-39.
- Bigus, J.P. and J. Bigus. 2001. *Constructing Intelligent Agents Using Java* John Wiley & Sons
- Cantu-Paz E.. 1998. "A survey of parallel genetic algorithms. Calculateurs Paralleles", *Reseaux et Systems Repartis*. Vol. 10, No. 2. pp. 141-171. Paris.
- Cantú-Paz E.. 1999. "Migration policies, selection pressure, and parallel evolutionary algorithms". Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference
- Cardon A.; T. Galinho; and J.P. Vacher. 1999. "A Multi-Objective Genetic Algorithm in Job Shop Scheduling Problem to Refine an Agents' Architecture", In *Proceedings of EUROGEN'99*, Jyväskylä, Finland. University of Jyväskylä.
- Carson J.S.. 1996. "AutoStat Output Statistical Analysis for AutoMod Users", *Proceedings of the 1996 Winter Simulation Conference*, 492-499
- Carson Y. and A. Maria. 1997. "Simulation optimization. Methods and applications". *Proceedings of the 1997 Winter Simulation Conference*, 118-126.
- Carlos A.; and C. Coello. 1999. "An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends", In *Congress on Evolutionary Computation*, pages 3-13, Vol. 1, Washington, D.C.. IEEE Service Center.
- Dileepan P.. 1993. "Common due date scheduling problem with separate earliness and tardiness penalties", *European Journal of Operational Research* 120, 375-381
- Drury, C. and K.R. Laughery. 1996. "Advanced Uses for Micro Saint Simulation Software". *Proceedings of the 1996 Winter Simulation Conference*, 510-516.
- Du J. and J.Y.T. Leung. 1990. "Minimizing total tardiness on one machine is NP-hard". *Mathematics of Operational Research*, 15:483-495.
- Erman L.D.; F. Hayes-Roth; V.R. Lesser; and R.D. Reddy. 1980. "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty". *ACM Computing Surveys*, 12(2):213-253.
- Fonseca C.M. and P.J. Fleming. 1993. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization". In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416-423, San Mateo, California, Morgan Kaufman Publishers.
- Fonseca C. M. and P.J. Fleming. 1997. "Multiobjective Optimization", In *Handbook of Evolutionary Computation*, pages C4.5:1-C4.5:9. Institute of Physics Publishing and Oxford University Press.
- Goldberg D.. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison- Wesley, Reading, MA.
- Goldberg D. E. and K. Deb. 1991. "A comparison of selection schemes used in genetic algorithms". In *Foundations of Genetic Algorithms*, pages 69-93. Morgan Kaufmann, San Mateo, California.
- Grefenstette J.J.. 1984. "GENESIS: A system for using genetic search procedures". In *Proceedings of the 1984 Conference on Intelligent Systems and Machines*, pages 161-165.
- Gutknecht O. and J. Ferber. 2000. "The Madkit agent platform architecture". In *1st Workshop on Infrastructure for Scalable Multi-Agent Systems*.
- Horn J. and N. Nafpliotis. 1993. "Multiobjective Optimization using the Niche Pareto Genetic Algorithm". Technical Report IlliGAI Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.
- Jacobson S.H. and L.W. Schruben. 1989. "Techniques for simulation response optimisation", *Operations Research Letters*, 8:1-9.
- Jayaraman, A. and A. Agarwal. 1996. *Simulating an engine plant*. *Manufacturing Engineering* 117(5):60-68.
- Joshi B.D.; R. Unal; N.H. White; and W.D. Morris. 1996. "A Framework for the Optimization of Discrete-Event Simulation Models". *Proceedings of the 17th ASEM National Conference*.
- Law A.M. and M.G. McComas. 2000. "Simulation-based optimisation". *Proceedings of the 2000 Winter Simulation Conference*: 46-49
- Mosca R. and P. Giribone. 1985. "The Nelder-Mead Method: A Setter Sequential Technique To Get Optimum Region In Discrete Simulation Of Complex Plants". *Proceeding III IASTED International Conference "Modelling and Simulation"*, pp 441-446, Lugano (CH)
- Mosca R; P. Giribone; and P. Guglielmo. 1986. "The steepest ascent method in the optimisation of discrete simulations of complex industrial plants: a new approach". *International Journal of Modelling and Simulation Acta Press - Volume 6*", n 3 pp96/101
- Mosca R; P. Giribone; and P. Guglielmo. 1986. "The simplex method in the optimisation discrete simulations of complex industrial plants: theory, applications, comparison". *International Journal of Modelling and Simulation Acta Press - Volume 6*", n 2 pp58/64
- Mosca R.; F. Queirolo; and F. Tonelli. 2002. "Job sequencing problem in semi-automated production process". *Proceedings of the 14th European Simulation Symposium*. Dresda. Germany.
- Nan P.; F. Queirolo; M. Schenone; and F. Tonelli. 2002. "Warehouse layout design: minimizing travel time with a genetic and simulative approach - Methodology and case study". *Proceedings of the 14th European Simulation Symposium*. Dresda. Germany.
- Ören T.I.. 2001. *Advances in Computer and Information Sciences: From Abacus to Holonic Agents*. Special Issue on Artificial Intelligence of Elektrik. 9:1, 63-70.
- Ören T.I.. 2001. "Impact of Data on Simulation: From Early Practices to Federated and Agent-Directed Simulations". In *Proceedings of EUROSIM 2001*, Delft, the Netherlands
- Ören T.I.. 2001. "Software Agents for Experimental Design in Advanced Simulation Environments". In *Proceedings of the 4th St. Petersburg Workshop on Simulation*, pp. 89-95.
- Pinedo, M.. 2002. *Scheduling - theory, algorithms, and systems. Second edition*. Prentice Hall, Englewood Cliff, N. J.
- Sargent R.G. 1999. "Validation and Verification of Simulation Models". In *Winter Simulation Conference*, IEEE, Piscataway, NJ, 39-48.
- Schaffer J. D.. 1985. "Multiple objective optimization with vector evaluated genetic algorithms. In Genetic Algorithms and their Applications". In *Proceedings of the First International Conference on Genetic Algorithms*, pages 93-100. Lawrence Erlbaum.
- Srinivas N. and K. Deb. 1993. "Multiobjective optimization using nondominated sorting in genetic algorithms". *Technical report, Department of Mechanical Engineering*, Indian Institute of Technology, Kanput, India.
- Swisher J.R.; P.D. Hyden; S.H. Jacobson; and L.W. Schruben. 2000. "A Survey Of Simulation Optimization Techniques And Procedures". *Proceedings of the 2000 Winter Simulation Conference*.
- Williams E.J. and R. Narayanaswamy. 1997. "Application of simulation to scheduling, sequencing, and material handling". In: *Winter Simulation Conference*. IEEE, Piscataway, NJ, 861-865.

## AUTHORS' BIOGRAPHIES

**Roberto Mosca** is full professor of "Industrial Plants Management" and "Economy and Business Organization" and he is currently the head of Department of Production Engineering at University of Genoa. He involved in several simulation project involving both academic and industrial partners.

**Filippo Queirolo** is a member of DIP Research Group at University of Genoa.

**Flavio Tonelli** is a complex systems management researcher in the Department of Production Engineering at University of Genoa.

# VALIDATING THE PRODUCTION WEEKLY PLAN BY SCHEDULING SIMULATION

Pietro Giribone, Roberto Mosca, Filippo Queirolo  
Department of Production Engineering – Savona  
University of Genoa  
Via Cadorna, I-17100  
Savona, Italy  
E-mail: [filippo.queirolo@dip.unige.it](mailto:filippo.queirolo@dip.unige.it)

## KEYWORDS

Production planning and scheduling; scheduling simulation; weekly plan; daily line load.

## ABSTRACT

Large Corporate and Small and Medium Enterprises are still going through deep innovation and re-engineering processes. Organisation should become leaner and more efficient.

Even planning and scheduling should be integrated in order to yield feasible and robust plan in an automatic way by continuous planning, from forecasting to daily line load.

We present a framework for weekly plan validation with respect to all the production constraints based on scheduling simulation and apply it in a real case involving the world-wide leader in Medication Delivery market.

## INTRODUCTION

A strong need exists in large, medium and small enterprises for exact validation of the plan issued by the Logistic personnel. Effective validation should lead to a feasible plan, acceptable for Production personnel.

In order to reach this goal validating the weekly plan is required with respect to all the relevant production constraints. This validation process usually starts during (weekly) meetings involving both Logistic and Production Departments and requires three time consuming stages: revision, modification and approval.

Automatically performing this process is desired in order to make more efficient the process and more reliable the plan. Simulation should be used as a support for performing this task, especially dealing with complex manufacturing system: since simulation accurately describes the process and its constraints [Carson and Maria 1997], it allows to perform the validation of the plan according to the feasibility and reliability requirements previously mentioned.

In order to fully validate a weekly plan, the low resolution plan provided by the Logistic Department should be detailed by scheduling the required products and quantities with respect to the corresponding due dates [Pinedo2002], production rate, availability of personnel and tools, WIP level and physical and logical constraints [Peterson 1998]. The scheduling process is expected to yield the allocation of each single item on one of the allowed machines, with

time resolution around minutes or even lower, depending on the degree of automation. Otherwise, production personnel could be obliged to modify the schedule according to the production constraints. In the latter case, a new adjusted schedule is delivered and communications issues arise. Limiting at the minimal level the modification of the plan once it has been delivered by the Logistic Department is crucial: the inferior bound for unexpected variations of the plan are due to inherent stochastic nature of the production process, such as equipment breakdowns. Thus, starting from a list of requirements a detailed job allocation should be performed. This is just a matter of scheduling.

As a result, the weekly plan validation should be performed by simulation, since all the relevant constraints should be considered in order to avoid any subsequent modification of the plan, and the simulation model should be integrated by scheduling algorithms, in order to detail the weekly plan at an operations level as required by simulation.

With respect to the knowledge of the Authors, no simulation framework have been presented, explicitly addressing this topic.

## SCHEDULING SIMULATION

The idea of scheduling simulation, as presented in this paper, has a twofold aim:

- firstly, to answer to the specific requirements of Companies for a new paradigm of planning system allowed to provide monthly/weekly plan validated at finite capacity with respect to all the deterministic constraints that affect operations;
- secondly, to support cost-reduction strategies by technology, especially referring to internet-working, largely distributed supply chains, remote planning and spreading of industrial plants world-wide.

This view results in a clear statements:

*to be able to reproduce manufacturing operations and scheduling process everywhere, in a reliable and robust manner, respecting the physical and logical constraints typical of the production process, progressively making leaner and more efficient the enterprise organisation.*

The proposed approach is based on the desire to emulate the real scheduling process and operations management. Scheduling simulation can be used as a tool aimed to support the planning activities, in order to maximise

customer satisfaction, minimise back orders, and reduce the production costs.

With more details, we propose to run the weekly plan (the timeframe mainly depends on the manufactured products) by a simulator of the production process, minimising the scheduling operation performed before simulation starting and introducing routines for job allocation during the simulation run. We call these procedures Delegate Functions (DFs).

On the basis of the weekly plan, operations are simulated since a situation of decision is detected. With respect to common scheduling approaches [Pinedo 2002], a decision is an assignment of a independent variable (according with the adopted scheduling approach it could be either a binary or multi-values variable).

When alternatives should be evaluated and compared, a specific DF runs in order to determine the best choice on the basis of the actual conditions of machineries and resources. The expected evolution of the production system is sometimes considered in order to fit medium term requirements and to avoid bottlenecks.

This is a key successful factor in scheduling simulation according with the well known issues due to continuous rescheduling and subsequent schedule instability [Herrmann et al.].

DFs are therefore suitable for short-medium term scheduling and most common dispatching: they play a key role in the optimisation of the Gantt and in the validation process of the medium-term plan.

DFs can be in different forms:

- algorithms belonging to Artificial Computation: a huge number of scientific papers that have been written highlight the synergy between simulation and Artificial Computation;
- Autonomous Agents: Agent-Directed Simulation is today considered a promising field [Oren 2001] and large interest has been shown in the simulation paradigm explicitly involving agents;
- any heuristic, expert or hybrid algorithm, simple or complex, with respect to the requirements of the specific application.

## WEEKLY PLANNING: A COMMON INDUSTRIAL FRAMEWORK

Largely adopted framework for manufacturing production planning involves three main stages: demand forecasting, master planning and weekly (or daily) rolling review of the plan [Peterson et al. 1998]. Demand forecasting [Beroldo et al. 2002] yields an aggregate prevision over a medium-long term with respect to the quantities of each product family to be likely requested. Forecasting is usually performed at multi-site level (MSL) and its output is provided to the MSL Master Production Scheduling (MPS) [Bernocco et al. 2003a]. Requested quantities for each single plant over a medium-long term are then used for determining the weekly master plan [Aloi et al. 2002]. It takes into account also the production plan, and other important considerations as backorders, availability of material, availability of capacity, management policy and goals [Proud 1999].

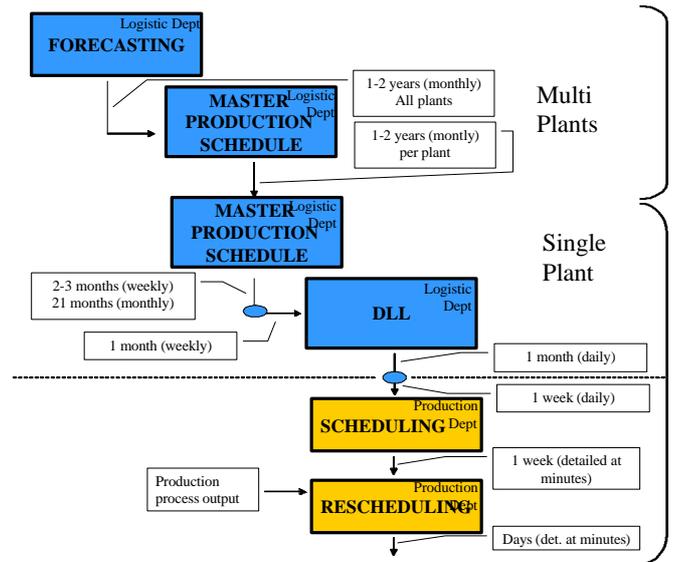


Figure 1: Common Planning Framework

## A CASE STUDY: BAXTER-BIEFFE MEDITAL

Baxter-Bieffe Medital production process for Clear-Flex bags is compounded by four phases: solution mixing, bag filling, sterilisation and packing. Figure 2 offers a simplified view of the departments, where mixing, filling and sterilisation are performed. They are: i) the tank room - phase 1; ii) the filling machines - phase 2; iii) the sterilisation and the unloading stations - phase 3.

Even if the production rate over each phase depends on the size, the vessels can be considered as the bottleneck of the whole process almost for every code.

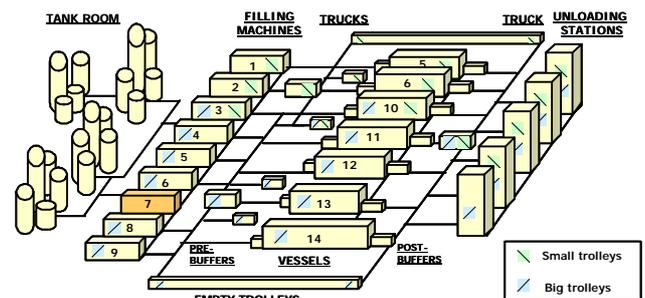


Figure 2: Schematic Baxter Planning Framework

### Planning Process “As Is”

From a single-plant prospective, planners receive a list of codes (i.e. products) to be produced, detailed in terms of inventory position (i.e. a measure of the product demand that can be satisfied by actual stock, expressed in terms of weeks) and grouped by product size.

Since the production rate depends on the size to be manufactured, grouping by size leads to product clusters based on the corresponding manufacturing performance.

On the basis of the experience of the production personnel with respect to the sterilisation phase and the corresponding knowledge base collected over the years, an evaluation process takes place in order to roughly estimate the capacity of the process according to the codes to be manufactured. This is a sort of Rough-Cut Capacity Planning (RCCP) procedure [Lee Berry 1997]. In this way, a first, basic allocation of the (grouped) codes is derived in accordance with the capacity constraints of the bottleneck production phase. It is called campaign allocation, since a campaign is a job that prescribes the manufacturing of fixed size products.

### Scheduling Process “As Is”

Moreover, a scheduling process takes place in order to optimise the performance (i.e. respecting logistic priority based on stock levels and the total set-up time) over the filling phase. During this phase, production personnel considers the most relevant constraints at mixing level. The performance of the sterilisation phase has been previously considered by the RCCP.

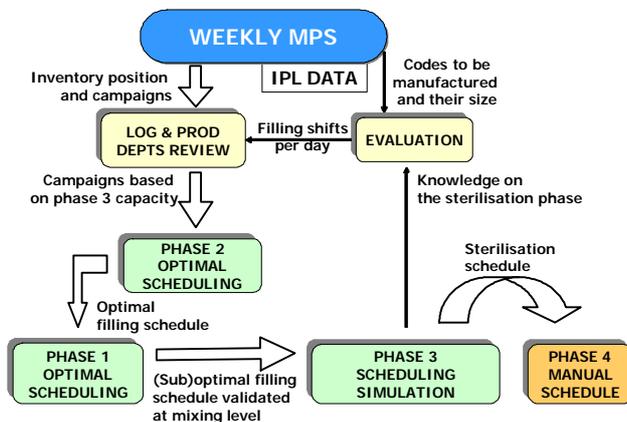


Figure 3: Schematic Baxter Planning Framework “To Be”

### Picture “To Be”

The goal of the re-organisation process and introduction of the system for daily line load can be summarised as follow: realise the automatic validation of weekly plan with high resolution and taking into consideration all the constraints. This result has been achieved by the implementation of a mixture of scheduling techniques and simulation.

A scheduling system based on both heuristics and combinatorial optimisation has been developed for phase 2 daily allocation. An hybrid scheduler, based on if-then rules and heuristics has been designed and implemented in order to perform the tank allocation. A scheduling simulator performs the allocation of the sterilisation tasks onto the vessels.

The weekly plan (expressed in terms of inventory positions and campaigns) is compared with the capacity constraint resulting from the evaluation of the production manager. The resulting new campaigns are then processed by the scheduler of phase 2 in order to determine the optimal job allocation over the filling phase.

The resulting schedule is used as input for the mixing

scheduler: it performs a capacity check backward allocating the formulas to be used by the filling machines.

In case of low capacity, the tank allocator shows the unfeasible codes and their position: in this way the planner can remove the corresponding job or modify its position in input of the filling scheduler. Then he re-runs the two scheduling systems until a feasible Gantt is reached.

Since the obtained sequence can be considered sub-optimal, with respect to phase 2 (adjustments due to phase 1 capacity constraints affect optimality of the Gantt), it only remains to validate the Gantt over the sterilisation phase.

Nevertheless this task is not trivial. It can be performed in two ways:

1. by common forward scheduling approaches;
2. by scheduling simulation.

Common scheduling approaches result not appropriate in this case since routing and trolleys selection should be rigorously taken into account and are dynamically changing. It means that the constraints cannot be stated at the beginning, but they largely depends on the specific allocation. Moreover if the system does not consider them the validation cannot be considered reliable and optimised; on the other side, no common scheduling paradigm exists that perform this evaluation in a changeable environment.

In this mind, a discrete event simulator that self-schedules the jobs has been designed, tested and implemented.

### SIMULATION OBJECTS

**Trolley:** the trolley is the product unit considered in the process. The goal of the production process is indeed to fill trolleys respecting the production schedule as most as possible, to sterilize them in the vessels, and finally to discard them, minimising wastes of time. The trolley can be of two types: small or big. Each trolley can contain a different number of bags, depending on the type of the bag and the dimensions of the trolley.

**Bag:** it is not really an object in the simulator. The bags are identified by a trolley, which has an attribute reporting the ids of the corresponding bags. The most important features of a bag are the cycle time (i.e. the production rate in terms of second per item) and the batch number.

**Filling Machines:** it is the machine which fills the trolley with bags. Each machine has a Gantt which contains information about the daily or weekly production. Loading a trolley, the filling machine print on it some relevant data:

- the batch number;
- the cycle time;
- the run number (which is a progressive number different for each filling machine).

**Vessel:** it is the machine which sterilizes the trolleys. Each vessel can sterilize only certain trolleys, depending on their size and on their cycle reference. The vessel receives the trolleys from its pre-buffer, and after the sterilization job is completed, it discards the trolleys on a post-buffer.

**Unload station:** it is a station where human resources unloads the trolleys. When a trolley is discarded, it is brought to the return-rail.

**Pre-buffer:** it is a pre-buffer which collects trolleys which can be sterilized together until the sterilization batch is complete. It is a FIFO queue.

**Post-buffer:** it is the post-buffer which contains the trolleys sterilized until it is discarded. It is a FIFO queue.

**Off-line buffer:** it is a buffer which can contain trolleys after they have been produced, before they are brought to a pre-buffer, or empty trolley where the return-rail is near full or full. It is a LIFO queue.

**Return rail:** it is a buffer for empty trolleys. After they are discarded on the unload stations, they are brought to the return-rail. When they are needed, they are brought to filling machines. If the rail is full or near full, some of them are brought onto off-line buffers. It is a FIFO queue.

**Wagon:** it is used to move trolleys from the return rails to filling machines or to off-line buffers, or from filling machines to pre-buffers or off-line buffers.

**Discard-wagon:** it is used to move trolleys from the post-buffers to return-rails or to unload stations, or from unload station to return-rails.

## BAXTER SCHEDULING SIMULATOR

In case of Baxter-Bieffe Medital production process, seven main decision should be taken. We summarise them by the questions that rise during the simulation (Table 1).

Table 1. Decision Points in Baxter Scheduling Simulation

Question/decision	Class
Pre-buffer or off-line buffer?	Optimisation
Which vessel?	Routing
Minimal number of empty trolleys?	Inventory Mngm
Incomplete sterilisation batch?	Routing
Small or big trolley?	Assignment
Which post buffer?	Routing
Which vessel with respect to the ending time?	Optimisation

The principle of the scheduling simulator developed for Baxter-Bieffe Medital is based on two kinds of decision-making procedures:

1. heuristics;
2. replication (or cloning).

The use of heuristics allows to take decision during the simulation. This is not novel [Bernocco et al. 2003], but it is integrated in a framework where simulation yields a Gantt (i.e. a schedule) for a phase of the production process. For instance, this is the case of the minimal number of empty trolley. In this case a heuristic typical of the inventory management problem has been successfully adopted, that is based on the principle of re-order point [Arnold and Chapman 2001].

On the other side, the replication is realised in those cases where no heuristics can support this decision since it would require prediction. Replication is indeed based on the

exploration of the alternatives in an exhaustive way. It is realised by cloning the simulator that generates the situation of decision. A simulation manager is thus introduced in the software architecture in order to implement this procedure. The concept of replication takes its origin from Branch and Bound [Pinedo 2002] and is here applied jointly with simulation in order to yield a scheduling process.

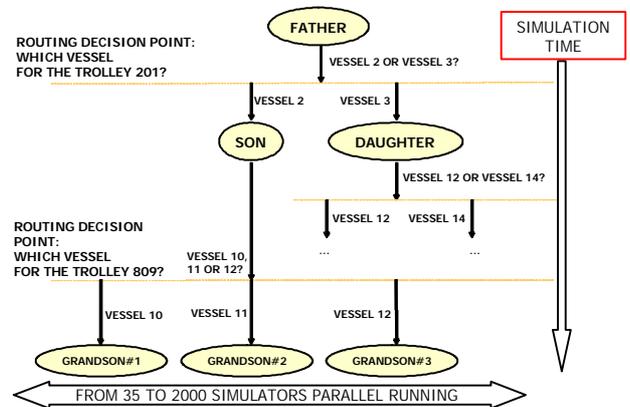


Figure 4: Replication Concept in Scheduling Simulation

## EXPONENTIAL GROWTH

The main problem we found using replication was the exponential growth of the number of the simulators. Indeed, considering that the average transportation time is almost 15 second, and considering that usually one third of the total movements is due to full trolleys towards a destination to be selected (i.e. a decision), we have a replication every 45 seconds for each wagon.

We had 2 wagons, so one duplication every 22,5 seconds. We can thus say that we have 8 duplication every 3 minutes. In this way we can estimate 160 duplications every hour. Considering that the number of destination varies from 1 to 5, we can further estimate that we had a replication of 2 simulators each time. So the number of plants in an hour could reach 2160!

For solving this problem we followed two ways:

1. the limitations of the plants growth;
2. the cutting of the search tree.

With respect to the limitations of the plants growth, by using heuristic and gating techniques we cut some alternatives that are unlikely to be promising.

With respect to the cutting of the tree, since the number of running simulator dramatically increases each time that a decision should be taken, we decide to bound the number of them by a threshold fixed a-priori. In this mind, when a decision point is reached and several plants are generated by replication, some simulator should be destroyed in order to not exceed the threshold.

The simulator to be eliminated have been selected by some metrics combined by a weighted sum in order to obtain a fitness function [Goldberg and Deb 2001].

They are:

- i) the number of trolleys sterilized;
- ii) the number of trolleys produced;

- iii) the number of trolleys discarded;
- iv) the number of working vessel;
- v) the number of empty trolley.

Nevertheless the rigorous application of this principle would result not effective since the simulator that have been just created do not significantly differ. As a result a second parameter should be introduced. This is the *minimal life time*, i.e. the minimal time during which a simulator cannot be destroyed even if its fitness is lower than others'. Notice that this further parameter lead to strongly exceed the threshold on the number of parallel simulators.

## RESULTS AND CONCLUSIONS

### Performance Evaluation

Fixing the threshold for the number of plants to 50 and the minimal life time to 1 minute, a real week can be simulated in 2-3 minutes, depending on the pc, with up to 2000 simulators parallel running. Table 2 and Figure 5 show the results obtained with a AMD1800 with 512Mb Ram. About 772184 events have been processed in average (5 runs).

Table 2. Time per Event, Including Replication

Event	Time [s]
EndProduction	111,59
EndDiscarding	22,11
GiveTrolley	17,84
GiveTrolleyToARail	9,70
GiveEmptyTrolleyToABuffer	5,44
<b>Total</b>	<b>182,95</b>

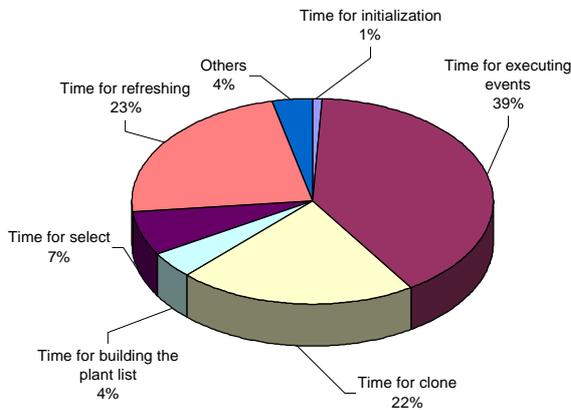


Figure 5: Times per Tasks [% of the Total CPU Time]

### Scheduling Performance

The implementation of the presented system (tank allocator, filling scheduler, scheduling simulator) led to successfully results.

The performance has been measured by:

1. the opinion of practitioners within the verification and validation process;
2. comparing the “as is” scheduling approach versus the scheduling simulator: the number of unloaded trolleys has been increased in average of +2.74 %;

3. comparing the whole planning system versus the “as is” planning and scheduling procedures (this process should be completed during the last months of 2003). Preliminary results showed an overall improvement of the performance that is between 5% and 10%.

Baxter is interested in the continuation and extension of this project Europe-wide.

## REFERENCES

- Arnold, T.J.F.; and Chapman, S.N.. 2001. *Introduction to Materials Management. Fourth Edition*. Prentice Hall International. Upper Saddle River. N.J.
- Bernocco, M.; Genta, M.; and Tonelli, F.. 2004. “Constrained Production & Procurement Planning for SCM by using Simulation and Top-Down analysis“. *Proceeding of 2003 Modeling and Applied Simulation Conference*. Savona. Italy
- Bernocco, M.; Queirolo, F.; Schenone, M.; and Tonelli, F.. 2003. “A Supervised Multi-Agent Approach for APS in multi-site production systems for demand validation and evaluation “ *Proceedings of SCSC '04*. Montreal. Canada.“
- Carson Y. and A. Maria. 1997. “Simulation optimization. Methods and applications“. *Proceedings of the 1997 Winter Simulation Conference*, 118-126.
- Goldberg D. E. and K. Deb. 1991. “A comparison of selection schemes used in genetic algorithms“. In *Foundations of Genetic Algorithms*, pages 69--93. Morgan Kaufmann, San Mateo, California.
- Herrmann, J.W.; Lin, E.; Vieira, G.E.. 2003. “Rescheduling manufacturing systems: a framework of strategies, policies, and methods“. *Journal of Scheduling*, Vol. 6, N. 1, pp 35-58.
- Lee Berry, W.; Vollmann, T. E.; and Whybark, D.C.. 1997. *Manufacturing planning and control system. Fourth Edition*. McGraw Hill. New York, NY.
- Peterson, R.; Pyke, D.F.; and Silver, E.A. 1998. *Inventory Management and Production Planning and Scheduling. Third Edition*. John Wiley and Sons. New York NY
- Pinedo, M. 2002. *Scheduling – Theory, Algorithms, and Systems. Second Edition*. Prentice Hall, Englewood Cliff, N. J.
- Proud, J.F.. 1999. *Master Scheduling. A practical guide to competitive manufacturing. Second Edition*. John Wiley and Son. New York. NY.

## AUTHORS' BIOGRAPHIES

**Pietro Giribone** is Full Professor of "Industrial Plants". Since 1979 he has specialized in experimental design applications for industrial simulators.

He is a member of the International Committee of the IASTED. He is part of the Board of Directors in ANIMP (National Construction & Engineering Association).

**Roberto Mosca** is full professor of "Industrial Plants Management" and "Economics and Business Organization". He is currently the head of Department of Production Engineering at University of Genoa. He has worked in the simulation field since 1969 developing interesting enhancements in the application of DOE.

**Filippo Queirolo** is a member of DIP Research Group at University of Genoa. He's research interest include Artificial Computation, Logistic, Multi Agent Systems, Operation Management, Simulation and Statistical learning. He is a member of SCS and founder of SACS.

# A SIMULATION CLIENT ACHIEVES HIGH SELF-SUFFICIENCY

Edward J. Williams  
Production Modeling Corporation  
Three Parklane Boulevard, Suite 1006 West  
Dearborn, Michigan 48126, United States  
E-mail: [ewilliams@pmcorp.com](mailto:ewilliams@pmcorp.com)

## KEYWORDS

Documentation, Model transfer, Software management, Simulation interfaces, System management.

## ABSTRACT

Simulation has become so well regarded that many businesses, in a variety of industries, now routinely realize its benefits; many others, those new to simulation, are eager to do so. Likewise, the use of simulation, long concentrated in the heavy manufacturing sector of the economy, has diversified into all sectors. Companies new to simulation often seek entry to this technology via retention of a consulting partner company already highly experienced and competent in its application. Too often, however, the company striving to incorporate simulation into its armoury of analytical and problem-solving tools becomes mired in dependency upon consultants indefinitely, even for what should be relatively routine modifications and extensions of the model originally constructed. In this paper is documented a successful, even rapid, emergence from such dependency – a client achieving self-sufficiency in simulation.

## INTRODUCTION

In the current case, industrial engineers at a rapidly expanding and strengthening pharmaceutical company had, by keeping abreast of both technical and business literature, noticed admiringly the increases in both productivity and efficiency often achieved via insights obtainable from discrete-process simulation analyses (Rohrer 1998). They and their managers became determined to introduce simulation technology as a routine policy into their own industrial and process engineering practices (Williams 1996). This determination specifically renounced as inadequate the mere receipt of a consultant's report and recommendations, or even the receipt of those plus a simulation model relegated to ornamental functions, as opposed to ongoing use, modification, and extension – a model whose recommendations would be implemented confidently (Scheeres 2003).

This paper will discuss first the forging of the relationship between the client pharmaceutical company and the consulting company. Next, it will describe the

specific steps taken to specify the scope of the initial study, choose an appropriate software tool, develop a prototype model, train the client engineers, and transfer the technology while removing all but the last vestiges of client dependency on consultancy.

## FORGING THE CLIENT-CONSULTANT RELATIONSHIP

The industrial engineers at the eventual client company, seeking a consulting partner in simulation analysis, began by assessing the suitability of various candidate consultants, with particular attention to availability of university liaisons, willingness to travel to the client's site, strong references, and willingness to *invest* (not *spend*) time in training, documentation, technology transfer, and data gathering as well as in modeling and analysis. Hence the client's engineers made heavy use of the advice on selection of a simulation-service vendor in (Williams 1993).

After the pharmaceutical company had chosen the consultancy company, managers and senior engineers of the two companies jointly constructed a contractual relationship emphasizing overall approaches, such as technology transfer directed to achievement of client self-sufficiency. Since this relationship cohered in a context of mutual trust, the discussion of myriad details was comfortably deferred to subsequent discussions, with the joint understanding that specifications of detail would be decided subsequently, as vigorously recommended by (McCormack 1984). The overall approach was confirmed to comprise a reconnoitering visit to the client site by a senior project engineer, assistance with choice and acquisition of software tools, construction of a prototype model, on-site training conducted by the same technical specialist who led the team effort of building this prototype, transfer of the model and knowledge of all techniques used in its construction to the client, and availability of brief consultations thereafter as needed.

## SPECIFYING SCOPE OF THE INITIAL STUDY

The senior project engineer spent four business days at the client's site to learn as much as possible about the client's products, procedures, and the operational and economic issues and improvement opportunities of greatest concern to client management. They reached agreement to focus project attention upon a blister

packaging line responsible for the packaging of allergy pills. This line already had a reputation as a bottleneck, and management, via marketing research, was confident production demands upon this line would increase over the next few quarters. This line receives as input individual pills of various varieties (for example, pills in some batches are time-release whereas pills in other batches are not). Furthermore, the line must package these pills in blister cards containing five, ten, or twenty pills as demanded by various customers. In contrast to the plastic pill bottles familiar to individual customers having a doctor's prescription filled at a pharmacy, blister cards hold pills encased in small plastic bubbles set against a lightweight cardboard backing. These cards are routinely used by hospitals, nursing homes, rehabilitation centers, and hospices. Such institutions must dispense large numbers of pills, keep them medically sanitary until ingestion by the patient under the watchful eye of a registered nurse, be able to quickly identify and investigate early evidence of medical problems plausibly connected with prescription regimens (such as adverse reactions between drugs, possibly prescribed by different doctors in different medical specialties), and be able to count pill inventories quickly and accurately on demand of a governmental audit (Cooper 1991).

Whenever production of this blister packaging line is changed from one stock-keeping unit [SKU] to another (and even packaging the very same pills into blister cards holding ten versus five pills each constitutes a change in SKU), significant changeover time overhead (typically four, six, or eight hours depending on the degree of dissimilarity between the outgoing and incoming SKU) must occur. These changeover times accommodate the absolute necessity of washing and checking all production equipment to ensure the most scrupulous cleanliness, and also to update and audit all production records as required by the Food and Drug Administration's [FDA] stringent policies backed by the United States government. Therefore, the client engineers were particularly interested in the ability of a simulation model to help them devise production schedules to minimize this overhead, yet not cause customers' orders to be delayed beyond their contractual delivery dates. Viewed thus, the challenges faced by the client engineers were job-sequencing problems analogous to those analyzed and attacked in (Mosca, Queirolo, and Tonelli 2002). A recent study (Johansson and Kaiser 2002) likewise illustrates the power of simulation when applied to such production challenges.

### **CHOICE OF SOFTWARE**

Together, the client engineers and the consultants examined four candidate computer software tools capable of building discrete-event simulation models with animation. Desiderata for the chosen tool, as enumerated in (Klingstam 2001) included ease of

learning and use, reasonable execution efficiency, ability to interface input and output operations with Microsoft® Excel spreadsheets, ample modeling power, ability to develop a model and its animation concurrently, and relatively low price (the last because the client's plans for eventual self-sufficiency included purchase of multiple copies of the chosen software) (Bowden 1998). Examples of criteria relegated to low importance were compatibility of the software with Unix operating systems (of which the client has none), availability of three-dimensional animation (the client's production systems contained no elevators, requirements for equipment clearance under bridge cranes, vertical storage systems, or other features inherently requiring three-dimensional animation for easy visualization), and high power to represent material-handling systems and equipment such as forklift trucks and automatic guided vehicles (not used at the client site). After comparing and contrasting the four candidates, engineers at the client and the consulting company jointly agreed on use of the SIMUL8® software package (Hauge and Paige 2001). The client engineers and their management then completed the purchase and installation of this software prior to the completion and delivery of a prototype model by the consultants.

### **DEVELOPMENT OF A PROTOTYPE MODEL**

The outline of process flow provided by the returning senior project engineer to the technical specialist readily sufficed for the construction of a prototype model. In the process, a thermoformer encases individual pills in plastic bubbles, originally softened by high heat and then mounted on a card. A cartoner places a specific number (which depends on SKU) of such cards into a carton. The carton is then sent to a check weigher which vets the weight of this carton. Cartons passing this check travel to a bander, which bands six cartons together; the resulting bundle then travels to a case packer which packs six cartons into a case. The case then travels to a labeler which affixes a product-identification and a shipping label to the case. Successive machines, represented as "work centers" in SIMUL8®, are joined by accumulating conveyors. Since the machines in actual production practice have dedicated operators, the client and consulting engineers decided separate modeling of labor for machine operation was unnecessary. Irrespective of SKU currently being produced, product moving on a particular conveyor is always of the same "footprint" size. A large-capacity SIMUL8® "storage" holds individual pills at the upstream end of this process; an acknowledged modeling assumption specified that this storage would never run out of pills. The operations run on a three-shift basis; each shift is eight hours long and includes a lunch break, shorter breaks, and provision for ten minutes' cleanup time at the beginning and at the end of the shift. A diagrammatic representation of this process appears in Figure 1 on the next page.

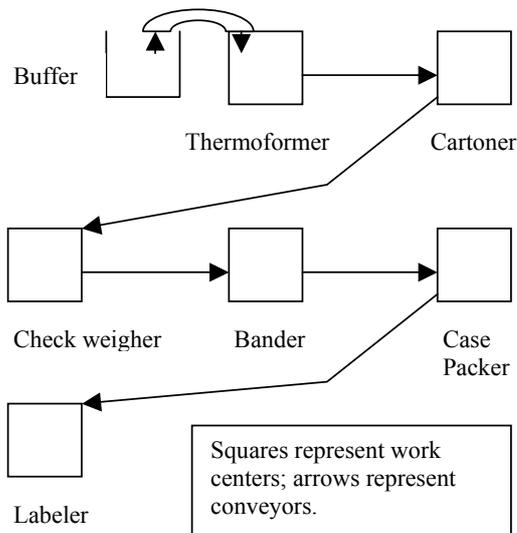


Figure 1: Basic production line configuration.

To make the model more convenient for the client's engineers to use, and to ease the impending technology transfer, as much model input information as possible was placed in a Microsoft® Excel workbook, with closely related data items grouped into distinct worksheets. For example, machine cycling rates were in one worksheet, changeover times in another, and shift patterns (e.g., lengths of break times and their placement within an eight-hour shift) were in yet another. Worksheet cells intended to be changed by the user were formatted with a green background to indicate "change permitted."

Throughout, this model was documented, especially internally, with special care, since the project plan explicitly specified that it would very soon be transferred to two audiences: process engineers about to be trained in simulation concepts, the SIMUL8® software tool, and the logical processes used internally to build the model, and derivatively to managers who would need to understand its operation and significance, but not its internal details. SIMUL8® supports internal commenting of a model in three ways: clicking a "Memo" button available in each of its basic constructs such as Work Center, Storage Bin, Work Entry Point, etc.; insertion of comments within Visual Logic code via a "Comment" code line (such comments then appear in green, as is the Visual Basic tradition); and inserting comments presumably pertinent to the model as a whole via the command File/Simulation Properties. All of these methods were used. Indeed, (Oscarsson and Moris 2002) have examined in detail the importance of documentation in such instances, and techniques for making the documentation effective to various audiences.

## TRAINING OF CLIENT ENGINEERS

After the prototype model was developed and verified (but *not* validated) by consulting engineers, it was electronically mailed to the client engineers. During the first on-site visit by the senior project engineer, he had taken care to explain the distinctions between verification and validation to the client engineers (Sargent 1996). One week later, allowing time for the client engineers to open the model on their newly acquired software and formulate basic questions about it, the technical specialist who led the effort of building and verifying it traveled to the client site for four days' training and consultation. Client management invested aggressively in this training, allocating five production engineers to it full-time and one additional production engineer to the first half of it.

The first two days of the training were essentially standardized, using a canonical curriculum which reviewed the conceptual foundations and basic methods of sound simulation practice in industry, and also examined all the essential functionality and fundamental constructs of SIMUL8®. This overview defined simulation, described the business and management motivations for using simulation (with illustrative applications), explained the functioning of the model clock (for example, it compresses time, and logically must never attempt to run in reverse), enumerated typically required inputs and available outputs, and discussed in detail a list of no fewer than fifteen frequent mistakes and their avoidance. Next, the overview instruction explained statistical concepts involved in discrete-process simulation, including the contrast between continuous and discrete distributions, the contrast between empirical and closed-form (theoretical) distributions, the importance of replication length, number of replications, and the choice of warmup time (including when warmup time should equal zero because the simulation is terminating, not steady-state).

Next, the canonical training discussed SIMUL8® usage thoroughly, including use of its basic constructs, incoming and outgoing routing options (of which there are many, particularly with reference to Work Centers), specification of travel times within the model, representation of resources and their travel times, use of Labels (called "Attributes" in almost all other simulation software) and of Information Stores (called "Variables" in almost all other simulation software), coding and use of Visual Logic triggered by various events in the simulated system (e.g., beginning of cycle, beginning of downtime, end of cycle, end of downtime, end of run, etc.), establishment of run parameters (run length, warmup length), customizing of the results report, and interpretation of results.

Promptly at 8am on the third day, the training leader opened the SIMUL8® model and the Microsoft® Excel workbooks upon which it depended for input and output. Each element in the model, whether work entry point, work center, conveyor, resource, storage, or work exit point, was examined in turn, and the purpose and construction of each line of underlying Visual Logic (SIMUL8®'s internal programming-logic language) was examined. The trainer "reconstructed the model out loud," thereby conveying to the client's process engineers the thought processes used to construct it. Examples of questions asked and answered, and issues discussed, during this phase of the training devoted to "model transfer to client," were:

1. Why is the Routing In mode of this Work Center specified as "Passive"?
2. Why is this segment of Visual Logic code placed in "On Work Complete" instead of "On Exit" relative to the Work Center involved?
3. How can we most conveniently rearrange the results report depending on whether we would like Work Center utilizations sorted alphabetically, in upstream-to-downstream order, or by decreasing utilization?
4. In model logic, how do we distinguish between an urgent item which is allowed to go to the front of a queue and an even more urgent item which is allowed to interrupt work currently in progress at a Work Center?
5. If at some future time we would like to assign a technician to be responsible to repair several different machines, how would we use the Resource construct of SIMUL8® to represent the situation accurately?

Next, the process engineers guided the training leader in the validation of the model, describing the errors which, given the project approach thus far (e.g., no member of the model development team had visited the client site prior to the current week), were inevitable and had been acknowledged as such. Examples of these errors were:

1. The batching portion of the thermoformer cycle was completely missing.
2. The unfortunate ability of some machines to produce occasional scrap had been overlooked.
3. Changeover times for switching from some SKUs to other SKUs were incorrectly specified.
4. Conveyors were the wrong lengths.
5. Intervals between downtimes were cycle-based on the production floor (for example, after a certain number of cycles, the bander would exhaust its supply of packaging tape), but time-based in the model.

The trainer and the process engineers then discussed the most effective approach to repair of each error. Most of this discussion took place on the production floor, as the process engineers provided a planned and extremely valuable guided tour to the trainer. For

example, error #3 was corrected within the pertinent input Excel® worksheet (an example of the attraction of using them for input flexibility). Errors #4 and #5 were readily corrected by revision of the conveyors and the work centers respectively. For example, correction of Error #5 required changing work center downtime specification from the SIMUL8® "Auto" choice to the SIMUL8® "Detailed" choice. In the former, only the uptime percent and the average repair time are specified; in the latter, the modeler has complete control over the mean-time-to-fail and mean-time-to-repair distributions, including specification of MTTF as clock-time based, busy-time based, or cycle-based. Error #5 also required revision and reinterpretation of the Excel® input: a number previously interpreted as "mean time to fail" was changed in value and reinterpreted as "mean cycles to fail." The first two errors in the list above required more fundamental changes to the model, such as adding constructs (e.g., work exit points representing scrap leaving the system) and/or revising Visual Logic within the system.

On the fourth and last day of the on-site training and consultation, the process engineers and the trainer worked together to implement corrections of various errors, including those described above. This experience gave the engineers useful guided experience in using the software, plus experience in the techniques of model step-by-step tracing, examination of the animation, and desk-checking of output to detect the presence of errors, locate their source, and implement corrections without the introduction of new errors. Likewise, the engineers also learned, via directly "hands-on" practice, to define and run experiments with appropriate warm-up times and number of replications. When it was "time to leave for the airport" late in the afternoon, all but four of the identified errors had been corrected, and a method of correcting those remaining had been agreed upon.

## **TRANSFERRING THE TECHNOLOGY**

After the trainer's return to the consultant's home office, both he and the senior project engineer checked frequently with the client engineers via telephone and electronic mail. Three months after the on-site training and collaborative model validation, the client engineers had corrected the remaining errors, completed model validation, and were exercising the model vigorously for ongoing experimentation. Most, but not all, of these experiments were run by making changes to the input data within the Microsoft® Excel workbook; a minority were run via changes to the model itself. As an example of this contrast, during the trainer's site visit, changeover times were indexed into four broad product categories, whose specific changeover times were read into the model from the Microsoft Excel® workbook. Revised changeover times were then accommodated by routine worksheet changes. However, at one point, the client engineers realized they would need to add a fifth

broad product category. Via email advice, they were able to modify the loop constructs within the pertinent Visual Logic code to accommodate this revision. Since the client's engineers needed less than ½ person-day, in aggregate, from the simulation consultants during this three-month period, both parties happily deemed the technology transfer a success and the original project plan worthy of reuse in potential future projects.

### CLIENT BENEFITS REALIZED

The client organization has realized significant benefits accruing from ongoing use of this simulation model. Among the most significant of these benefits are the following:

1. Accurate predictions of machine utilization under a variety of different scheduling scenarios;
2. Accurate predictions of conveyor utilization and occupancy under a variety of different scheduling scenarios;
3. Greatly improved (both in accuracy and "distance to time horizon") capacity planning, analogous to that reported for a microbrewery in (Bergin, Davidoff, and Weston 2002);
4. Increased awareness of the importance of reducing setup times by assigning and scheduling tasks concurrently whenever possible, coupled with new availability of quantitative assessment of these benefits in achieving leaner manufacturing, as espoused in (Parks 2003);
5. Realization and quantitative proof that scheduling to make long changeover times fall near the beginning or end of a shift, versus near the middle of a shift, yields both more easily implementable personnel assignment schedules and increased throughput;
6. Development of preventive maintenance schedules, using simulation, to best interface with production schedules and their inherent changeovers, as achieved in (Alfares 2002) for analogous production lines packaging powdered detergent, liquid soaps, and shampoos;
7. Valuable long-term understanding of the steps (such as rigorous data collection and development of a simulation usage strategy) required to integrate discrete-event simulation into the engineering process, as extensively documented in (Holst 2001).

### SUMMARY AND CONCLUSIONS

This paper has described a simulation study undertaken with the specific goal of client self-sufficiency in the technology defined as having equal rank with other typical goals such as quantifiable improvements in process efficiency. The project plans defined during the forging of the client-consultant relationship proved adequate to reach this goal, and hence will be reused. The client has realized significant and quantifiable benefits from this study.

### ACKNOWLEDGMENTS

Professor Onur M. Ülgen, University of Michigan – Dearborn, and President, Production Modeling Corporation, Dearborn, Michigan, Chris DeWitt, Project Manager, Production Modeling Corporation, and two anonymous referees have provided valuable inspiration for, commentary on, and constructive criticisms to improve this paper.

### REFERENCES

- Alfares, Hesham K. 2002. "Developing Optimum Preventive maintenance Schedules Using Simulation: A Case Study." *International Journal of Industrial Engineering – Theory, Applications, and Practice* 9(3):311-318.
- Bergin, D., Peter H. Davidoff, and F. C. "Ted" Weston Jr. 2002. "The Right Place for a Bottleneck." *IIE Solutions* 34(12):34-39.
- Bowden, Royce. 1998. "The Spectrum of Simulation Software." *Industrial Engineering Solutions* 30(5):44-46.
- Cooper, James W. 1991. *Drug-Related Problems in Geriatric Nursing Home Patients*. Binghamton, New York: The Haworth Press, Incorporated.
- Hauge, Jaret W., and Kerrie N. Paige. 2001. *Learning SIMUL8: The Complete Guide*. Bellingham, Washington: PlainVu Publishers.
- Holst, Lars. 2001. *Integrating Discrete-Event Simulation into the Manufacturing System Development Process: A Methodological Framework*. Licentiate in Engineering Thesis, Division of Robotics, Department of Mechanical Engineering, Lund University, Lund, Sverige.
- McCormack, Mark H. 1984. *What They Don't Teach You at Harvard Business School*. Toronto, Ontario: Bantam Books, Incorporated.
- Johansson, Björn, and Jürgen Kaiser. 2002. "Turn Lost Production into Profit – Discrete Event Simulation Applied on Resetting Performance in Manufacturing Systems. In *Proceedings of the 2002 Winter Simulation Conference*, Volume 2, eds. Enver Yücesan, Chun-Hung Chen, Jane L. Snowdon, and John M. Charnes, 1065-1072.
- Klingstam, Pär. 2001. *Integrating Discrete Event Simulation into the Engineering Process: Strategic Solutions for Increased Efficiency in Industrial System Development*. Thesis for Degree of Doctor of Philosophy, Department of Production Engineering, Chalmers University of Technology, Göteborg, Sverige.
- Mosca, Roberto, Filippo Queirolo, and Flavio Tonelli. 2003. "Job Sequencing Problem in a Semi-Automataic Production Process." In *Proceedings of the 14<sup>th</sup> European Simulation Symposium*, eds. Alexander Verbraeck and Wilfred Krug, 343-347.
- Oscarsson, Jan, and Matías Urenda Moris. 2002. "Documentation of Discrete Event Simulation Models for Manufacturing System Life Cycle Simulation." In *Proceedings of the 2002 Winter Simulation Conference*, Volume 2, eds. Enver Yücesan, Chun-Hung Chen, Jane L. Snowdon, and John M. Charnes, 1073-1078.
- Parks, Charles M. 2003. "The Bare Necessities of Lean." *Industrial Engineer* 35(8):39-42.

- Rohrer, Matthew W. 1998. "Simulation of Manufacturing and Material Handling Systems." In *Handbook of Simulation*, ed. Jerry Banks, 519-545. New York, New York: John Wiley & Sons, Incorporated.
- Sargent, Robert G. 1996. "Verifying and Validating Simulation Models." In *Proceedings of the 1996 Winter Simulation Conference*, eds. John M. Charnes, Douglas M. Morrice, Daniel T. Brunner, and James J. Swain, 55-64.
- Scheeres, Junell. 2003. "Making Simulation a Reality." *Industrial Engineer* 35(2):46-48.
- Williams, Edward J. 1993. "Selection of a Simulation-Service Vendor." *Industrial Engineering* 25(11):18-19.
- Williams, Edward J. 1996. "Making Simulation a Corporate Norm." In *Proceedings of the 1996 Summer Computer Simulation Conference*, eds. V. Wayne Ingalls, Joseph Cynamon, and Annie V. Saylor, 627-632.

## AUTHOR BIOGRAPHY



**EDWARD J. WILLIAMS** holds bachelor's and master's degrees in mathematics (Michigan State University, 1967; University of Wisconsin, 1968). From 1969 to 1971, he did statistical programming and analysis of biomedical data at Walter Reed Army Hospital, Washington, D.C. He joined Ford Motor Company in 1972, where he worked until retirement in December 2001 as a computer software analyst supporting statistical and simulation software. After retirement from Ford, he joined Production Modeling Corporation, Dearborn, Michigan, as a senior simulation analyst. Also, since 1980, he has taught evening classes at the University of Michigan, including both undergraduate and graduate simulation classes using GPSS/H™, SLAM II™, SIMAN™, ProModel®, SIMUL8®, or Arena®. He is a member of the Institute of Industrial Engineers [IIE], the Society for Computer Simulation International [SCS], and the Michigan Simulation Users' Group [MSUG]. He serves on the editorial board of the *International Journal of Industrial Engineering – Applications and Practice*. During the last several years, he has given invited plenary addresses on simulation and statistics at conferences in Monterrey, México; İstanbul, Turkey; Genova, Italy; and Rīga, Latvia.

# PROPOSAL OF A FRAMEWORK FOR PRODUCTION PLANTS REMOTE CONTROL: A PRELIMINARY TEST CASE

Romeo Bandinelli, Mario Rapaccini  
romeo.bandinelli@siti.de.unifi.it,  
rapaccini@ing.unifi.it  
Università di Firenze  
Dip. di Energetica "Sergio Stecco"  
Sez. Impianti e Tecnologie Industriali  
Via Cesare Lombroso 6/17  
50134 Firenze  
Italy  
tel +39-055-4796722  
fax +39-055-4224137

Sergio Terzi, Marco Macchi  
sergio.terzi(marco.macchi)@polimi.it,  
Politecnico di Milano  
Department of Economics, Industrial and  
Management Engineering  
Piazza Leonardo da Vinci 32  
20133 Milano  
Italy  
tel +39-02-23992803  
fax +39-02-23992700

## KEYWORDS

Parallel and Distributed Simulation, High Level Architecture, Inter Process Communication, Remote Factory

## ABSTRACT

This paper illustrates a proposal of a framework for production plants remote control. The architecture has been developed under HLA-RTI environment, with the use of *<next event>* as paradigm for time management. XML has been used as the formalism for both information coding and non-persistent data-structuring, while persistent objects are represented as HLA objects. The proposed framework can be used in conjunction with any COOTS simulator software, and has been tested in a local environment with Simple++ simulator software.

## 1 INTRODUCTION

The continuous increase of ICT applications is causing a radical change within the manufacturing industry. The effects on structures and processes are visible to everybody, for either world-wide enterprises or local industries. Summarising, this process is determined by some common issues: globalization (activities have to be managed with reference to a global environment), interconnection (coordination is possible through structured communications among remote groups) and e-manufacturing (industrial processes are computer-based controlled). Technological drivers occurring in this development are focusing significant investments, especially where headquarters and productive sites are spread in a wide territory, thus having the necessity for a narrow integration. In such a distributed environment, the main decision management process might be deployed in a distributed way, since to preserving the independence of each actor from one side, but also in order to provide a coherent creation of

value. Within this distributed scenario, tools for performance analysis and supply-chain processes design/management are specifically required.

With reference to the mentioned issues, the paper aims to illustrate a proposal of a framework for production plants remote control to be adopted for the distributed management and coordination of productive nodes; in particular, this objective is achieved by the use of web-based distributed simulation. In order to provide a coherent presentation of the work, the paper will be organized as follows: § 2 introduces the research idea where the framework was developed and how the work was conducted; § 3 analyses the available technologies adopted into the proposed framework and illustrates the proposed framework; § 4 describes the preliminary test case developed according to the proposed framework; § 5 reports some conclusions and highlights further developments.

## 2 THE RESEARCH IDEA

The present paper aims to illustrate a preliminary research work conducted in the area of remote factory control adopting a distributed simulation approach. This work was elaborated thanks to the contributions and the knowledge of the international research group of fourth *Special Interest Group* (SIG4) of the IMS-NoE community [8], specifically interested in the establishment of a remote scheduling factory control. Within the SIG4 community, the whole research idea is generally named *Remote Factory* project.

The main idea (*Fig. 1*) of the *Remote Factory* project deals with the establishment of a virtual arena, physically provided by web-based and parallel and distributed technologies, where one industrial plant could be emulated/simulated in terms of its physical resources, while the PP&C logics are reproduced in a detached environment. Thanks to this separation and to simulation technologies, over an emulated plant could be executed and tested more and more PP&C logics, in

order to identify the best solution using a kind of a benchmarking approach. In such a way, PP&C experts of the enterprise headquarter could be able to identify *a priori* PP&C solutions for each industrial plant and for the whole SC, avoiding inefficient local decisions<sup>1</sup>.

The present paper, in particular, concerns with a preliminary research work freely carried on by two SIG4 members in order to investigate one efficient solution for the adoption of the distributed simulation approach: University of Florence (UNIFI) and Politecnico di Milano (POLIMI).

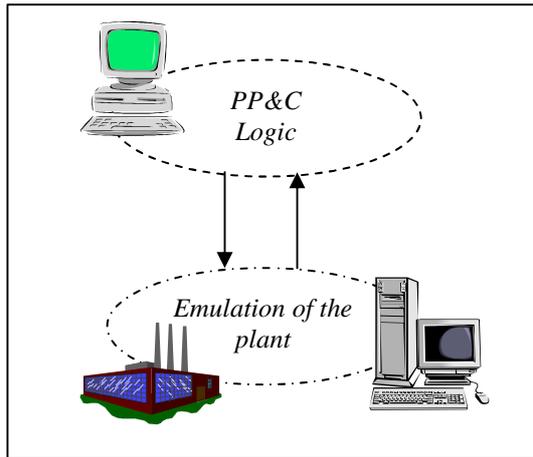


Figure 1 – Remote Factory idea

The developed distributed architecture was tested on a simple industrial test case, where one industrial plant was emulated in a physical model (PM) connected, in a distributed simulated environment, to a logical control model (LM), based on the protocols-negotiation multi-agent logic by Solberg and Lin [9].

### 3 THE PROPOSED ARCHITECTURE

As mentioned, the *Remote Factory* idea deals with the adoption of a PDS (*Parallel and Distributed Simulation*) environment, where two separated models (PM and LM) could interact.

#### 3.1 Requirements for an architecture for remote factory control

Establishing a remote factory control means to apply a remote management over a production plant, managing PP&C decisions. Therefore, a remote control means, at first, to define which kind of information might flow from one *Control System* (reproduced into a logical model - LM) to the *Production Plant* (emulated in a simulation environment) and, if it is needed, vice versa

<sup>1</sup> The *Remote Factory* idea deals also with problems different from the remote headquarter control; in particular, at the present the main interest of the *Remote Factory* project (and of the SIG4 group) deals with the establishment of such a benchmarking service for solving the dichotomy which afflicts the world of scheduling research, where PP&C experts are totally detached from industrial reality; other information about this could be read in [8].

(figure 2).

Physically, a control system defines, using its internal rules and logics, which kind of *production plan* might be performed by resources of the plant (e.g. work-centres, docks, lines) in order to satisfy a due performance (e.g. due-date timing). Work of the control system is to define job scheduling (sequencing, loading, dispatching), communicating its taken decisions to plant resources in terms of *Tasks* to be performed (e.g. “Job X in machine Y, for Z time-units”). Defined *Tasks* can be communicated one or more times, depending to the scheduling system ontology. In fact, traditional scheduling tools elaborate only one general *production plan* for a due production time period (e.g. for one shift, or one day). On the contrary, advanced scheduling solutions (e.g. *Multi-Agent Systems MAS* – see par. 4, or *Genetic Algorithm GA*) try to continuously elaborate a new *production plan* following what happens into the plant, re-defining scheduling *tasks*.

By the production plant side, a different kind of information might be achieved, corresponding to the resources status description. This information is required for setting up the scheduling algorithms of the traditional tools, while the most advanced solutions (e.g. MAS) need it continuously, in order to follow up production plant history.

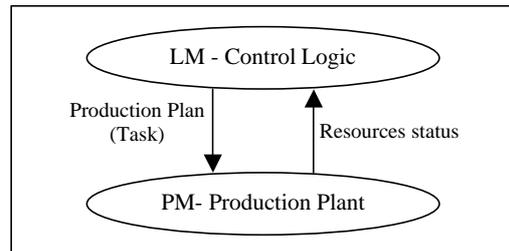


Figure 2 – Remote control information exchange

This way, the needed architecture for remote factory control might:

- (i) enable a distributed (simulation) environment for remote management,
- (ii) consider the different kind of information flows,
- (iii) provide to each model (PM and LM) the requested elements (in terms of IN and OUT flows),
- (iv) be able to manage diverse sort of models in terms of plant dimensions (number of resources), for PM, and scheduling ontology, for LM.

In the next paragraph, the proposed architecture will be illustrated in terms of technological solutions.

#### 3.2 Technical foundations of the architecture

The needed PDS distributed environment was identified in HLA (*High Level Architecture*, [12]). As known, HLA is the most important PDS framework, recently defined as a IEEE standard, that was originally developed by the U.S. Department of Defence for

military purposes. Within the HLA framework, a distributed simulation is accomplished through a “federation” of concurrent “federates” (distributed models), interacting between themselves by means of a shared data model, specified in a proprietary language (OMT - *Object Modelling Template*) and federation services (basically time and data distribution management services). The federation services are provided by the *Run Time Infrastructure* (RTI) software tool, based on the HLA interface specifications. HLA has been chosen instead of other framework like CORBA [1], RMI [7] or DEVS [3] because of its robustness and the mature time management approach. Moreover, HLA has been adopted by UNIFI and POLIMI in a previous project [10]. Simple++ [5] has been chosen as simulation software, either for the physical model than for the logical one. Simple++ has been chosen because of its diffusion among COOTS (*Commercial Off Of The Shelves*) simulation tools, representing a typical environment that a future Remote factory user could adopt, with a complete support for object oriented programming and a user-friendly interface.

As known, nowadays totally HLA-compliant simulator’s commercial software doesn’t exist. So, it’s not advisable to define an architecture where a totally HLA-compliant simulator is needed, because this choice would force the use of a specific simulator written in C++ or Java, and not a commercial tool. For these reasons the introduction of a component between the simulator and HLA environment was needed. The realization of this add-on could be done according two ways. The first one can be summarized in the definition of a *Delegated Simulator* module. This module is responsible for all the logic of information exchange between federates. The second solution proposes the introduction of a software “living” between the simulator and the RTI. This software, called *Proxy*, has the responsibility to guarantee the communication between the RTI environment and the simulator, and vice versa<sup>2</sup>. For this work, the second way has been chosen, with the use of a *Proxy*, thanks to the flexibility that it provides. The *Proxy*, written in java, was responsible for the information exchange between the simulator and the RTI. While the simulator has a synchronous way to communicate by TCP-IP, RTI has an asynchronous way: the *Proxy* had to store information coming from RTI and transmit it to the simulator as soon as possible and vice versa.

A clear separation from information regarding persistent objects (i.e. SM work centres’ state) and not-persistent objects (i.e. production plan) has been done. While the firsts have been implemented as RTI objects, instantiated at the beginning of the simulation and destroyed at the end, the second one has been developed as HLA interactions. This choice allows an

easier management of the time, with an improvement of performances in comparison to an architecture without interactions and a correct time-sequence information exchange.

Moreover, this proposal aims to differentiate the communications from the PM to the LM and vice versa. In fact, while the firsts ones have been transmitted as specific values of the HLA objects’ attributes, the second ones have been implemented with HLA interactions, with the use of XLM as the formalism for both information coding and not-persistent objects data-structuring. In the proposed architecture, XML is used in order to communicate the production plan (from the controller to the plant), production executions and statistics (from the plant to the controller). For this reason, a C++ library for coding and encoding XML strings has been written and loaded into Simple++.

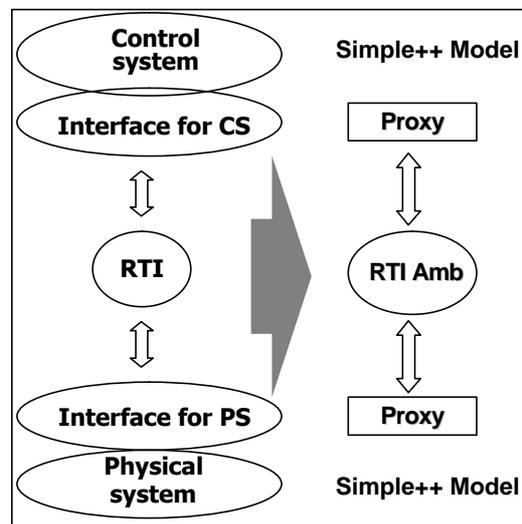


Figure 3 – Overall vision

An overall vision of the architecture is summarized in figure 3.

The XML schema used in order to communicate *production plan* is reported in figure 4.

```
<?xml version="1.0" standalone="yes"?>
<task>
  <load>
    <lot_ID>1</lot_ID>
    <processor_ID>54CE</processor_ID >
    <start_time>1:00:00.0000</start_time >
    <duration>10:00:00.0000</duration>
    <job_ID>8</job_ID>
  </load>
  ...
</task>
```

Figure 4 – XML schema for production plan

The use of XML inside an HLA environment extends generality of contents of messages; moreover, adding more lot-related information would be very easy.

<sup>2</sup> More information about “Delegate Simulator” and “Proxy” can be found in [11]

## 4 THE TEST CASE

As a test-case, we adopted an advanced-scheduling MAS solution, remotely controlling a shop-floor. Thus, the test-case was composed by two main components: (i) the shop-floor plant simulation, and (ii) the shop-floor MAS control logic, implemented with Simple++.

This architecture will be described starting from the shop-floor, and dividing it into three areas: (i) the shop-floor structure, (ii) the shop-floor control flows and (iii) the shop-floor control execution.

### 4.1 The shop-floor structure

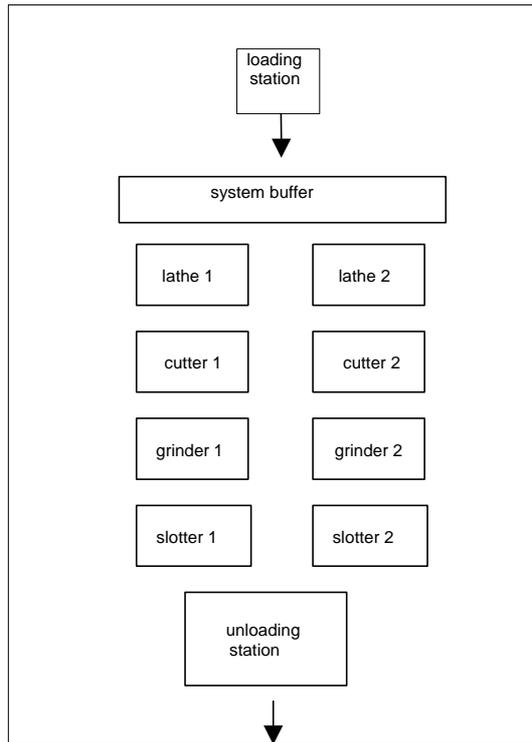


Figure 5 – Shop floor structure

The shop-floor is composed by 8 stations, grouped two by two. Each 4-station lay-out is dedicated to a specific process, as depicted in Fig. 5.

In order to implement reusable and modular architecture, some general-purpose classes were developed within the PM. In fact, the class *Processor* can represent four different objects:

- a buffer in,
- a buffer out,
- a native processor, able to describe any real work-centres,
- a data storage, able to store information about shop-orders execution.

This class was used to implement all the resources available in the PM of the described test-case, as well as the “*system buffer*”, that is a virtual buffer where inactive jobs are moved. Other classes were used in order to generate entities (*Job* and *Part*) and manage material flows.

### 4.2 The shop-floor control flows

The PM is able to receive a *production plan* and to storage it in the data storages of the work-centres. Then, each processor analyzes the work-order according to a FIFO logic. Moreover, statistical information about job completion is stored to be transmitted later to the LM.

### 4.3 The shop-floor control execution

According to the proposed architecture, two data flows have been identified: the tasks flow from the PM to the LM, and the PM system *events* (i.e. resource status) from the LM to the PM.

Any occurrences relevant to production scheduling (i.e. set-up completed, start of material loading, end of processing, breakdown/failures of work-centre, maintenance beginning, restoring time) were stored as an “event”. This information were then required from the LM, in order to either re-schedule or confirm the *production plan*.

In order both to minimize clock stop and to avoid inconsistency states of the PM, the information transmitted during a simulation run has been minimized. All the information used for statistics analysis were stored into the PM and communicated to the LM at the end of each run, while all the productions parameters needed for scheduling were recorded into the logical model. The flow from the PM to the LM contains only the necessary information of the jobs, while the other flow contains only the update state of the status’s attribute of each machinery.

### 4.4 Example of execution

At the beginning of the simulation run, the proxy creates a number of HLA objects of type “processor” equal to the number presented in the physical model. During the simulation run, HLA objects are synchronized with the real state of the PM by the proxy. This is done by a synchronous socket between the PM and the proxy. Every time a job is executed, the simulation clock is stopped, and socket is opened in order to communicate the new state of the objects. Then, proxy communicates this updating to the other federators with HLA *<next event>* time-management logic. A typical step of the execution’s process could be described as follows.

Every time a work activity is completed, an event that stops the simulation clock occurs. The updated state of the PM is communicated to the Proxy, that stores it temporary, and then updates the RTI environment. Since the controller subscribed the needed objects at the beginning of the simulation run, consequently RTI delivers information to the LM. The LM achieves all the information to start the negotiation for the successive working for the job that caused the stop event. The scheduling process can be activate also by a fictitious event generated by the PM, in order to simulate the dynamic of the end of a scheduling

process. This is done with the introduction of a system class (system processor), that is programmed in order to generate an event every time a negotiation's process ends in the LM (depending on the contract net logics).

When these events occur, the PM causes the simulation clock to stop, and this operation permits the PM to receive the results of the scheduling process (task). If the negotiation process finishes without the assignment of any task, the LM communicates, as a result, another fictitious event, that will occur when the next scheduling process finishes. The tasks, that represent the job's order, are communicated to the PM with the use of RTI interaction (communication class), where the attributes ( type of working, job's identification, working center's identification, beginning time of the working, working's length) are stored as an XML string. The presence of the system processor solve also a logic's weakness: during the simulation run, it's possible to have the PM completely free from processing. In this state the PM doesn't generate any event, so the simulation clock would be never stopped, in order to receive working order: in this case the fictitious event solves the problem.

The proposed architecture is based on discrete-event distributed simulation, using HLA <NextEvent> paradigm for time management, where an unimportant <LookAhead> value is associated to the LM, while a <LookAhead> proportional to the predicted time for the sheduling process is associated to the LM. As a consequence, the PM is in time advance if compared with the global simulation clock (federation time), this coinciding with the LM clock. As already said, the run is stopped when any system event happens, and status changings are published by the PM. Then, an authorization to go until the next event is requested to the RTI. RTI gives the authorization after all the messages have been delivered to the interested federates. With this logic, the PM will not stop again until the successive event, so it'll not be able to receive others tasks in the meanwhile. During this time, LM can be:

- waiting for the following event, so waiting to go on after the last production orders have been executed;
- waiting for publishing production orders the PM is going to receive.

If LM is in the state 2), the publication of the production plan will be causing LM to go on and to reach state 1). For both the states, when the LM's clock will start again, a new production's plan will be elaborated as a consequence of the PM status-changing.

Figure 6 shows the way an XML string is transmitted and the way the proxy is able to manage the <next event> HLA command. Figure 7 shows a log made by Simple++ about the PM production plant.

```

Proxy 92
ID_processore><Istante_di_inizio_lavorazione>18
e><Durata>258</Durata></ID_operazione>2</ID_oper
Sent Interaction @ 17091.0
NextEvent TimeAdvanceGranted @ 17091.0
CheckUpdated @ 17091.0
CheckUpdated @ 17091.0
CheckUpdated @ 17091.0
Interaction to send : Communication
  Attributo: Message
  Valore: <?xml version="1.0" standalone="yes"?>
  trollore.GestLotto51</ID_lotto><ID_processore>
  cessore><Istante_di_inizio_lavorazione>17804</I
  ata>510</Durata><ID_operazione>4</ID_operazione
  Sent Interaction @ 17091.0
Interaction to send : Communication
  Attributo: Message
  Valore: <?xml version="1.0" standalone="yes"?>
  trollore.GestLotto23</ID_lotto><ID_processore>

```

Figure 6 – Output of the proxy during the simulation

	ID_processore	Istante_di_inizio_I	Durata
Lotto52	Milano.Controllore.Fresa2	3:12:25.0000	5:35.0000
Lotto44	Milano.Controllore.Stozzatrice1	3:13:43.0000	3:35.0000
Lotto52	Milano.Controllore.Stozzatrice1	3:18:00.0000	3:35.0000
Lotto53	Milano.Controllore.Pettifica2	3:51:48.0000	8:30.0000
Lotto51	Milano.Controllore.Tornio1	4:44:44.0000	12:00.0000
Lotto42	Milano.Controllore.Tornio1	4:56:44.0000	12:00.0000
Lotto54	Milano.Controllore.Fresa1	3:22:10.0000	5:35.0000
Lotto44	Milano.Controllore.Tornio1	5:08:44.0000	10:00.0000
Lotto026	Milano.Controllore.Fresa1	3:30:44.0000	7:49.0000
Lotto55	Milano.Controllore.Pettifica2	4:00:18.0000	7:05.0000

Figure 7 – Output of Simple++ order table during the simulation

## 5 CONCLUSIONS AND FURTHER RESEARCHES

This paper proposes a web-based, HLA-compliant simulation framework for production plans remote control. Adoption of this framework lets the user to choose any COOTS simulators for system modelling, neither binding the simulation execution nor the modelling procedures to a specific kind of technology. An important aspect of this work is the generality of the described framework. Particularly, the use of XML as the standard for the information exchange allows scalability and full unbinding to the users in system modelling. Specially, we defined a standard that differentiate communication regarding persistent objects, like work centers, from non persistent objects, like job orders.

Tests demonstrated the possibility and the conceptual correctness of the architecture. Surely, a more intensive set of tests will be useful in order to verify framework robustness. Our tests were made in a local LAN, with a single production plan and without stochastic elements. Nor the internet velocity has been considered neither the CPU performances has been tested. Even if this standard can't be considered fully tested, it solves some critical issues in the distributed simulation area. Firstly, the architecture for information exchanging among federates solves the synchronization problem, also recurring in previous works [4]. Then, the combined use of HLA interaction and XML guarantees the right sequence in the information's arriving. Last but not least, the modularity approach and object oriented

programming of each element of the system permit the full separation of the information management.

As future developments, an in-depth study of the architecture with different types of LAN would be very interesting, in order to evaluate the effect of delay in information delivering. It would also be hoped the introduction of stochasticity in the physical model, in order to evaluate it, and finally a full integration in the *Remote Factory* project idea would be the natural continuation of this work.

## 6 REFERENCES

- [1] Zeigler, Ball, Cho, Lee, Sarjoughian, 1999, "Implementation of the DEVS Formalism over the HLA/RTI: Problem and Solution.
- [2] Huang Xueqin, Miller John A., 22-26 Aprile 2001, "Building a Web-Based Federated Simulation System With Jini and XML", Simulation Symposium, 2001. Proceedings, pages 143-150.
- [3] Zeigler, Kim, Buckley, 1999, "Distributed Supply Chain Simulation in a DEVS/CORBA execution environment", Proceeding of the 1999 Winter Simulation Conference.
- [4] Carofiglio Andrea, Di Benedetto Paolo, 2001, "Il Progetto REMOTE FACTORY: Utilizzo della web based simulation per il benchmarking di sistemi di schedulazione e controllo", 2001
- [5] Tecnomatix Technologies  
Homepage URL <http://www.tecnomatix.com>
- [6] Bettini G., Rapaccini M., Tucci M.,- Automatic Modelling Of Manufacturing Systems With Conventional Stochastic Discrete Events Simulation Languages, Proceedings of 9th European Simulation Symposium, ESS97, Passau (D), 19-22th October 1997, pp. 411-415.
- [7] Page, Moose, Griffin, 1997, "Web based simulation in SimJava using Remote Method Invocation", Proceeding of the 1997 Winter Simulation Conference.
- [8] IMS-Network of Excellence (IMS-NoE, 2003), [www.ims-noe.org](http://www.ims-noe.org)
- [9] Solberg J.J., Lin G.Y.J. (1992). Integrated shop floor control using autonomous agents. IIE Transactions, Vol. 24, No. 3, pag. 57-71
- [10] Wild Web Integrated Logistics Designer, 1999-2000, Research Project funded by M.U.R.S.T.
- [11] M. Tucci, R. Revetria, Different Approaches in Making Simulation Languages Compliant with HLA Specification, Proceedings of SCSC 2001, pp. 622-628, Orlando (FL), July 15-19, 2001 (ISBN 1-56555-241-5)
- [12] Defence Modeling and Simulation Office (DMSO), 2001, DMSO High Level Architecture Homepage URL <http://hla.dmsomil/>
- [13] Cavalieri S. M. Macchi, S. Terzi, (2002), Benchmarking Manufacturing Control Systems: Development issues for the performance measurement system. In: Proceeding at IFIP Performance Measurement Workshop, Hanover, Germany

- [14] XML, (2003), [www.w3.org](http://www.w3.org)

## 7 AUTHORS BIOGRAPHY

**Romeo Bandinelli** took his Laurea Degree in Mechanical Engineering at Florence University in April 2002 discussing the thesis "Remote Factory Control with Distributed Simulation". Actually, he is a PhD student of University of Florence, Department of Energetic, Plants and Industrial Technologies Section. His current research interests are Parallel and Distributed Simulation applied to industry and supply chain context, ICT, business process re-engineering .

**Sergio Terzi** is a PhD student of Politecnico di Milano, Department of Economics, Industrial and Management Engineering, Laboratory of Production Systems Design and Management. He is also taking his PhD in conjunction with CRAN laboratories, University of Nancy I, France. He received his B.S. in Management Engineering degrees from the University of Castellanza in 1999 and from the same university he received his M.Sc. degrees in Economics in 2002. His current research interests are Distributed Simulation applied to industry and supply chain context, Technologies enabling Product Lifecycle Management within SME and Modelling of Production Systems.

**Mario Rapaccini** took his Laurea Degree with honors in Mechanical Engineering at Florence University in April 1996. He is a professional engineer since 1996. In May 2000 he achieved Ph.D. discussing the thesis Advanced tool for configuration and impact assessment of Integrated Municipal Solid Wastes Management Systems. Currently, he's assistant professor in SSD ING-IND/35. Research topics covered are: managerial economics and business organisation, ICT, simulation modelling and analysis (SM&A), business process re-engineering (BPR). He's fellow of AiIG, ANIMP, AIRO and ANIPLA.

**Marco Macchi** graduated in October 1997 in Management and Production Engineering at Politecnico di Milano. He is currently researcher at the Department of Economics, Industrial Management Engineering at Politecnico di Milano. His current fields of interest are Design and Automation of Manufacturing Systems, Modelling and Simulation of Manufacturing Systems, Application of Multi-Agent Systems, Computer Integration in Manufacturing Systems Engineering, Maintenance Management. He has published more than 20 papers on international journals and national and international conference proceedings. He is member of Special Interest Group on Advanced Techniques in Production Planning and Control of WG 5.7 of IFIP.

---

The paper is the result of a joint work conducted by the authors; Romeo Bandinelli wrote par. 3.2 and 4, Mario Rapaccini par. 1, Sergio Terzi par. 2 and 3.1, Marco Macchi contributed to par. 5.

# A SIMULATION-BASED ANALYSIS OF THE CYCLE TIME OF CLUSTER TOOLS IN SEMICONDUCTOR MANUFACTURING

Heiko Niedermayer  
Institute of Computer Science  
University of Tübingen  
D-72076 Tübingen, Germany  
E-mail: niederma@informatik.uni-tuebingen.de

Oliver Rose  
Institute of Computer Science  
University of Würzburg  
D-97074 Würzburg, Germany  
E-mail: rose@informatik.uni-wuerzburg.de

## KEYWORDS

simulation, manufacturing, semiconductor, cluster tools

## ABSTRACT

Cluster tools are widely used in modern semiconductor manufacturing facilities. In parallel mode they offer high throughput at the cost of a complex behaviour with regard to lot cycle times. The reason is that cluster tools are behave like small factories themselves. We analyze the slow-down of the processing of a lot that is caused by other lots in the tool and examine how the slow-down factor can be used for scheduling and for predicting lot cycle times. This cycle time analysis is mandatory for production planning and can only be done by simulation so far.

## INTRODUCTION

Since the middle of the 1990s cluster tools are becoming more and more important in semiconductor manufacturing. The most recent manufacturing facilities consist almost exclusively of cluster tools. Cluster tools are machines that combine several processing steps in one machine. They can be regarded as small factories inside a factory. They consist of loadlocks, processing chambers, and handlers. Figure 1 shows the structure of a simple cluster tool with 2 loadlocks, 5 chambers, and 1 handler.

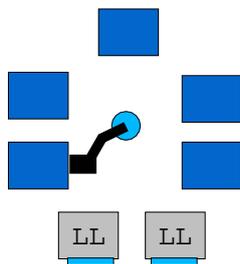


Figure 1: A Simple Cluster Tool Model

Each loadlock can be loaded with one lot. A lot is a box with wafers, e.g., 25 wafers. Then, the tool processes the lot. Most modern cluster tools have 2 loadlocks. Each wafer of the lot is scheduled inside the cluster tool by the scheduler of the tool. The handlers are used for

moving the wafers between the chambers and loadlocks. The chambers are machines to process wafers.

One advantage of clustering processing steps is that the processing of the wafers is pipelined. This reduces the cycle time of the lot as only the processing time at the bottleneck of the steps limits the cycle time and not the sum of the raw processing times of all steps.

An additional advantage is that cluster tools save clean-room space. Inside the cluster tool there is vacuum, hence, a low number of particles. As a consequence, the clean-room quality outside the tool can be lower than in traditional fabs.

A disadvantage of cluster tools is that their behavior is more complex than the behavior of simpler machines. The cycle time of a lot is not constant but depends on the situation inside the cluster tool during the processing of the lot. This is due to the fact that in parallel mode cluster tools are able to process lots in parallel that share the same resources.

When a machine processes only one lot the cycle time is simply determined by a constant or by a single random variable. When a cluster tool processes lots in parallel this is more complex. Each lot overlaps with other lots. During the overlaps the lots share the same resources and the lot cycle time depends considerably on the lot combinations inside the cluster tool. The shared use of resources slows down the processing of the lots. As Figure 2 illustrates comparing case A (single mode) and case B (overlap) the gain in makespan is less than the change in start time:  $\Delta c \leq \Delta d$ .

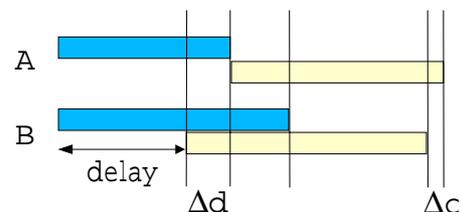


Figure 2: Overlap Scenario

Despite of the fact that the processing of each lot is slowed down the overall throughput and the utilization of the expensive machines inside the cluster tool are

higher than in single mode where only one lot is processed at a time.

Considering the different overlaps and the different lot types, the number of possible situations is huge. The overlap size can range from almost 0 seconds to the complete cycle time of a lot, say, 4000 seconds. Figure 3 illustrates that the cycle time of a particular lot depends considerably on its amount of overlap with the lot that is produced in parallel on a cluster tool.

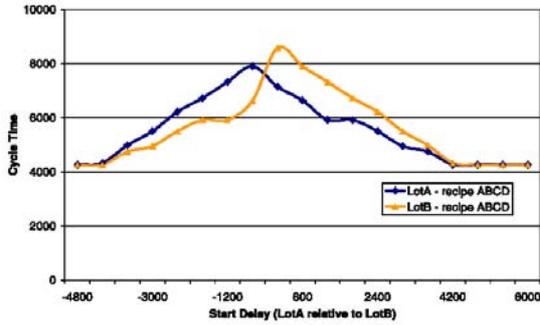


Figure 3: Cycle Time and Overlap

Additional factors are the number of recipe combinations and the different lot sizes. Thus, the cycle time cannot be computed in advance. Instead the complete scenario has to be simulated to determine the lot cycle times. At the moment simulation is the only approach to determine the performance of cluster tools in a detailed manner.

If we need to evaluate a schedule for a cluster tool we have to simulate the schedule to determine the lot cycle times, lot completion times, and the makespan.

For simulation, we use the cluster tool simulator CluSim that was developed at the Department of Computer Science at the University of Würzburg by Mathias Dümmler and a number of students (Dümmler 1999; Bohr 1999; Schmid 1999). Dümmler proposed a genetic algorithm for optimization computing the fitness for each schedule with the simulator. CluSim is also used at Infineon Technologies for cluster tool optimization.

When we use simulation for computing the lot cycle time this takes much more CPU time than simply taking a fixed cycle time from a data set (maybe with a setup taken from a setup matrix). Therefore, search approaches that test various schedules are more expensive for cluster tools than for other machines.

In this paper, we analyze how lots that are processed in parallel slow down each other. In the next Section we introduce the slow-down factor and some of its properties. Then we show that to a large degree the slow-down factor can be explained by a change of

bottlenecks. Finally, we use the slow-down factor for approximation and scheduling.

## RELATED WORK

A lot of cluster tool research focused on the scheduling inside the cluster tool and on simulation. Analytic performance analysis was done by (Perkinson et al. 1994). They analyzed cluster tools with one loadlock and no parallel chambers, identical deterministic transport and process times. Developers of simulation software still use these Perkinson models for evaluating the correctness of their simulator. Later Perkinson et al. extended their model allowing for, e.g., redundant chambers (Perkinson et al. 1996). There are also approaches using petri nets, for instance, for single mode cluster tools (Srinivasa 1998).

Simulation for analyzing cluster tool performance was used in (Atherton et al. 1990) and (Koehler et al 1999). Both papers show that simulation is mandatory for accurate prediction of performance estimates like cycle times or chamber utilizations.

A detailed introduction to cluster tools can be found in (Atherton and Atherton 1995).

Considering large fab scheduling problems efficient methods are needed to schedule facilities with cluster tools. Our approach of simulating or measuring slow-down factors and use them for scheduling or for cycle time prediction can save a lot of time during optimization.

## SLOW-DOWN FACTORS FOR LOTS

To study the effects of overlaps we introduce the slow-down factor. Lot B has an influence on lot A and usually the processing of lot A will take more time than without lot B (Figure 4).



Figure 4: Lot A Is Slowed Down By Lot B

*Definition 1:* Slow-down Factor

The slow-down factor of lot A while processed in parallel with lot B is defined as

$$SDF(A, A+B) = \frac{Cycletime(A, A+B)}{Cycletime(A)} \quad (1)$$

where  $Cycletime(A, A+B)$  is the cycle time of lot A when it is processed together with lot B and  $Cycletime(A)$  is the cycle time of lot A when it is processed alone (single mode).

The slow-down factor is a measure for how much lot B disturbs lot A. We can use this information both for scheduling and for approximating the lot cycle times. For the rest of the paper, we only consider cluster tools with 2 loadlocks because of their importance in manufacturing. We created 4 test sets with 12 recipes each. Most recipes of the sets “Dresden” and “Dresden fast handler” were inspired by descriptions of etch centers at Infineon Technologies’ Dresden factory. The set “Villach” is based on descriptions of Endura cluster tools at Infineon Technologies’ Villach factory (Seidel 2001). The last set “Simple” contains recipes where a single chamber is the bottleneck.

### Slow-down factors and start delays

Considering the overlap of two lots we have to deal with the influence of different delays between the start times of the two lots. Processing the wafers is a cyclic process. The initial delay between the first and the second lot determines when the wafers of the second lot start to disturb the wafers of the first lot in this cyclic process. This may lead to different solutions for the internal schedule and therefore to different slow-down factors.

The figures show that depending on the lot combination the slow-down factor can be almost independent of the delay as well as highly variable for different delays. In Figure 5 the peaks are roughly 20 % higher than the average slow-down factor for this combination.

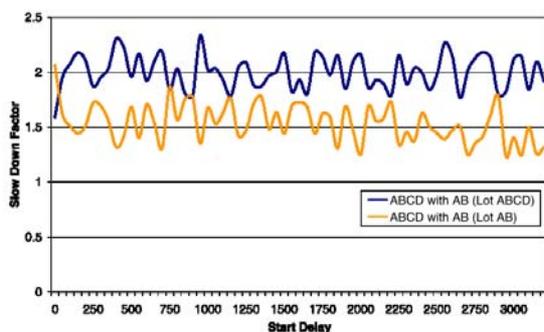


Figure 5: Slow-down Factor and Start Delay

Figure 6 shows that the first lot may be preferred by the cluster tool. This depends on the internal scheduler of the tool. Our simulator definitely tends to prefer the first lot. When the lot is the first (delay = 0) then the figure shows a slow-down factor of 1.5. When it is the second lot in the tool then the slow-down factor is almost 3. The slow-down factor decreases for higher delays, as the lot becomes the first lot when the other lot is replaced with the next lot.

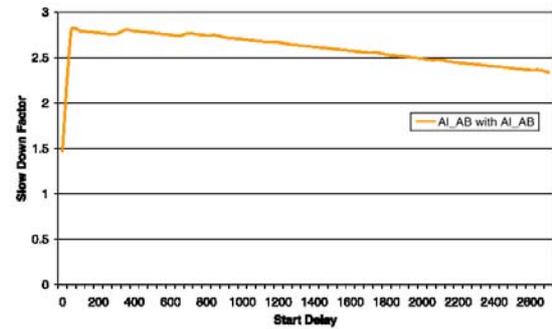


Figure 6: First Lot Preferred

### The value range of slow-down factors

The optimal slow-down factor is 1. This means that the lot was not slowed down and it was as fast as in single mode. One might expect that 2 is a maximum and is reached as slow-down factor for identical lots, but this is not true. The scheduler may prefer some lots and penalize others, as fairness and throughput may be conflicting targets in some occasions. Precedence constraints can cause a wafer to block wafers of the other lot that could be processed with a higher throughput rate.

It is obvious that on average the slow-down factor should be less than 2. Otherwise, parallel-mode processing would be worse than single-mode processing and the lots should be processed one after the other.

### Slow-down factors and pump and vent times

When a lot enters the cluster tool its loadlock needs to be pumped to the vacuum level of the cluster tool. The time for this operation is called pump time. During this time this lot does not influence other lots. So, the slow-down factor during this interval is 1 for both lots.

When all wafers of a lot are completed then normal air pressure has to be restored in its loadlock. The time for this operation is called vent time and during this time the slow-down factor is 1. However, pump and vent times are small compared to the overall cycle time of lots with standard lot sizes of, say, 25 wafers. So, we only consider the effects of pump and vent times when we have to deal with small lot sizes like in the next section.

### Slow-down factors and lot size

As Figure 7 shows the variation of the slow-down factor and therefore of the cycle time is larger for small lot sizes and decreases when the lot size is increased.

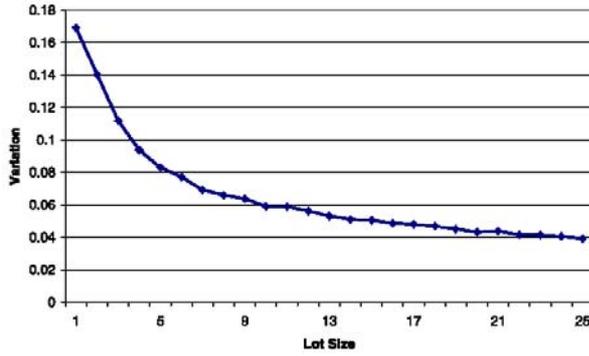


Figure 7: Variation Due to Lot Size

The slow-down factor itself did not significantly change for different lot sizes in general. However, when there is variation the variation tends to increase the slow-down factor. Thus, averages over all possible combinations will result in larger slow-down factors for small lot sizes.

Let us consider a lot with just one wafer. The time the wafer has to wait for a critical chamber to become available strongly effects the lot cycle time. Hence, obviously the transient is more important for lots with small lot sizes. Most wafers of lots with large lot sizes are processed during the steady state.

### Slow-down factor variation

Finally, we examine the variation of the slow-down factors within the same recipe combination. Table 1 lists the average slow-down factors for all combinations, their average variation within each lot combination and the average minimum and maximum slow-down factor for each lot combination.

Table 1: Average Slow-down Factors (SDF)

Test set	SD F	Var.	Min.	Max
Dresden fast handler	2.0	0.05	1.7	2.1
Dresden	2.2	0.07	1.8	2.3
Villach	2.0	0.08	1.6	2.1
Simple	1.9	0.05	1.7	2.1

Table 1 also shows that not all recipe combinations make sense, since in average the slow-down factor is roughly 2. A slow-down factor above 2 is not better than processing the lots one after another in single mode.

## SIMULATION AND APPROXIMATION OF SLOW-DOWN FACTORS

### Simulation

For the computation of simulated slow-down factors we created specific simulation studies. For any lot combination we simulated one lot of type A (lot A) being processed parallel to lots of type B. To ensure that lot A is always parallel to a lot of type B we used more than one of these lots (lot B, lot C). As illustrated in Figure 8, for each lot combination we also simulated different delays between the start of the first lot of type B and lot A.

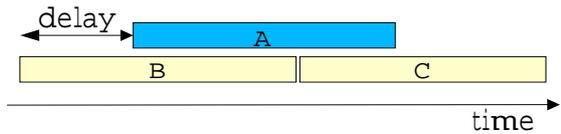


Figure 8: Simulating Slow-down Factors

### Approximation

The slow-down factor indicates how recipes disturb each other. For different combinations there is usually a change in bottlenecks and we assumed that this change determines the slow-down factor to a considerable extend.

#### Definition 2: MBRPT

MBRPT (maximum flow bottleneck raw processing time) is our approximation approach for the slow-down factor. The approximation is computed as follows:

$$SDF(A, A+B) \approx \frac{MRPT(Bottleneck, A, A+B)}{RPT(Bottleneck, A)} \quad (2)$$

where  $RPT(Bottleneck, A)$  is the bottleneck work load of a work load distribution for lot A alone and  $MRPT(Bottleneck, A, A+B)$  is the bottleneck work load of a work load distribution for lot A and lot B where the work load of all chambers that are not used by lot A is set to 0.

To compute a workload distribution we ignore the precedence constraints of the recipes. We recommend a heuristic algorithm using the LFJ rule (least flexible job first). The general problem of optimally distributing work is NP-hard and very similar to scheduling parallel machines with machine dedication. The LFJ rule is optimal for this problem when the sets of the machine dedication are nested and the processing times for parallel machines are equal (Pinedo 2001).

MBRPT approximates the simulated slow-down factors with an average error of 25 to 35 % depending on the scenario. This error is rather high, but this is no surprise as MBRPT assumes completely fair scheduling while the scheduler of the cluster tool may prefer lots. The bias of MBRPT varies from - 2 % to 2 %, i.e., MBRPT is practically unbiased.

## SCHEDULING APPROACH BASED ON SIMULATION RESULTS

### Approximating lot cycle times

Given the cycle time of the lots in single mode (processed alone in the cluster tool) and given a simulated slow-down factor for each lot combination we can use these values to compute approximate lot cycle times. For each overlap we determine the length of the overlap and how much of the work of each lot has been completed during this overlap. With this idea we can predict the lot cycle times for all lots in a scenario with only few floating-point operations.

We simulated 20 scenarios with 20 lots each and then compared the predictions based on the simulated slow-down factors with the simulation results. Table 2 shows the average prediction error for the end time and cycle time of a lot. As the slow-down factor varies up to 20 % for different delays between the overlapping lots the prediction quality is limited by this variation. Additionally, prediction errors add errors to the predictions for the next lots. Analysis showed that the error is high for lot combinations with poor performance and when a lot is treated unfair by the cluster tool simulator (slow-down factor > 3). The error is smaller for lot combinations with high throughput.

Table 2: Prediction Errors

Test set	Avg. Error [end time]	Avg. Error [cycle time]
Dresden fast handler	14.3 %	26.3 %
Dresden	11.3 %	26.4 %
Villach	14.8 %	19.7 %
Simple	6.5 %	16.3 %

### Scheduling with slow-down factors

The simulated slow-down factors can help to decide whether a lot combination is good or whether its performance is poor. When using a search algorithm to explore the search space of all possible schedules, the slow-down factor may be a good heuristic which paths to examine and which paths to ignore. As the simulated slow-down factor is an average taken from experiments with different delays it is a good measure for general lot compatibility while it may not be suitable for highly accurate predictions of particular lot cycle times for a scenario as in the last section. Pilot studies using dispatching rules on the basis of slow-down factors show promising results.

## CONCLUSIONS

Among other advantages parallel mode cluster tools offer high throughput and high utilization. We demonstrated that the lot cycle times for cluster tools cannot be determined without simulating or predicting

the complete scenario. This is caused by lots overlapping and sharing resources, hence, slowing down each other. Slow-down factors help to understand how lots disturb each other and can be used for a fast approximating of lot cycle times and as heuristic for scheduling.

Further studies have to be made for more representative results on the properties of slow-down factors and on the quality of the approximation approach. The predictions on the basis of slow-down factors have to be improved as they are promising for scheduling heuristics and can provide lot cycle time predictions with only few floating-point operations instead of long simulation runs.

Standard scheduling approaches do not take into account that the lot cycle times of parallel lots are correlated. This will be a field of further research in cluster tool optimization.

## REFERENCES

- Atherton, L.F. and R.W. Atherton. 1995. *Wafer Fabrication: Factory Performance and Analysis*. Kluwer.
- Atherton, R.W.; F.T. Turner; L.F. Atherton; and M.A. Pool. 1990. "Performance Analysis of Multi-Process Semiconductor Manufacturing Equipment." In *Proceedings of the IEEE/SEMI Advanced Semiconductor Conference 1990*.
- Bohr, M. 1999. "Schedulingverfahren für Cluster Tools in der Halbleiterfertigung." Master thesis. Department of Computer Science. University of Würzburg, Germany.
- Dümmler, M. 1999. "Using simulation and genetic algorithms to improve cluster tool performance." In *Proceedings of the 1999 Winter Simulation Conference*. 875-879.
- Koehler, E.J.; T.M. Wulf; and A.C. Bruska. "Evaluation of Cluster Tool Throughput For Thin Film Head Productions." In *Proceedings of the 1999 Winter Simulation Conference*. 714-719.
- Perkinson, T.L.; P.K. McLarty; R.S. Gyurcsik; and R.K. Cavin III. 1994. "Single-Wafer Cluster Tool Performance: An Analysis of Throughput." *IEEE Transactions on Semiconductor Manufacturing*, 7 (3), 369-373.
- Perkinson, T.L.; P.K. McLarty; R.S. Gyurcsik; and R.K. Cavin III. 1996. "Single-Wafer Cluster Tool Performance: An Analysis of the Effects of Redundant Chambers and Revisitation Sequences on Throughput." *IEEE Transactions on Semiconductor Manufacturing*, 9 (3), 384-400.
- Pinedo, M. 2001. *Scheduling. Theory, Algorithms, and Systems*. 2<sup>nd</sup> edition. Prentice-Hall.
- Schmid, M. "Modellierung und Simulation von Cluster Tools in der Halbleiterfertigung." Master thesis. Department of Computer Science. University of Würzburg, Germany.
- Seidel, G. 2001. "Simulation und Optimierung von Cluster Tools in der Halbleiterfertigung". Master thesis, Institute of Mathematics. Technical University of Graz, Austria.
- Srinivasan, R.S. 1998. "Modelling and Performance Analysis of Cluster Tools Using Petri Nets." In *IEEE Transactions on Semiconductor Manufacturing* Vol. 11, 1998.

## **AUTHOR BIOGRAPHIES**

**HEIKO NIEDERMAYER** is Ph.D. candidate at the Institute of Computer Science (Chair of Computer Networking and Internet) at the University of Tübingen. He received an M.S. degree in Computer Science from the University of Würzburg. His e-mail address is:  
`niederma@informatik.uni-tuebingen.de`.

**OLIVER ROSE** is assistant professor in the Department of Computer Science at the University of Würzburg, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from the same university. He has a strong background in the modeling and performance evaluation of high-speed communication networks. Currently, his research focuses on the analysis of semiconductor and car manufacturing facilities. He is a member of IEEE, ASIM, INFORMS, and SCS. His web address is:  
`www3.informatik.uni-wuerzburg.de/~rose`.

# INTEGRATION OF PROCESS AND CONTROL SIMULATION INTO THE ENGINEERING PROCESS

M. Hoyer, C. S. Horn, R. Schumann  
Forschungsschwerpunkt AUBIOS  
University of Applied Sciences and Arts Hannover  
Ricklinger Stadtweg 120, D-30459 Hannover, Germany  
E-mail: markus.hoyer@mbau.fh-hannover.de

G. C. Premier  
School of Technology  
University of Glamorgan  
Llantwit Road, Pontypridd, CF37 1DL, Wales, UK

## KEYWORDS

Chemical plant, process engineering, control engineering, model library

## ABSTRACT

Testing of a chemical plant is done mainly during its start-up and commissioning phase and in general requires a considerable amount of time and money to correct hardware and software problems. Using model based plant simulation directly after completion of detailed plant engineering, the main testing and debugging could be done by simulated virtual plant thus reducing the time and cost of the start-up phase. This paper describes an approach to generate the required plant models automatically from a model catalogue in parallel to the engineering process.

## INTRODUCTION

Simulation technology has become a widely used technique in all phases of the chemical engineering cycle, from process synthesis and conceptual design, through basic and detail engineering. It is also used in process control, monitoring and operator training (Schuler, 1995). Growing economical and ecological constraints require further tightening of the engineering cycle and thus demand the application of simulation technology during all design/development phases. The development of the necessary process models, however, requires highly sophisticated expert knowledge and is in general time-intensive. Over the last decade, this has led to an increasing interest among research groups to develop methodologies for model generation based on computer-aided systems.

Advanced computer-aided modelling environments share at least some of the following characteristics:

- basic modelling objects, e.g. phenomena-based
- model representation with high abstraction to facilitate model re-use
- comprehensive data model to ease model maintenance
- implementation of work flows to facilitate reproducible modelling and automation of modelling tasks

(Stephanopoulos et al. 1990) and (Bieszczad 2000) focus on the development of a phenomena-based modelling language (MODEL.LA). Furthermore, the modeller is assisted in specifying the modelling problem and thus providing a basic work flow.

(Jensen and Gani 1999) describe a process-modelling tool (ModDev) which is composed of a knowledge-based system and a generic modelling language. Model abstraction is achieved by uniformly distributed regions (shells) and connections between those regions. Models for unit operations are derived by aggregation from those fundamental building blocks.

(Linninger et al. 2000) present an approach for computer-aided model generation and an associated environment (TechTool). This is based on a generic object-oriented (object inheritance framework) and phenomena-based mathematical language. Meta-modelling is employed to facilitate model re-use and adaptability of the framework and to achieve a purely declarative formulation of modelling problems.

(Tränkle et al. 2000) describe a process modelling tool (ProMot) that supports the modeller through an object-oriented modelling language and a graphical user interface. The modeller can build process models from basic structural or behavioral modelling entities through aggregation and/or inheritance by either means. Knowledge representation is provided by a frame definition language.

(Bogusch et al. 2001) describe a comprehensive framework (ModKit) aimed at supporting the entire model development process. A knowledge-based approach has been adopted for the model representation including phenomena-based objects. A data model aimed at chemical engineering data (VEDA team 1999) has been developed. Work flows have been implemented which allow for partial automation of modelling tasks.

(Fritz and Engell 1997) describe an architecture for the simulation of batch processes (BaSiS). The system is characterized by object-oriented components (model builder, simulator and output client), which provide the framework for the implementation of specific interfaces, e.g. for a specific simulator. Thus a substantial degree of flexibility with respect to the simulation task is achieved.

However, all these approaches are only suitable if CAE and simulation experts are available throughout the

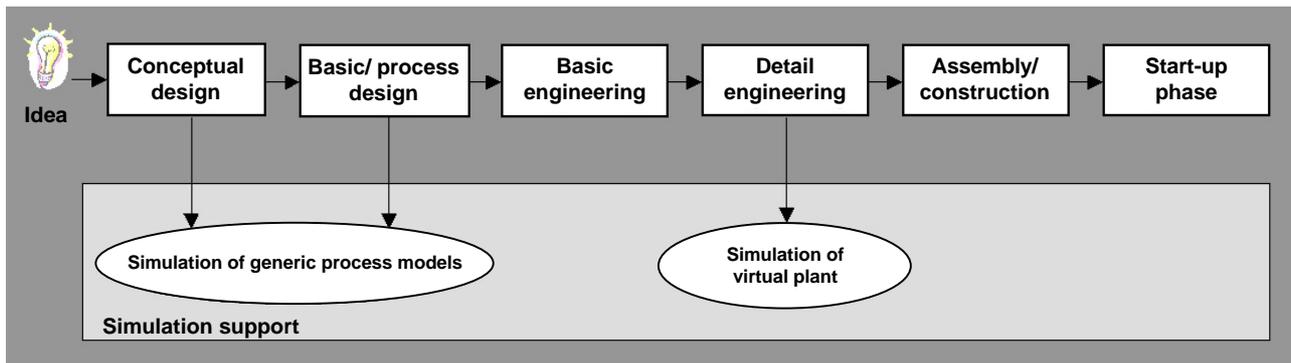


Figure 1: Simulation Support During the Engineering Process of a Chemical Plant

whole design cycle. Therefore, for all phases of process engineering, where such experts are not available (which is frequently the case in small or medium sized industrial companies), new ways have to be found and new tools developed to support the planning process by simulation. Also, all but one of the above approaches concentrate on the early phases of the engineering process and tend to support simulation for conceptual design, Fig. 1. The approach proposed in this paper is aimed at the later phases of the engineering process and particularly at the detail engineering phase in which the final specification of parts and components is done by project engineers without specific modelling and simulation expertise. After completion of the detail engineering, all information is prepared for the construction and assembly of the plant. The functionality of the plant, however, is not usually tested until the start-up phase, during which time, all hardware and software problems become obvious and must be resolved in a time and money intensive debugging procedure. By simulation of the plant, the test and debugging phase could be shifted back to the end of the detail engineering phase where in principal all required technical information for the building of the plant has been compiled. Currently, however, this information cannot yet be transformed automatically into a simulation model of the plant which could be used by the planning engineer for test purposes. This paper outlines a general approach for the automatic generation of plant simulation models in parallel to the engineering process based on a component model catalogue, not requiring specific modelling expertise of the planning engineer to run a simulation. Based on such models, testing and debugging could become available to the planning engineer, on the simulated virtual plant before the real plant is built, thus reducing time and cost during the start-up phase.

## GENERAL CONCEPT

After completion of detail engineering all components of the plant are completely specified: So e.g. the planning engineer has chosen a pump with its typical

characteristics from the catalogue of a specific manufacturer. In order to simulate the functionality of the pump for the process planning engineering or the control planning engineering disciplines, specific simulation models must be made available. These should allow the simulation of the flow through the pump depending upon the fluids and pressures, for the process engineer. Likewise, the dynamic response of the flow to a change of the driving input, would often be required by the control engineer etc.. In electrical engineering, especially for printed circuit board (PCB) design, simulation models of electrical components are provided by the suppliers, often before the silicon itself becomes available and can be used to simulate the function of the electrical circuit during the board design. Such a systematic approach is still missing in chemical and mechanical engineering. The intriguing idea to get the simulation models from the component suppliers in order to distribute the effort for the creation of the simulation model is one basic principle for the automatic model generation concept proposed here, see Fig. 2.

Let us assume for a moment that such a systematic model collection methodology exists and that all required component models are stored in the CAE system for process engineering in a simulation model catalogue. In parallel to the selection and specification of components for the plant during detail engineering, a plant simulation model could now be automatically aggregated from the component simulation models – and this idea forms the other basic principle for the automatic model generation concept proposed in this paper, see Fig. 2.

Although the general idea is simple, the realisation of such a modelling concept requires the solution of a number of complicated tasks. In order to restrict the complexity the following description concentrates on models for the process and control engineer which allow them to test the respective functionality of the plant that is of interest to them. Other modelling aspects are excluded for the moment but may be added later.

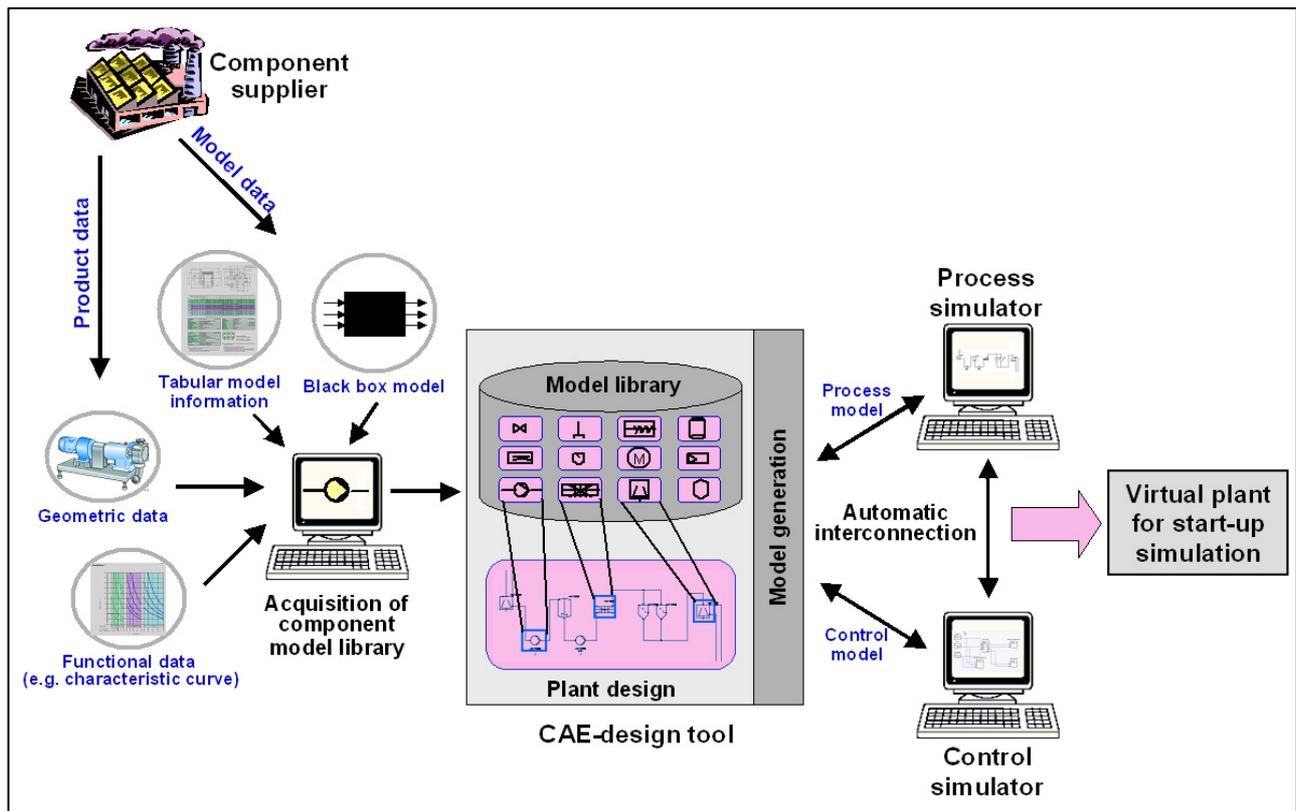


Figure 2: General Concept for Model Catalogue Based Simulation

## TASKS AND REQUIREMENTS

For the proposed approach, the following tasks and requirements will be considered:

- structure of the component models
- quality of the component models
- organisation of the model catalogue according to model aspects such as model type and storage format
- automatic model aggregation within a process engineering CAE tool

### Component Model Structure

Component models should be provided by the component suppliers. In order to make this possible two principal approaches can be taken:

- (1) Table based models. A generic model is defined for each component class with a prescribed structure and parameterisation. In order to acquire the component model information, the supplier must only provide the parameterisation for his specific component which could be collected using templates. This would simplify the modelling task for the supplier but would require the generation of generic models for all necessary component classes – a considerable work load.

- (2) Black box models. An input/output structure is defined for each component class with prescribed signal types. The suppliers provide a black box model for the defined input/output model structure – this would require an explicit model designed by the supplier, but would require a relatively small effort for the definition of the input/output structure, for the required component classes. The problem here is to define a compatible exchange format for such black box models (DLLs, models for specific simulators etc.).

In a prototypical implementation of the approach presented in this paper, both model types have been taken into account. No matter which approach is used, the principal question arising is how models for significantly different simulation aspects – be they for process simulation or control simulation – can be combined efficiently in the model catalogue to support all required simulation tasks.

### Component Model Quality

The quality of component models is inextricably linked to the process model performance and is determined by the validation of the process models for the envisaged simulation task. The validity of the models can be characterized by the experimental or theoretical conditions under which these models were derived. These conditions and model characteristics form an

integral part of the model description and have to be supplied by the model developer. The range of validity of a model can be expressed in the form of declarations and in terms of valid model parameters and input signals, for example. Declarations regarding valid process conditions have to be brought to the attention of the process engineer during component specification, e.g. requiring confirmation. Invalid model parameters can be rejected during model specification. Thus only valid parameters can be specified. During a simulation run invalid input signals can be handled with alerts and automatic interruption of the simulation. These procedures provide different degrees of control with respect to the quality of simulation models.

### Organisation of the Model Catalogue

All kinds of process models and control models have to be collected in the model catalogue. According to their nature, these models reflect different aspects of the process – the process models concentrate on flows, temperatures or compositions of flows whereas the control models represent the dynamic interaction of all kinds of signals for control. For the storage of such simulation models the following alternatives can be chosen:

- (1) General model description language. By choosing a description language like XML the models could be formulated and stored independently from any specific simulator format. For the simulation, however, such models have to be converted to the format of the simulator used.
- (2) Simulator specific model format. Having specified standard simulation tools, the models could be stored in the specific formats of the simulators. Thus, the models can be directly used for simulation, though only for the specified simulator. For other simulators the models must be converted.

For the prototypical realisation, the second alternative was chosen to avoid the need for the definition of a general model format and a conversion utility.

### Process Engineering CAE Tool and Automatic Model Aggregation

During basic and detail engineering of a chemical plant in general, an object tree is generated in the process engineering CAE system, reflecting among other information, the connections of the planned components and their parameterisation. The stored engineering information must be augmented by a reference to all required component simulation models, including their connections to other component models (I/O references) and parameterisations. The model I/O references reflect not only the connections within one simulation world (process or control) but also the interaction of process and control simulation worlds: A pump may be directly driven by a control input which is generated in a control

scheme and the reaction of the flow to a change of the control input is generated by the process simulation. The changing flow may be measured by a flow sensor and transferred to the control scheme thus closing the control loop via the process model. This leads to an automatic integration of process and control simulation. The process simulator serves as process model for the control simulator whereas the control simulator provides the necessary control actions for the process simulation.

### IMPLEMENTATION CONCEPT

An object oriented CAE tool for process planning - Comos PT - serves as the basis for the integration of process and control simulation into the engineering process. The simulation of process models is carried out with the process simulator gPROMS, the block oriented simulator Matlab/Simulink serves as simulation tool for control models, see Fig. 3. Within Comos PT component model libraries are stored as a basis for (chemical) process simulation (with gPROMS) and control simulation (with Simulink). During the planning process within Comos PT component model objects are arranged and specified for process and control system equipment, containing all relevant model parameters and connection information. To allow the aggregation of simulation models, references to the model library with process models for gPROMS and control models for Matlab/Simulink are added to the Comos PT model objects. Once the engineering process has been

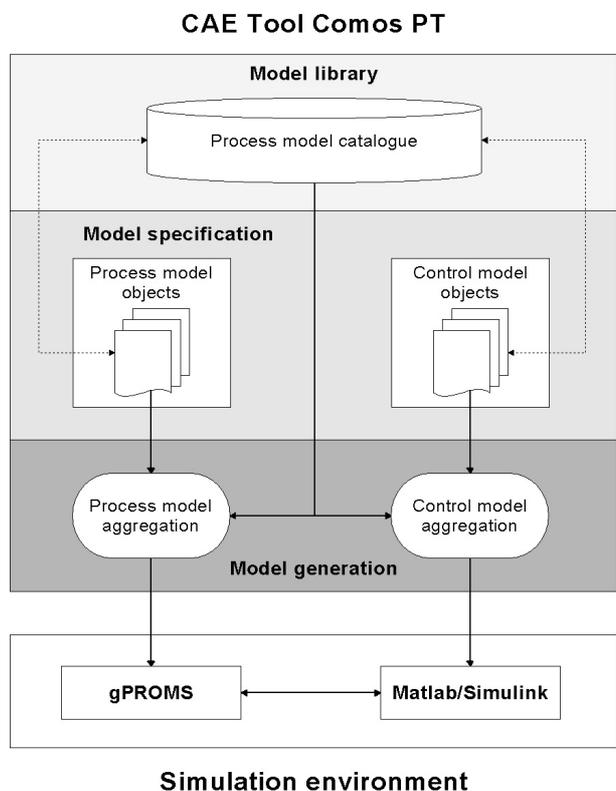


Figure 3: Generation of Specified Process Models from the Model Catalogue

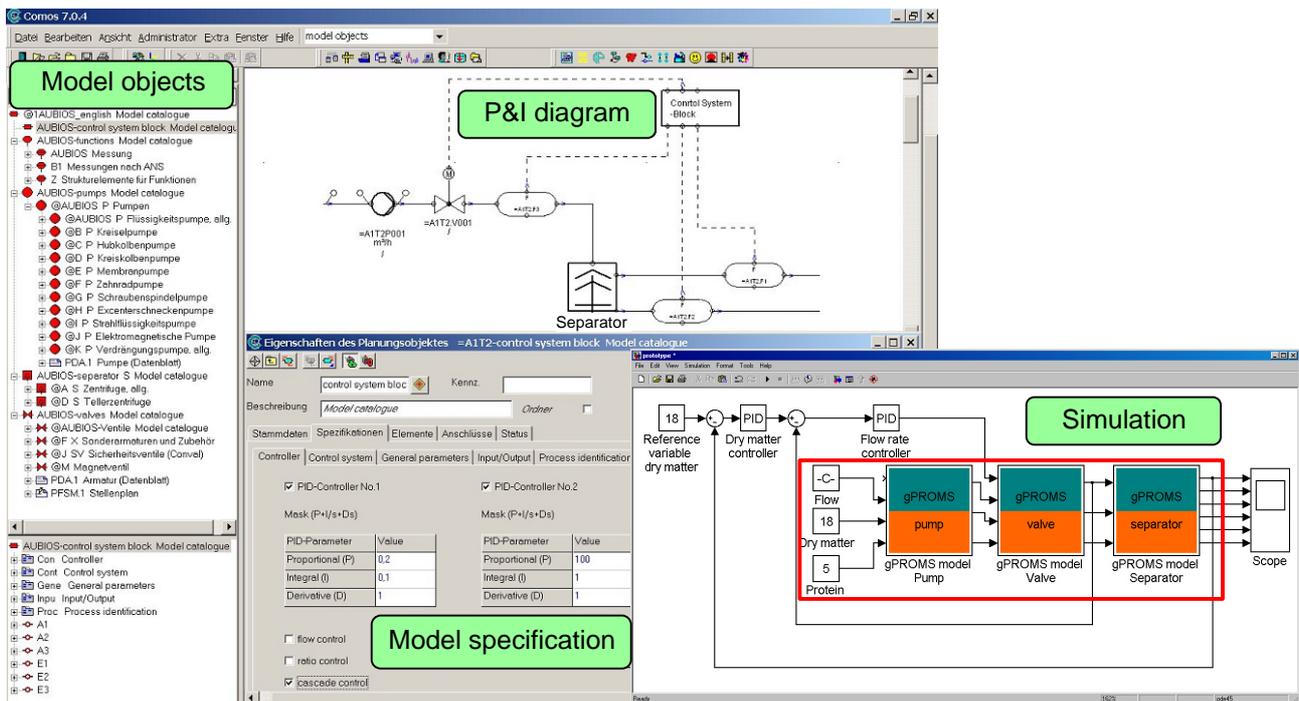


Figure 4: Application Example: Process Planning in Comos PT and Associated Simulink and gPROMS Simulation

completed the aggregation to the resulting plant simulation models and the parameterisation of the component simulation models can be done on the basis of the specified Comos PT model objects and their I/O connections according to Fig. 3.

## APPLICATION EXAMPLE

The proposed concept was tested for the separation step of a fresh cheese production process, in a prototypical implementation. In this step a disk stack centrifuge (separator) separates the coagulated milk into sour whey and fresh cheese. The fresh cheese production is a batch process and the separation is prone to disturbances that influence the separation efficiency, especially at the end of the cycle. In order to maintain constant product quality a control strategy for dry mass and protein content of the fresh cheese is required. For the separation components simulation models have been developed for the component model library in Comos PT as required for simulation with gPROMS and Matlab/Simulink, respectively. During the design of the separation process and its control scheme within Comos PT the model objects are arranged and specified in the P&I diagram and the corresponding object tree. Based on this information the aggregation of the simulation models for gPROMS and Matlab/Simulink is done, at present still manually, in the near future by an automatic model generator utility. Fig. 4 shows on the left, the Comos PT interface with the model objects, flow chart and model specifications and on the right, the aggregated simulation models in the Matlab/Simulink environment. The gPROMS process models are integrated as special blocks in the Matlab/Simulink environment.

Thus, the planned configuration of the separation process and its control system can be simulated by the planning engineer without specific modelling or simulation expertise.

## CONCLUSIONS

The integration of simulation into the engineering process for chemical plants allows, in general, the optimization and testing of the designed process at an early stage. The proposed catalogue based modelling approach aims at the simulation of the planned plant directly after completion of plant engineering, such that the plant's functions can be tested and debugged before it is built, yielding considerable time and money saving for the plant start-up phase. The required simulation models for the plant components should be collected directly from the component suppliers to distribute the effort for the component model generation. The plant simulation model is aggregated in parallel to the engineering process, making use of the information provided during the standard engineering process. Using this concept, plant simulation may become available in the future as standard test and debugging tool for the normal planning engineer without the need to become a modelling or simulation specialist.

Future work will include the development of an automatic model aggregation utility (model generator), with due regard to the generalization of the model generation systematic. The integration of more detailed simulation models (perhaps as complex as CFD models) will be considered. Industrial process realizations will be investigated, (e.g. by replacing the Matlab/Simulink simulation by the emulation of an industrial process control system), in order to create a simulation

environment as close to reality as possible for the planning engineer.

## REFERENCES

- Bieszczad, J. 2000. *A Framework for the Language and Logic of Computer-Aided Phenomena-Based Process Modeling*. PhD Thesis. Massachusetts Institute of Technology.
- Bogusch, R.; B. Lohmann and W. Marquardt. 2001. "Computer-Aided Process Modeling with ModKit". *Computers Chem. Eng.* 25, 963-995.
- Fritz, M. and S. Engell. 1997. "An Open Software Architecture for Batch Process Simulation". *Computers Chem. Eng.* 21, Suppl., 769-773.
- Jensen, A. K. and R. Gani. 1999. "A Computer Aided Modeling System". *Computers Chem. Eng.* 23, Suppl., 673-678.
- Linninger, A. A.; S. Chowdhry; V. Bahl; H. Krendl and H. Pinger. 2000. "A Systems Approach to Mathematical Modeling of Industrial Processes". *Computers Chem. Eng.* 24, 591-598.
- Schuler, W. 1995. *Prozesssimulation*. VCH, Weinheim.
- Stephanopoulos, G.; G. Henning, and H. Leone. 1990. "Model.La. A Language for Process Engineering. Part I and Part II". *Computers Chem. Eng.* 14, 813-869.
- Tränkle, F.; M. Zeitz; M. Ginkel and E. D. Gilles. 2000. "PROMOT: A Modeling Tool for Chemical Processes". *Mathematical and Computer Modelling of Dynamical Systems* 6, No. 3, 283-307
- VEDA team. 1999. "The Chemical Engineering Data Model VEDA". Part 1 – Part 6. Technical Report, Lehrstuhl für Prozeßtechnik, RWTH Aachen.

## AUTHOR BIOGRAPHIES

**MARKUS HOYER** was born in Lohne, Germany and studied production technology at the University of Applied Sciences and Arts Hannover, where he obtained his degree in 2001. He worked for half a year for Volkswagen before moving to the University of Applied Sciences and Arts Hannover, where he is now a research assistant at the Forschungsschwerpunkt AUBIOS. He is researching for his PhD at the University of Glamorgan. His e-mail address is: markus.hoyer@mbau.fh-hannover.de.

**CARSTEN HORN** was born in Hannover, Germany and studied process technology at the University of Applied Sciences and Arts Hannover, where he obtained his Dipl.-Ing. degree in 1988. He received his PhD in Chemical Engineering from the University of Glamorgan in 1994. He is working for the University of Applied Sciences and Arts Hannover since 1994, carrying out research in various aspects of process simulation, CFD and CAD, currently with the Forschungsschwerpunkt AUBIOS. His e-mail address is: carsten.horn@mbau.fh-hannover.de.

**GIULIANO PREMIER** was born in Treviso, Italy and graduated with a BSc (Hons) Mechanical Engineering from The Polytechnic of Wales, after which he worked as a design/development Engineer in sonar systems. He accepted a senior lectureship at the University of Glamorgan in 1989 and subsequently received a PhD in the Control of Anaerobic Digestion for the same institution in 2003. His e-mail address is: gcpremier@glam.ac.uk

**REIMAR SCHUMANN** was born in Stuttgart, Germany and studied technical cybernetics at the University of Stuttgart, where he received his Dipl.-Ing. degree in 1976. He received his PhD degree in Control Engineering from the Technical University Darmstadt in 1982. From 1983-1989 he worked for VDO Mess- und Regelungstechnik in Hannover as R&D manager Process Control Systems. In 1989 he became Professor in Control Engineering at the University of Applied Sciences and Arts, Hannover, where he is currently serving as head of the Forschungsschwerpunkt (research centre) AUBIOS. His e-mail address is: reimar.schumann@mbau.fh-hannover.de.

# SIMULATION OF MANAGING THE GENERATION OF ECOLOGIC ENERGY IN LOCAL ENERGY MARKET

Eugeniusz Sroczan  
Poznan University of Technology  
Institute of Electric Power Engineering  
ul. Piotrowo 3A, 60-965 Poznan, Poland  
sroczan@put.poznan.pl

Andrzej Urbaniak  
Poznan University of Technology  
Institute of Computing Science  
ul. Piotrowo 3A, 60-965 Poznan, Poland  
urbaniak@put.poznan.pl

## KEYWORDS

Power system, energy management, simulation, neural network, ecologic energy, decision support

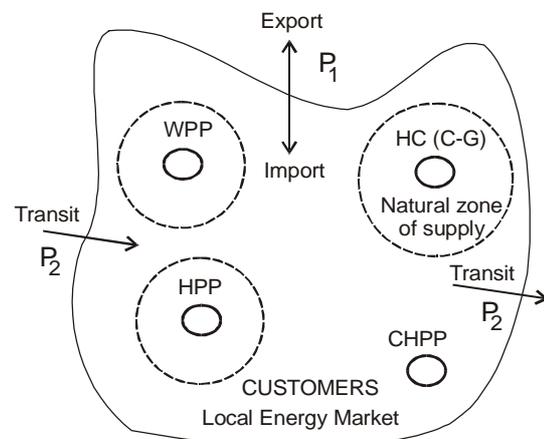
## ABSTRACT

The main aim of this paper is to describe strategy of decision taken by power system manager to minimize costs of energy in terms of spot market. The proposed attempt takes into consideration some problems of ecologic sources of energy, which are committing with conventional power plants. In the given circumstances of demand and load of power plants and networks simulation of expected values of energy is realized in an interactive mode. There are discussed some results of the simulation for the model of power system, including conventional thermal power plants, hydropower plants and wind farm. Neural network technology is applied to define load frame for set of committing thermal power units, hydropower plant and wind power plant.

## ECOLOGIC SOURCES OF ENERGY AND LOCAL ENERGY MARKET

Sources of renewable energy effect the load of committing thermal units in accordance with their mode of operation. Hydro-plants cover the peak load or base load of power system, that mode of operation depends on disposed volume of water inflow. Complex of wind turbines effects equally when velocity of wind is proper, but the power system is loaded additionally when energy of wind exceeds rated values. Local Electric Energy Market (LEEM) allows producers (sources) and providers (networks) to sell energy to consumers nearby the sources. The commitment and competitions among producers and providers (transmission and distribution networks) should assure the interests of energy consumers. The crucial problem is to balance between energy production and its demand with regard to stated prices and power flow in a local

power system expecting varied set of energy sources. The main aim of this paper is to simulate decision making strategy of power system manager (operator) in order to minimize costs of energy in terms of varied weather, different customers behavior and influence of spot market. The proposed attempt is based on fuzzy neural networks and algorithm applied to calculate values of energy in the given circumstances of demand and load of different kinds of power plants and networks.



Figures 1: The Structure of Local Energy Market–Energy Sources; WPP-wind power plant, HC(C-G)-heat central (co-generation mode), HPP-hydro power plant, CHPP – conventional heat power plant

## SIMULATION OF COMMITMENT OF ECOLOGIC ENERGY SOURCES

Electric power demanded by final consumers of energy reflects changes in the level of generation, transportation and distribution costs (Chowdhury at al. 1990, Ringlee and Wiliams 1963). Determined and undetermined volume of power demand in the power system (PS) involves the procedures of peak load covering by set of power units. The set of committing

power units, operated in an optimal mode, effects the real costs of transaction made on the energy stock (spot market or balancing market). The response of final customers and energy distributors on varied costs of energy depends on tariff's prices. In case of increasing costs of energy the power demand will be decreasing in the near future. Therefore the income of energy provider does not change, however the costs of purchasing sales are higher. Fluctuation of hourly loads should be minimized as low as possible to obtain the optimum constraints of energy generation (Chowdhury at al. 1990, Sroczan 1996, Baltierra at al. 1998), but it is only possible when energy management system (EMS) is applied by consumers and demand side management (DSM) is operated by energy providers respectively. Therefore the power system manager should carefully balance the habits either of energy consumers and distributors or power plants and network requirements (Sroczan 1999). If procedures of local optimization are omitted, costs of delivered energy would be increased more then necessary, from theoretical point of view.

The structure of local energy market (LEM) in many countries is similar - some levels of competitions among conventional heat power plant (CHPP), transmission and distribution networks (TDN) allow the producers (sources – CHPP, HPP, WPP and H-C) and providers (TDN) to sell the energy at the actual market price. In developing countries this kind of energy management effects the possibility of growing either power plant capacity or power and energy flow in networks (providers and distributors) - from point of view of modernization or developing the production possibilities. Some aspects of these very involved and essential problems, especially including PP with limited energy production, are discussed in this paper.

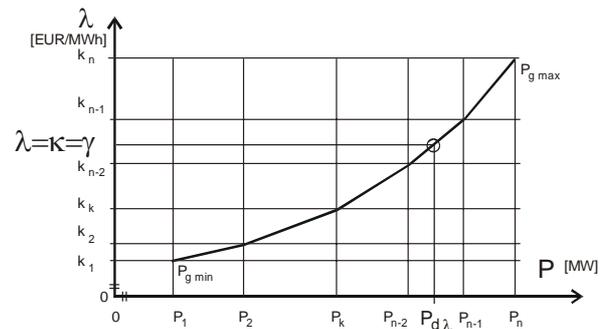
Limitation of energy generation refers especially to ecologic energy, which additionally depends on wind and hydro resources. There are decision support system and classical load dispatch algorithm applied to solve the problem of cost's minimization of generated energy. Obtained quality of power and energy balance effects the expected income of energy providers.

The proposed model of managing the LEM (Sroczan 1999) is based on maximization of expected delivery income (EDI) calculated for the  $k$ -th energy provider, which is described as  $\Omega$  – a set of committing power plants including renewable sources and final distributors. Expected income, calculated for this set considers following relationships: power plant, energy stock, provider, distributor, consumer, in  $j$ -th hour of considered time period of calculation T:

$$\max\{EDI_k\} = \max \left\{ \sum_{i,k \in \Omega} \sum_{j \in T} \left[ \begin{aligned} & (a_{jk} + b_k P_{ijk} + c_k P_{ijk}^2) + \\ & - (a_{ij} + b_i P_{ijk} + c_i P_{ijk}^2) \end{aligned} \right] \right\} \quad (1)$$

$$\sum_i P_{ij} = P_{PSj} + \Delta P_{FLij} \quad (2)$$

where:  $EDI_k$  - income of  $k$ -th energy provider which is contracted power and energy from  $i$ -th power source (PP), belonging to set  $\Omega$ ;  $a_i, b_i, c_i, a_k, b_k, c_k$  – characteristic's factors describing the relationships between hourly exploitation costs of  $i$ -th power plant and  $k$ -th distributor, defined for set  $\Omega$ ;  $P_{ds}$  – demanded power of PS;  $\Delta P_{FL}$  – power losses in given network branch  $ij$ .



Figures 2: Calculation of Incremental Cost for the Set of Conventional Thermal Power Plant, Hydropower Plant and Wind Farm;  $\lambda$  – increment of generation costs which is balancing the power demand  $P_d=P_g$ ,  $\gamma$  – cost substituting the increment of water consumption,  $\kappa$  – wind power cost substituted by current cost of power generation in PS and balancing P<sub>gk</sub>.

Equation (2) describes constraints of power balance in PS including power losses in transmission and distribution networks. The structure of considered PS is shown in the fig.1. Sources of energy defined as ecologic are classified as wind (WPP) and hydro power (HPP) plants. Both class of energy sources depend on weather constraints. All sources are operating with respect to natural distance between plant and consumers. It means that long distance transmission of energy is unprofitable but sometimes is necessary. The final price of delivered energy is calculated as average value, defined in the contract between power plant and wholesale energy provider, but in reality it should be calculated or modified in the real-time mode by PS energy operator office in accordance with weather constraints.

The equation (1) includes also some impact of ecology constraints in form of decrement or increment of the costs yielded by the operation of hydroelectric plants, transmission and distribution networks as well as environment protection costs.

In the fig. 2 there are shown the incremental costs of power generation in PS. Relationships among the thermal, wind and hydroelectric unit costs of generation are shown as the increment related to actual PS power demand. The varied value of  $\gamma$  and  $\kappa$  enables balancing the generated and demanded power with regard to kind of renewable energy source and power generation limits

(2) The optimal increment of power generation costs is equal:

$$\lambda_t = \frac{\partial C_{i,n,t}}{\partial P_{i,n,t}} = \gamma \frac{\partial W_{j,n,t}}{\partial P_{j,n,t}} = \kappa \frac{\partial C_{k,n,t}}{\partial P_{k,n,t}} \quad (3)$$

for all committing units.

Each of the increments of power  $P_{i+1} - P_i$ , for  $i=1,2,\dots,n$  describes a part of units' characteristics for generation costs or total plant costs. Comparison of generation costs of different kinds of power sources allows to sort the set of committing units with regard to increment of the  $\lambda$  (or  $\gamma$  or  $\kappa$ ) coefficient.

The essential problem of the price of power and energy with regard to ecologic sources depends on PS manager decision and LEEM operator (Sroczan 1996, Malko 1997). If the decision is optimal the costs will be fulfilled in each time  $t$  the relationship:

$$C_t \rightarrow \min \left\{ \sum_{i=1}^n C_i(P_{gi}) \right\} \quad (4)$$

where:  $C_i(P_i)$  – cost of generation in  $i$ -th PP at partial load of  $P_g$ .

The proposed criterion function based upon decision algebra, supports the decisions made for optimization of power flows in considered PS to meet the constraints of contracts:

$$EMV(C_i) = \sum_{i=1}^n EDI(S_i) \cdot P(S_i) \quad (5)$$

where:  $EMV(C_i)$ - expected monetary value of decisions in a given probability  $P(S)$  of power demand;  $S_i$  - probability of achieving the real value of the given power demand.

### NEURAL NETWORK AS A TOOL FOR SUPPORT MANAGER'S DECISION

Structure of neural network consists of balanced adder, dynamic module realizing the linear function and non-dynamic module, which is performing a non-linear function. Task of identification is based on estimation coefficients and also the weights of the neuron's network based on the error processing  $e(t)$  between the output of the model  $y^m(t)$  arrangement and the output of the object  $y(t)$ . The module of the linear dynamic function is described by the transfer function, in this case realizing given relations between inputs  $v_i$  and outputs  $y_i$ .

The AI network, implemented in the loop of manager decision support system, consists of three layers within one hidden layer. The first layer is similar to input signals, which are obtained from real-time system. Evaluation of considered parameters is realized

in the hidden layer. Output signals of neuron network describe the current state of discussed power system, defined for chosen parameters, negligible for operator's decision. Process of training the NN is described as iterative tuning of coefficient  $w_i$  of each inputs  $x_i$  to meet the value  $y(k)$  at the output of NN. Relationship between input and output is established by defined constraints.

The weight coefficient  $w_i$  is calculated, for the  $k$ -th step of learning as:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) \quad (6)$$

where:  $i, j$  – number of joined neurons,  $k, k+1$  – old and new number of  $w_i$ .

The process of learning may be tuned by operator, each input vector  $x(k)$  is compared with stated vector  $s(t)$  and they are defined as:

$$x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T \quad (7)$$

$$s(k) = [s_1(k), s_2(k), \dots, s_m(k)]^T \quad (8)$$

The data learning NN are given for each group of input vector  $x(k)$  and  $s(k)$ , for  $k=1, 2, \dots, p$ , where  $p$  describe the learning pattern. For each pattern the error  $e(k)$  is defined as:

$$e(k) = y(k) - s(k) \quad (9)$$

where:  $y(k)$  is a current response of NN for input signal  $x(k)$ .

The pattern vector  $s_p$  is calculated a priori from data imported from SCADA system of PS for given states of committing CHPP, WPP and HP. It is possible to build the pattern matrix for the simulator structure. The procedure of learning the NN by the operator-expert enables to minimize the error gained by the NN. For  $m$  neuron output and  $p$  patterns the error is described as:

$$E = \frac{1}{2} \sum_{k=1}^p \sum_{j=1}^m e_j^2(k) \quad (10)$$

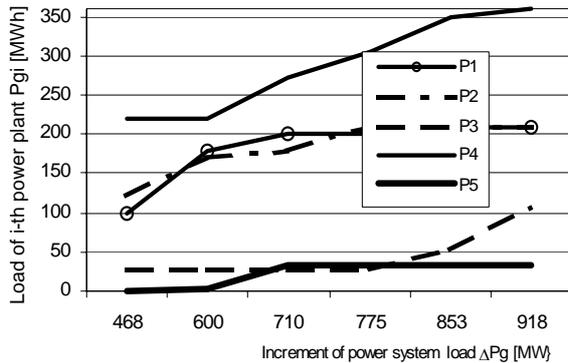
The assessed impact of PS manager within the strategy of load the hydro-power plant depends on value of  $\gamma$ - system equivalent price of water, volume of water that is consumed in each hydro-power plant and  $\kappa$ - wind power cost.

### RESULTS OF SIMULATION

The assumed principles of commitment include time and power gain without limits and occurred from short and long planning term. Power plant framework

optimization is made using linear or dynamic programming. The impact of manager's decision on natural environment is defined by different values of  $\gamma$ , and  $\kappa$  coefficient that are substituting cost of wind power and the increment of water consumption in accordance with increment of generations cost in CHPP. The  $\gamma$  coefficient is responsible for hydropower plant allocation in the queue of load covering in accordance with disposed water volume. The  $\kappa$  coefficient is responsible for the WPP scheduling. There are considered the procedures of cost optimization: weather and load forecast of the PS on different time horizon, power demand as the response on DSM policy, actual and predicted states of PS. In the Table 1 there are shown results of simulation for assumed values of PS power demand and different value of  $\gamma$  and  $\kappa$ - system values (price) of ecologic energy.

The assessed impact of the PS manager on the strategy of load the hydro-power plant depends on value of water system equivalent price, defined for water which is consumed in each hydro-power plant. Loading the wind farm depends on weather and PS demand, too. The value of  $\kappa$  as well as  $\gamma$  change the power range of generated energy, so the hydropower plants and wind farms are committing with thermal plants.



Figures 3: Simulated Load Characteristics for Given Set of Power Plants – CHPP, HPP, WPP;  $\gamma=0,2$  [\$/m<sup>3</sup>] and  $\kappa=0,0223$  [\$/MW]

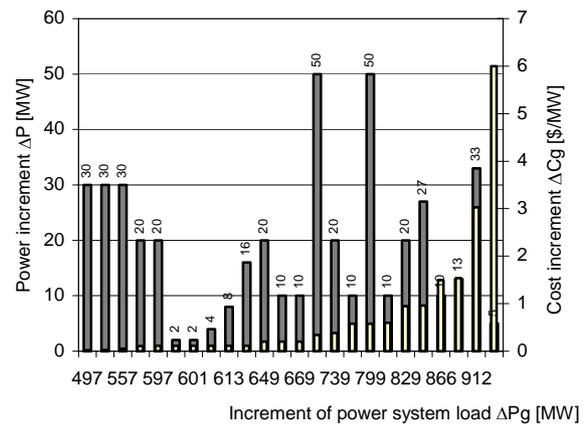
The value of EMV calculated for each PP and for both values of  $\gamma$  and  $\kappa$  shows that for low  $\gamma$  hydro power plant the load comes as the first one. If the value of  $\gamma$  is increasing the load is covered by power plant WPP. It means that the cost of emission the combustion gases are similar because of steady load of CHPP. Therefore the decision of PS manger effects the natural environment in two aspects: if the energy is generated by HPP or WPP the renewable energy is utilized and total amount of CO<sub>x</sub>, SO<sub>x</sub>, NO<sub>x</sub> and H<sub>2</sub>O is decreased.

Some results of simulation the manager's impact for energy generation cost are shown in the tab. 1.

Table 1: Cost of Energy Generation for the Model Power System

Case number	Value of coefficient		Cost of energy
	$\gamma$ [\$/m <sup>3</sup> ]	$\kappa$ [\$/MW]	$C_g$ [\$/h]
1	0,2	1	2101,4
2	0,2	0,0223	1868,9
3	0,2	1,5	2181,4
4	1,5	1,5	5804,4
5	1,5	1	5648,0
6	1,5	0,0223	5311,5
7	0,0223	0,0223	1345,8
8	0,0223	1	1526,2
9	0,0223	1,5	1601,2

The considered model of PS includes: 85% of rated power in CHPP, 11,5% in HPP and 3,5% in WPP.

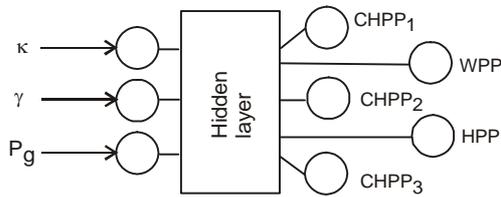


Figures 4: Results of Calculation the Incremental Costs for the Set of CHPP, HPP, WPP;  $\gamma=0,2$  [\$/m<sup>3</sup>] and  $\kappa=0,0223$  [\$/MW];  $\Delta C_g = \{0, 6, 2\}$  [\$/MW] – increment of generation costs which is balancing the power demand  $\Delta P_{g,\epsilon} \in \{468, 918\}$

The simulated power system consists of three sets of aggregates; the first one contains the thermal sources of energy (CHPP), the second one hydroelectric (HPP) and wind units (WPP) as well power-limited (WPP) as energy-limited (HPP).

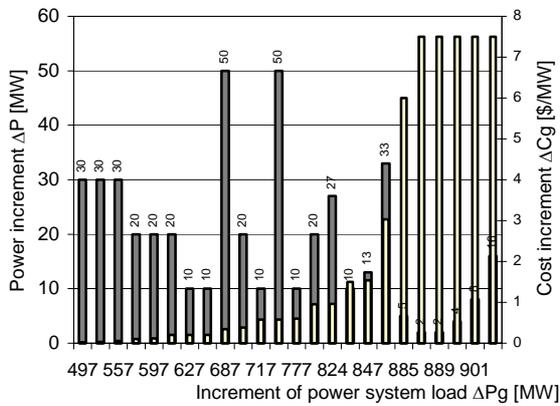
In the figures 3. and 4. there are shown the schedule of loading the modeling power plants and allocation of load increment  $\Delta P_i$  versus power system load  $P_g$ . It is assumed that increment of cost of the power  $P_d$  generated in PS approximate  $\lambda$  (fig. 2) for the conventional thermal plant (CHPP). The cost's increments of energy generated by ecologic sources are calculated with regard to an equivalent cost factor  $\gamma$  for hydropower plants (HPP) and factor  $\kappa$  for energy

generated by wind farms. The load schedule is obtained by using a neural network (fig. 5.) with three inputs – load’s and manager’s preferences; and outputs for each unit in the given power plants.



Figures 5: The Structure of Neural Network Dedicated to Define the Power Unit Load

The figures 6. and 7. presents the results obtained for  $\gamma=0,2 << \kappa=1,5$ . The HPP are loaded earlier then WPP and the cost groves about 16,7% of the previous one.



Figures 6: Results of Calculation the Incremental Costs for the Set of CHPP, HPP, WPP;  $\gamma= 0,2$  [\$/m<sup>3</sup>] and  $\kappa= 1,5$  [\$/MW];  $\Delta C_g = \{0, 7,5\}$  [\$/MW] – increment of generation costs which is balancing the power demand  $\Delta P_{g\lambda} \in \{468, 918\}$

Quite different preferences of energy manager are illustrated in the figures 8. and 9. The relation between  $\gamma=1,5 >> \kappa=0,0223$  shows that WPP set is loaded as the first one, before the HPP units, but the cost of energy increases considerably – about 243,5% in relation to case 3 (tab. 1.).

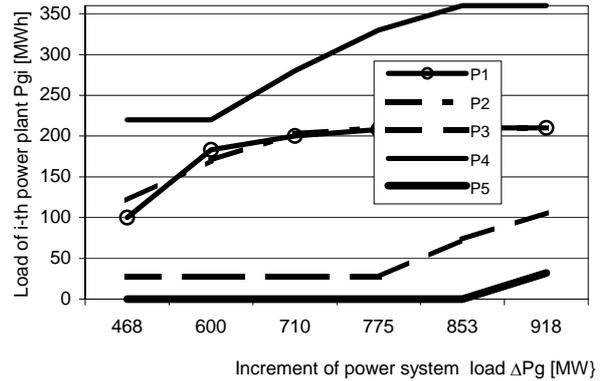
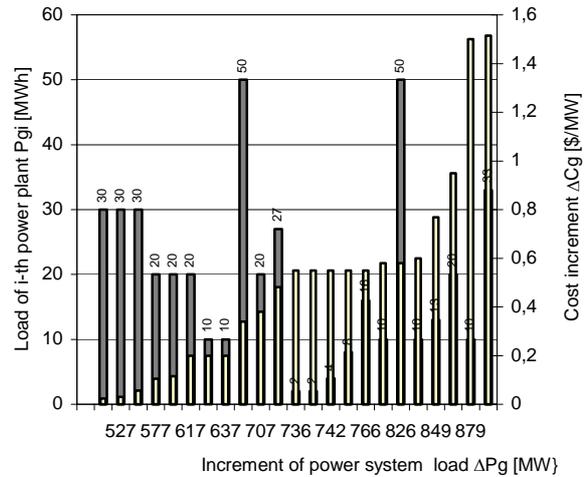


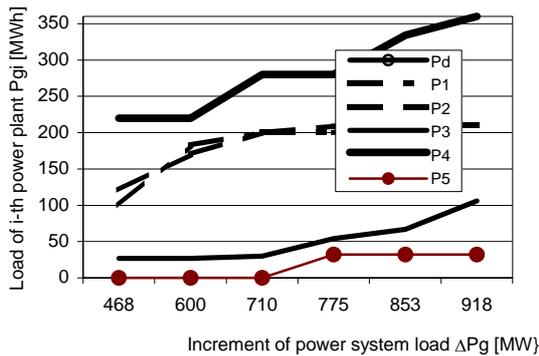
Figure 7: Simulated Load Characteristics for Given Set of Power Plants – CHPP, HPP, WPP;  $\gamma= 0,2$  [\$/m<sup>3</sup>] and  $\kappa= 0,0223$  [\$/MW]

The third class of units (HC-CG) is generating power and heat in co-generation mode. Delivery costs are taken into account in the form of the simulated losses for the given network configuration. Transmission and distribution networks’ topology must to be taken considered by using penalty factors (Chowdhury at al. 1990, Sroczan 1999, Baltierra at al. 1998). The losses are comparable because of small power variation.



Figures 8: Results of Calculation the Incremental Costs for the Set of CHPP, HPP, WPP;  $\gamma= 1,5$  [\$/m<sup>3</sup>] and  $\kappa= 0,0223$  [\$/MW];  $\Delta C_g = \{0, 1,5\}$  [\$/MW] – increment of generation costs which is balancing the power demand  $\Delta P_{g\lambda} \in \{468, 918\}$

As it is shown in the figures 3., 7. and 9., the load of CHPP varies within the necessary range to cover the load in case of shut down the WPP set and discharge of HPP or WPP units, as the result of the manager’s decision. It is assumed that the units of WPP should be shut-down without additional cost of operation.



Figures 9: Simulated Load Characteristics for Given Set of Power Plants – CHPP, HPP, WPP;  $\gamma=0,2$  [\$/m<sup>3</sup>] and  $\kappa=0,0223$  [\$/MW]

## CONCLUSIONS

To solve the problem of LEEM operating there are some procedures developed within the field of simulation the expected costs of energy supply for the given wholesale providers and power plants.

The range of manager's decision is described by acceptable values of  $\lambda$ ,  $\gamma$  and  $\kappa$  coefficients, balancing the load of committing units with regard to ecologic sources of energy. Due to varied weather constraints the energy from this kinds of sources is more expensive than energy from conventional power plants.

The optimal operating the LEEM is possible with respect to market rules by using the proposed simulation procedures.

The described system simulates routines of the manager of power system dedicated for checking the broker's decisions in case of buying and selling with rational level of risk

The problem of validation the manager's decision takes into account both terms of analysis: the technical and economic effects. Considering the ecologic – renewable sources of energy supplying the local markets it is possible to formalize the impact of manager's decision on natural environment.

Distributed generation allocated in the hydropower and wind plants effect the flow of energy in the power network, especially in weather disturbances. To preserve the set of thermal units the commitment of hydropower and wind farms is necessary.

Using the described process of simulation it is possible to obtain the minimal costs of energy and optimal allocation of the given energy resources with respect to environment protection rules and PS demand.

## REFERENCES

- Baker G.C. 1999. The wave of deregulation: Operational and design challenges. IEEE Power. Eng. Rev. Nov.
- Baltierra A.E., Moitre D., Hernandez J.L., Aromataris L. 1998. Simulation of an Optimal Economic Strategy of a Wholesale Competitive Electric Energy Market. in *Proc. of 10<sup>th</sup> European Simulation Symposium*, Nottingham, England 1998. p. 255-259.
- Chowdhury, Saifur Rahman 1990. A Review of Recent Advances in Economic Dispatch, IEEE Trans. on PWRs, Vol.5, No.4, Nov., pp.1248 -1259.
- Malko J. 1997. Optimization of power generation structure on local market, Rynek Energii No. 5 (12), (in polish).
- Ringlee R.J., Williams D.D. 1963. Distribution of system loads by the method of dynamic programming. *Power Apparatus and Systems*. No 64 p. 615
- Sroczan E.M., Urbaniak A., Simulation of routines of power system manager's decision effecting the natural environment, in: *Simulation in Industry*, Giambiasi N., Frydman C., (eds.), SCS Europe BVBA Publ. Marseille, (France) 2001 (488-492)
- Sroczan E., "Application of Artificial Intelligence to Algorithms of Power Plant Identification with the Purpose of Load Dispatch", In *Proc. of 8<sup>th</sup> European Simulation Symposium ESS '96* " Genoa, Society for Computer Simulation International, Genoa 1996. p. 592-596

**EUGENIUSZ SROCZAN** was born in Poznan, Poland, went to Poznan University of Technology. He is working as an assistant professor at the Poznan University of Technology (PUT). He obtained from PUT a master and engineering degrees in area of industry automatic and the Ph.D. in area of electric power system engineering. Author and co-author of papers on power system economic operation, energy management systems in industry and automation of the water and waste-water treatment plant. Since 1984 year he is the President of Polish Electricians Society at the PUT Branch. Member of the SCS International since 2001. His E-mail address is: sroczan@put.poznan.pl

**ANDRZEJ URBANIAK** is working as professor at the Poznan University of Technology (PUT). He obtained from PUT a master and engineering degrees in area of industry automatic. The Ph.D. degree he obtained from Technical University of Lublin and after Ph. D. degree he obtained at TU of Warsaw. Author and co-author of papers on operational research, computer science in environment preservation, management systems in industry and automation of the water and waste-water treatment plant. Member of the SCS International since 2001. His E-mail address is: urbaniak@put.poznan.pl

# COMPONENTS FOR THE ACTIVE SUPPORT OF THE ANALYSIS OF MATERIAL FLOW SIMULATIONS IN A VIRTUAL ENVIRONMENT

Bengt Mueck  
Wilhelm Dangelmaier  
Matthias Fischer  
Heinz Nixdorf Institute, University of Paderborn  
Fürstenallee 11  
33102 Paderborn, Germany  
{mueck, whd, mafi}@hni.uni-paderborn.de

## KEYWORDS

Visualization of production processes, Human Computer Interface

## ABSTRACT

Simulation and visualization are well-known methods for the understanding and analyzing of manufacturing processes. In visualizations of manufacturing processes the viewer can move around freely and unguided. Thus knowledge and conclusions are only acquired on a random base. This article outlines a system and methods that support the viewer to keep an eye on noticeable/significant processes/points in the material flow simulation and to optimize these processes.

This article describes the development of a tool, which enables the viewer of a simulation to interactively improve significant production processes. The viewer moves in a virtual 3D-environment (walkthrough system) and can acquire automatically calculated indications for significant processes. At the same time the simulation should simulate significant objects in a more detailed way. If the viewer is interested in a significant process, he is automatically guided to the relevant place where he can examine the critical situation by interference in the simulation. Since the critical moment is in the past and is thus already missed by the viewer, the viewer is able to rollback the simulation to a time before he entered the simulation.

## OBJECTIVES

Visualizations of manufacturing processes typically allows the viewer to move around freely and unguided. This leads to random process mostly based on the experience of the viewer to acquire knowledge and conclusions. A tool is required which enables the viewer of a simulation to interactively detect and improve significant production processes. The viewer is able to move in a virtual environment and acquire (semi-) automatically calculated indications for significant processes. If the viewer is interested in a significant process, he is automatically guided to the relevant place where he can examine the critical situation by interaction with the simulation. Since the critical moment is in the past and there-

fore already missed by the viewer, the viewer is able to rollback the simulation to a time before the significant process occurred.

Such a system has to possess the following characteristics: there have to be methods for the automatic detection and rating of significant points and processes in a simulation. Important and unimportant points are differentiated in a pre-filtering process in order to facilitate the choice of huge numbers of significant points. There have to be interaction techniques with the user. Methods to guide the viewer in a virtual environment have to be available. Suitable algorithms for the visualization of points with differently high significance in a room must support the visualization in a walkthrough-system. Since the attention of the viewer is drawn to significant points, both the position of the viewer and the significance of the object serve as indicator for a high detailing of the simulation. A rollback in the simulation as well as in the visualization has to be made available for the viewer. Suitable algorithms for the visualization of points with differently high significance have to take into account the special characteristics of simulation environments in a walkthrough-system.

The virtual scene or its 3D-models of a dynamic simulation environment are generally too complex to visualize them in real-time (Möller and Haines 1999). Typically approximation methods are applied in order to visualize them with a low loss of quality in real-time in a walkthrough-system and to enable an easy navigation for the viewer of the virtual scene. Two basic approaches are the complexity reduction (approximation) (Luebke 1997 and Garland 1999) and the calculation of hidden objects (Visibility-Culling) (Durand 2000). Exact information about the priorities in the virtual scene is made available by the specific simulation requirements concerning the significant points. The simulation knows that the virtual objects at significant points are very important. These significant points can be at any place in the scene. This knowledge enables the approximation algorithms to use a high rendering quality or a high rendering power for the simulation-models of significant points and accordingly to neglect the other parts of the virtual scene. Such approximation algorithms and data structures use the

specific characteristics of virtual scenes in simulation environments and enable a higher rendering quality and rendering power than a system in which no assumptions of the scenes can be made.

Objects with a high significance have a big influence on the simulation. Thus, a more detailed view is reasonable out of the perspective of the user as well as of the simulation. Thus methods are developed which carry out the simulation of these objects in a more detailed way (like Dangelmaier, Fahrentholz and Mueck 2002) and therefore also more accurate.

## **STATE OF THE ART**

Today material flow simulations offer installed 3D-modules (Klingstam and Gullander 1999). Examples are simulation tools such as QUEST by Deneb, Taylor ED by Enterprise Dynamics (Nordgren 2001), or eM-plant by Technomatix with integrated, virtual environments. These systems offer only conditionally the possibility to dynamically influence and analyze a scene. Furthermore they take place on the computer on which the simulation runs. The simulation has to share the available computing time with the visualization. This decreases the speed of execution of the simulation and it also reduces the rendering quality of the visualization. It is especially difficult for the visualization to forecast the fluctuations of the required computing time of a simulation. Often a constantly high refresh rate is not possible with many polygons.

These systems can also analyze material flow simulations. An active support does not take place. If the viewer moves in the relevant system, he cannot experience critical processes which take place in his back and are not visible for him. User guidance to significant points is not known. The locating of critical objects is thus left to the experience of the viewer or to chance.

## **SYSTEM DESIGN OF A 3D-SIMULATION ENVIRONMENT**

The draft at the beginning results in many requirements the system has to meet. There are requirements for the modeling and simulation as well as for the visualization and for the rendering methods in the 3D-system (walk-through-system).

### **Experimentation platform**

There has to be a solid experimentation platform with allows the user to look at the simulation in a virtual environment and also to experiment interactively. For this purpose a simulator has to be chosen and connected online to a visualization component.

Thereby a number of interactions for the modification/parameterization of the model are necessary for the

visualization. The modifications/parameterizations carried out in the visualization have to be processed correctly in the simulator. Especially the rollback demands basic methods of the simulator.

### **Identification, rating, and visualization of significant objects**

Not all processes in a simulation are equally important for the user. He wants to know where irregularities and problems occur (e.g. a blocking machine or empty stock). Such processes are significant for problems in the manufacturing. In the here-described system significant simulation processes should be automatically identified and explained to the user.

For this purpose significant points have to be identified. Suitable methods for the support of the modeler to set up sensors for significant processes have to be integrated. Beside the semi-automated modeler supportet sensor approach to identify significant processes a fully automated method is under development.

A proper rating of the significance of the objects has to be made. If an error, which puts many objects in the condition "significant", occurs at one point of time, then only the actuator of the error should to be indicated to the user. For this purpose, the number of significant points has to be filtered. If the user has already solved the problem or prioritized a spot, then the significant incidences at this object are to be reported preferentially.

The number of significant points for the user is put in the foreground by an adjusted visualization. At the same time the visual view of other objects in the virtual scene, which are not significant, has to be reduced. Therefore appropriate visualization methods are to be applied (e.g. semitransparent objects).

### **Guidance to significant points**

In today's approaches there is no real guidance of the user. Thus the identification of critical processes and connections is highly dependent on the experience and the intuition of the user. It takes places in disorder and rather coincidentally. In our system the direct guidance allows the user to acquire more efficient, structured and thus faster the relevant information and go to places that have a strong influence on the simulation (see figure 1).

The user should have the option to be guided to „significant“ points and to change or at least parameterize the simulation model anew. Therefore suitable interaction techniques are developed.

If the user makes use of the option to be guided to a significant point, he is directly transported to this point or the route from his current position to this point has to

be determined. In order to give the viewer the impression that he moves in a real hall, he moves along usual traffic route. Machines and walls can cause an unsatisfied optical view and disorientation by occlusion. If visible objects lie between the significant object and the position of the viewer, the viewer has to go round these objects. For this purpose traffic routes have to be mod-

eled and a suitable route has to be automatically determined in the model (see figure 2). The user should be guided by markings or guided automatically. Along his way both the simulation and the visualization should to take place highly detailed.

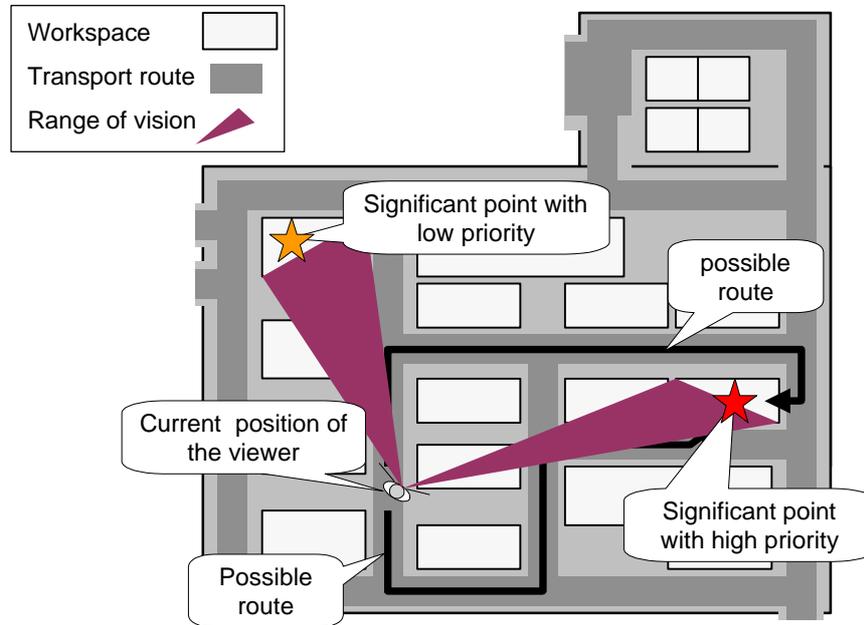


Figure 1: Automatic calculation of the route

### Changing the Resolution of the Simulation

Simulations in a high resolution require a long execution time. In large models with a high resolution this can lead to an execution, which is slower than in real time. The use of parallel computing systems provides more computing power and thus permits an acceleration of the computation. The problem is thereby only shifted to another level.

In order to give the viewer the feeling of a high detailing at all places of the simulation, a method was developed whereby only the part of the simulation, where the viewer stands, is carried out highly detailed (Dangelmaier, Fahrentholz and Mueck 2002). Places that are farther away from the viewer are simulated on a rougher level. If the viewer changes his position, a higher detailing in the simulation is automatically updated. Thus the viewer gets the impression of a continuous high detailing without a high computing power. Bigger simulations can be experienced. But an inexact calculation of those parts, which are not located within the visual focus of the user has to be accepted. This leads to inexact simulation results.

Since both the attention of the viewer to significant objects, and the influence of significant processes on the simulation are increased, the level of detailing of these objects should be determined by the viewer position and the significance of the objects. This has to be integrated in the experimentation platform.

### Multi-Point-Approximations for virtual scenes with significant points

In the following the system requirements for the walkthrough-system in the area data structure and real-time-algorithms are explained, which are necessary for the visualization in a virtual scene:

- Real-time navigation: a walkthrough-system is needed which allows for a free navigation in the scene, i.e. the rendering-algorithms have to be able to calculate pictures of the virtual scene with at least 10-20fps (frames per second).
- Dynamics: our scene must be dynamic, i.e. objects can be inserted or deleted by the simulator or the user at any time.
- Kinematics: typically the scene has high kinematics. Many objects (in an extreme case all objects) move with high speed to any places in the virtual scene.

- Complexity of the scene: the models of the virtual scene result from exported models of CAD-systems. Typically they have a high complexity of polygons. Thus the virtual scene consists of several gigabytes of scene data, which does not fit as a whole in the main terminal or cannot be visualized in real-time without application of expert methods.

The last requirement for the walkthrough-system makes the solution for the other three requirements more difficult. The virtual scene can only be modeled by means of 3D-approximation methods. The workflow can be organized efficiently by automatic approximation methods, which do not demand an after-treatment of 3D-models. However, the most important questions concern the nature of the required approximation.

The detailing and the approximation quality of the objects in the virtual scene are contrary to the traditional approaches because they are not dependent on the position and the direction of view of the user. Especially positions with significant processes and the way to these

positions serve as a better indicator for a high detailing than the simple distance or the projected size of the object. The significant points in the virtual scene cause a deliberate imbalance of the qualitative visualization.

Multi-Point-Approximations are methods which improve the rendering quality of objects in the range of significant points independent of their distance and which reduce the rendering quality of non-significant points (see figure 2). Thus a limited rendering-time is purposefully used for the parts of the scene, which are interesting for the simulation because the user is interested in these parts. There is no “waste” of rendering-time by a high weighting of close, non-significant objects as it is the case in traditional methods which do not allow to make assumptions for the design of the scene. Approximation methods and rendering-algorithms are required which allow for the purposeful weighting of different independent points of the virtual scene.

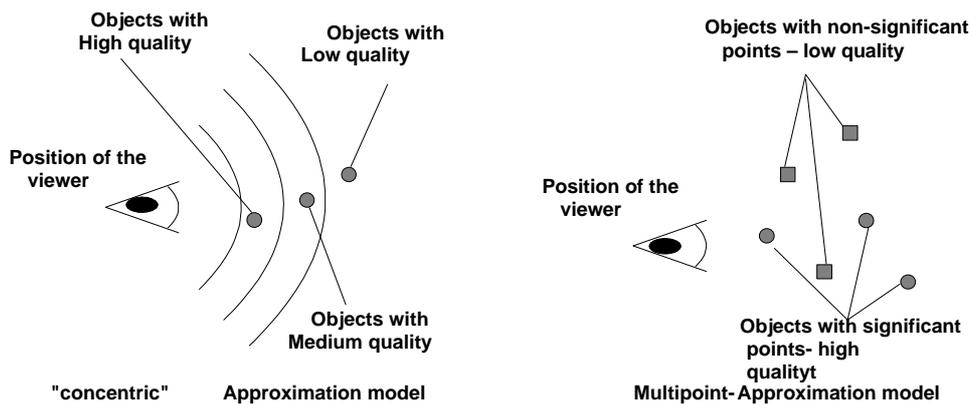


Figure 2: Comparison "concentric approximation model" and Multi-Point-approximation model

Thus the algorithmic problems are to a great extent influenced by simulation requirements. Also requirements for the “visualization + simulation” have to be taken into account. A simulation with randomly arranged, significant points which wants to call the viewer’s attention to these points has to use visualization methods in order to visually emphasize these points. For this purpose, methods such as the dynamic use of “transparency” or the “lowering of non-significant objects” have to be applied (see figure 3 and figure 4). In doing so, there is a change of visibility behavior of the static parts of the scene in a

relatively short time interval. This results again in the requirement for the approximation method and visibility-culling-algorithms of the rendering of the scene to consider the dynamic changes of the topology.

Concluding, the simulation involves many specific, methodical requirements for the data structures and algorithms for the rendering of the virtual scene, which are not all covered by traditional methods in the approximation and visibility-culling field.

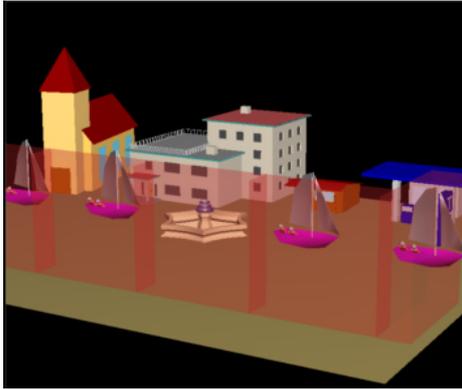


Figure 3: In this setting the fountains and boats are „significant“ points. The wall in the front is transparent.

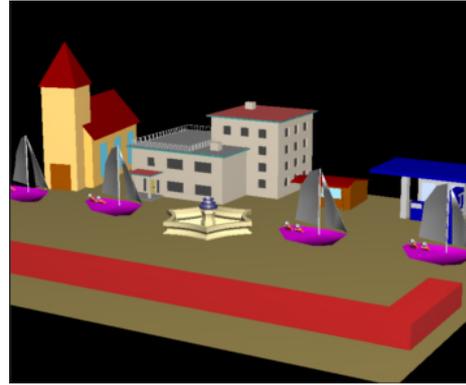


Figure 4: Like figure 3, however instead of transparency only the outline of all objects (here the wall), which obstruct the view on „significant“ points, is drawn.

## CONCLUSION

Today there is not yet an active support for the analysis of material flows in simulations. This article outlines a system, which determines and visualizes objects to the viewer that are significant for the simulation. Both a more detailed simulation and a visualization of these objects take place. Thus, the user is directed to the most important objects of the simulation instead of walking aimless through the virtual environment. These objects are simulated and visualized more detailed. The viewer can concentrate his attention and his interactive optimization on objects, which have a high influence. The analysis and interactive optimization can take place faster and more efficient. The system described in this article is currently under development and should demonstrate the benefits and the feasibility.

## REFERENCES

- Dangelmaier, W.; Fahrentholz, M. and Mueck, B. 2002. Adjusting Dynamically the Resolution in discret simulations. In: Verbraeck, A.; Krug, W. (eds.): Simulation in Industry 14th European Simulation Symposium (ESS 2002), p. 56-61, SCS-Europe BVA, Ghent, Belgium
- Durand, F. 2000. A Multidisciplinary Survey of Visibility. In: ACM SIGGRAPH course notes: Visibility, Problems, Techniques, and Applications
- Garland, M. 1999. Multiresolution modeling: Survey & future opportunities. In: STAR - State of the Art Reports, EUROGRAPHICS '99, EUROGRAPHICS Association
- Klingstam, P. and Gullander, P. 1999. Overview of simulation tools for computer aided production engineering. In: Computers in Industry, No. 38, p. 173-186
- Luebke, D. 1999. A survey of polygonal simplification algorithms. Technical Report TR97-045, University of North Carolina at Chapel Hill, Department of Computer Science
- Möller, T.; Haines, E. 1999. Real-Time Rendering. A K Peters, 63 South Avenue, Natick, Massachusetts 01760
- Nordgren, W. B. 2001. Taylor Enterprise Dynamics. In: Peters, B. A.; Smith, J. S.; Medeiros, D. J. und Rohrer, M. W. (edt.): Proceedings of the 2001 Winter Simulation Conference, p. 269-271

## AUTHOR BIOGRAPHIES



**Bengt Mueck** studied computer science at the University of Paderborn, Germany. Since 1999 he is a research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM at the HEINZ NIXDORF INSTITUTE of the University of Paderborn. His main research interests are logistics systems and tools to simulate those systems.



**Wilhelm Dangelmaier** was director and head of the Department for Cooperate Planning and Control at the Fraunhofer-Institute for Manufacturing. In 1990 he became Professor for Facility Planning and Production Scheduling at the University of Stuttgart. In 1991, Dr. Dangelmaier has become Professor for Business informatics at the HEINZ NIXDORF INSTITUTE; University of Paderborn, Germany. 1996, Prof. Dangelmaier founded the Fraunhofer-Anwendungszentrum für Logistikorientierte Betriebswirtschaft.



**Matthias Fischer** studied computer science at the University of Paderborn, Germany. Since 1995 he is a research assistant at the HEINZ NIXDORF INSTITUTE. His main research interests are computer graphics, real-time rendering algorithms, and distributed computing.

# A DATA DRIVEN APPROACH TO AUTOMATED SIMULATION MODEL BUILDING

Charu Chandra  
Industrial & Manufacturing Systems Engineering  
University of Michigan-Dearborn  
4901 Evergreen Road  
Dearborn, MI 48128, USA  
E-mail: charu@umich.edu

Jānis Grabis  
Department of Operations Research  
Riga Technical University  
Kalku 1  
Riga, LV-1658, Latvia  
E-mail: grabis@itl.rtu.lv

## KEYWORDS

Automated model building, simulation modeling, multi-stage manufacturing system, data model.

## ABSTRACT

This paper develops a simulation model building approach aimed at reducing model development efforts. The model building approach is intended for modeling of multi-stage manufacturing systems. A data model is used for representing the manufacturing system in the standardized manner. The data model is created from multiple raw data sources. A simulation model is automatically generated on the basis of the predefined template using information provided in the data model. Different manufacturing systems can be modeled by changing information in the data model. The automated generation allows avoiding model building errors caused by the large scale of the modeling problem. The model building approach is applied to study scheduling at initial stages of the manufacturing process of an automotive company. The automated approach is suitable for the problem because the system contains large number of similar objects and the company operates several similar systems.

## INTRODUCTION

High degree of complexity is the characteristic of a majority of large manufacturing systems. The complexity is caused by interactions among multiple-products, production stages and processing technologies, and dynamic and stochastic behavior of these systems. Therefore, simulation is widely applied for modeling and analysis of manufacturing systems (e.g., Bhaskaran 1998, Petrovic 2001). However, expensive model building is one of the main drawbacks of simulation (Law and Kelton 2001). High model building expenses may be especially preventive, if simulation is applied in preliminary studies of the system or as a supplementary tool. For instance, a simulation model can be used to examine analytical models under conditions not observable on the current system (Ignall et al. 1978).

Several approaches aimed to reduce expenses of the simulation model building have been proposed in the literature. Baker (1997) describes a methodology for incorporating classic operations research models into

simulation models. Numerical algorithms are implemented using high-level programming languages. A simulation software user can insert these routines in the model and manipulate them through a user-friendly interface. Swaminathan et al. (1998) develop an agent-based simulation model building approach, where predefined agents are responsible for performing standard functions of supply chain management. A supply chain simulation model can be developed by assembling these agents. Son et al. (2000) describe an initiative by the National Institute of Standards and Technology to develop neural libraries of simulation model components in order to reduce simulation model building efforts. The standardized model components are used to develop a simulation package independent model. A package specific translator is used to generate an executable model. Werner and Weigert (2002) describe integration between a specific manufacturing and planning simulation model and an enterprise resource planning (ERP) system. The simulation model is continuously recreated using the most recent data retrieved from the ERP system.

There are a number of other works advocating the use of templates to improve model building efficiency (e.g., Pater and Teunisse 1997). However, the usage of templates or library objects still requires a substantial number of manual operations, especially, in modeling large scale systems having a large number of objects.

This paper describes a data driven simulation model building approach. The approach is designed for developing simulation models to be applied for preliminary analysis of multi-stage, multi-product manufacturing systems. It is aimed to reduce efforts associated with model building. For this purpose, manufacturing units are approximated by their generic representation which captures common functions of the units such as, handling of incoming product flows, flow transformation and handling of outgoing product flows. The manufacturing system is defined using a data model. The data model is assembled and subsequently standardized from several raw data sources. It contains information about both structure and operational characteristics of the system. A simulation model generator uses these data to automatically create a simulation model. The model is created on the basis of a predefined template.

The proposed model building approach is applied for testing scheduling algorithms at stamping plants of an automotive company. Modeling of each stamping plant includes dealing with several units such as external suppliers; blanking, pressing and assembly departments. Each plant processes up to a thousand products and their components using more than a hundred work stations. Data necessary for the model building are extracted from several different data sources such as relational data bases. The automated model building approach is attractive because, (a) the simulation model is to be used as the supplementary tool for testing of scheduling algorithms (i.e., only limited resources for simulation model building are provided), (b) there is a large number of similar objects in the system (e.g., at certain level of abstraction, processing of all products is almost identical) and, (c) all stamping plants are similar though their dimensions vary.

The main contributions of this research are as follows:

- Elaborating tools for data gathering from multiple sources;
- Developing a standardized data representation of multi-stage manufacturing systems;
- Separating the simulation model from its input data;
- Developing tools for automated model generation.

The rest of the paper is organized as follows. The following section describes the conceptual framework. It is followed by more detailed description of the proposed methodology and application examples.

## MODEL BUILDING APPROACH

The proposed simulation model building approach utilizes two main concepts: 1) separation between data and the model; and 2) a generic representation of manufacturing units. The main stages of the model building approach are shown in Figure 1.

Data necessary for the simulation model building are located in several different raw data sources such as, data bases and spreadsheets. A data converter is used to gather these data from all sources and to create a data model. The data model represents the data in a format suitable for generation and execution of a simulation model.

The data sources may have different formats, and definitions of data fields may differ among data sources. Therefore, a modeling taxonomy is used to establish standardized data definitions (Chandra et al. 2002). The taxonomy is a single and comprehensive source of information about characteristics of a general system. An ontology enables communications between different components of the particular system under consideration

on the basis of the standardization provided by the taxonomy. In this case, the ontology defines mapping between the data sources and the data model. It defines both content and structure of data.

A model generator automatically creates a simulation model using the data provided in the data model. It assumes certain characteristics of the manufacturing system. Manufacturing units involved in the system are believed to have common functions. These common functions are handling of incoming and outgoing flows, flow transformation and control. A generic unit performing all these functions is constructed (Figure 2). Each manufacturing unit is approximated by its generic representation. The control function determines the way each generic function is performed at a particular unit. For instance, the handling of outgoing flows can be performed in either a push or pull manner. A network of the generic units represents the entire multi-stage manufacturing system.

The simulation model is generated on the basis of predefined template. The template does not contain any simulation objects. It only contains procedures for executing control of the generic functions and data declarations. The procedures have a uniform design. Different procedures can be developed to perform the same activity. Thus, different management policies can be analyzed.

Intermediate data are used to improve efficiency of data exchange between the data model and the simulation model.

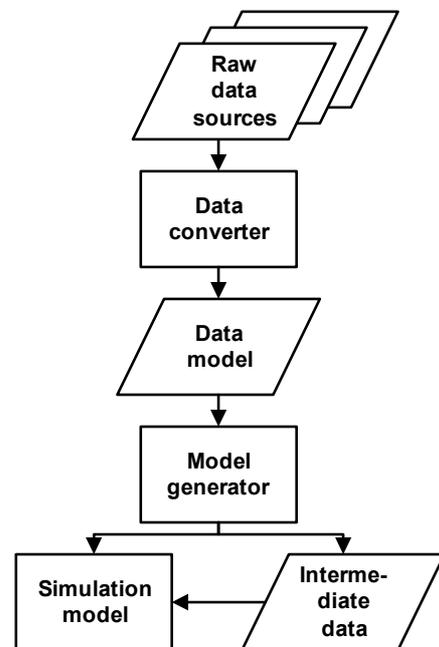


Figure 1: The Model Building Approach

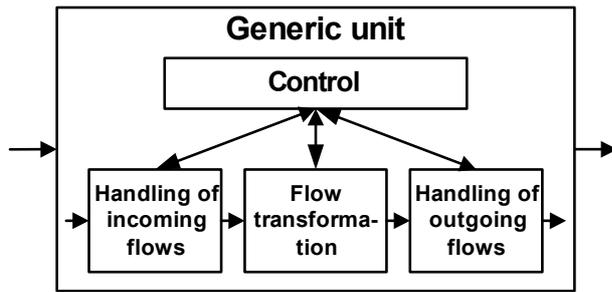


Figure 2: The Generic Unit

## MODEL BUILDING STAGES

### Data model

The data model organizes data describing the system in a manner suitable for the simulation model building and execution of the simulation model. These data describe structure of the system, properties of production units and products produced, and relationships of the system with its external environment including customers. For purposes of execution of the simulation model, structuring of the data should ensure quick access of necessary data items.

The data converter generating the data model is shown in Figure 3. The taxonomy contains a standardized description of concepts relevant to the system. It is not built specifically for the simulation model building, but is a part of more general enterprise wide initiative for data standardization. The taxonomy facilitates common understanding for terms such as products, resources, processing time, etc.

The data conversion process is illustrated by an example. The raw data source contains data fields characterizing throughput for each resource in items per hour and a corresponding efficiency measurement in percents. The converter uses these data fields to determine processing time for each resource in hours per item (this values is used by the simulation model) and places the derived data item in the appropriate position of the data model. Another example is translation of the term Work Center used in one of the raw data sources. The converter identifies this term with the taxonomical term Resource, which is also defined in the same way in the simulation data model. If a product is produced internally, it should have at least one resource assigned. But the generic representation of units requires a resource assigned and products purchased from external suppliers. Therefore, the converter assigns a dummy resource to the external products. These and similar conversion rules are described in the ontology.

The data model consists of multiple tables containing information about structure and operational characteristics of the system. The structure of the system is described by bill of materials, etc. The structural information is also represented using several

specialized tables, which are designed to facilitate data retrieval by the simulation model. The operational characteristics describe processing time, setup time, transportation time. The data model is implemented as a Microsoft Excel workbook. Table 1 lists tables included in the data model. The concept of product and unit pair is introduced to make distinction between the same products processed at different units. Time parameters are specified using a string describing a probability distribution.

The elaborated data model representation allows describing a wide range of manufacturing networks. The main characteristics of these networks are as follows. A product can be produced at several units and it can be a component of several products produced at different units. A resource belongs to one particular unit (as specified in Table ResourceUnit). It has finite capacity. Several products may share the same resource, and a product can be produced by using alternative resources as specified in Table PairResource.

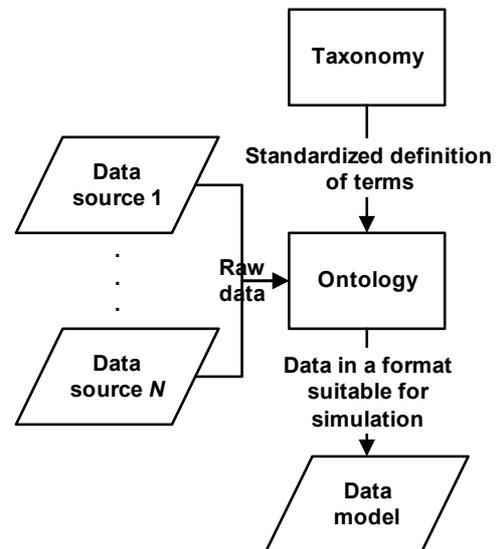


Figure 3: The Data Converter

Table 1: Tables of the Data Model. Type S Refers to Structural Data and Type O Refers to Operational Data.

Table	Type	Description
Definitions	S	Dimensional data (e.g. number of products) and modeling control data (e.g. number of replications)
Demand	O	Customer demand per week
Schedule 1	O	Scheduled order sizes
Schedule 2	O	Scheduled resource assignments
UnitsProducts	S	Shows products produced by each unit

Table	Type	Description
Pairs	S	Defines pairs
PairDestinations	S	Defines possible destinations for a product from the pair
ResourceUnit	S	Shows resources for each unit
PairResource	S	Shows which resources can be used to process a product from the pair
BOM 1	S	Bill of materials, indicates components of each product by component number
BOM 2	S	Bill of materials, indicates items of each component needed
SetupTime	O	Setup time for products according to resource used
ProcessingTime	O	Assembly time for products according to resource used
TransTime	O	Transportation time for products according to destination
ResourceFailure 1	O	Time between two consecutive resource failures
ResourceFailure 2	O	Resource downtime duration

### Simulation model

The simulation model is automatically created by the model generator. The model generator creates one submodel for each generic unit and one submodel to represent external customers. The simulation model is generated in the ARENA simulation modeling environment (Rockwell Software 2001). A generated submodel representing the generic unit is shown in Figure 4. The representation of the manufacturing system consists of several such submodels.

Block 1 at the beginning of each period (week) generates an entity representing a production order for each product produced by a particular unit. Block 2 assigns values of identification attributes to the entities. Block 3 reads data from the production schedule provided in order to determine the batch size and the resource to be used. The block has capabilities to change initial resource assignments according to current circumstances. The scheduled resource assignments can be changed, if the schedule does not contain any assignments. One entity represents the entire batch of products. After leaving Block 3, the entity carries

information about the product it represents, the batch size and the resource assigned. The entities are held in Block 4 until the assigned resource and all components of the product become available. Each product is held in its own queue, and the holding condition is also product specific. These queues and holding conditions are organized using the set of queues, and the set of expressions option, respectively. The production setup process is represented by Block 5. The setup process requests a product specific resource according to the assignment. The processing time depends upon the product and the resource used. Block 6 is used for additional checking for availability of components before the final assembly is started. Block 7 retrieves the components from the inventory. The inventory is represented using a multi-dimensional array. The assembly process is represented by Block 8. It requests a product specific resource according to the assignment. The processing time depends upon the product and the resource used. Block 9 checks whether to change the scheduled resource assignments. Block 10 splits the production batch in transportation batches (products can be sent to different parent units). The production batch is split in as many transportation batches as the number of parent units. Block 11 determines the size of each transportation batch according to its intended destination. The transportation process is represented by Block 12. The transportation time depends upon the product and the destination. Block 13 represents receiving of components from other units. Block 14 updates inventory data for the components. Block 15 checks whether or not to change the scheduled resource assignments.

The generated model is the conventional ARENA model. A user can edit the model, use the standard output reporting features and perform other manipulations.

At the beginning of simulation, modeling data from the data model are loaded in the simulation model. Before loading, the intermediate data have been created by converting the data model tables from the Excel format into the text format because ARENA reads text files much faster than Microsoft Excel files. Some of the data tables are loaded into ARENA arrays for access by ARENA objects, while some others are loaded in Visual Basic arrays for access by control functions.

All VBA blocks invoke a main Visual Basic procedure. An entity attribute characterizing the function to be performed is assigned to the entity before it enters a VBA block. The main procedure reads attributes of the entity to determine a specific procedure to be called and parameters of the specific procedure. The specific procedures read data from the production schedule, reassign resources, update inventory data, check material availability, determine transportation route for products, etc. All procedures are part of the model template.

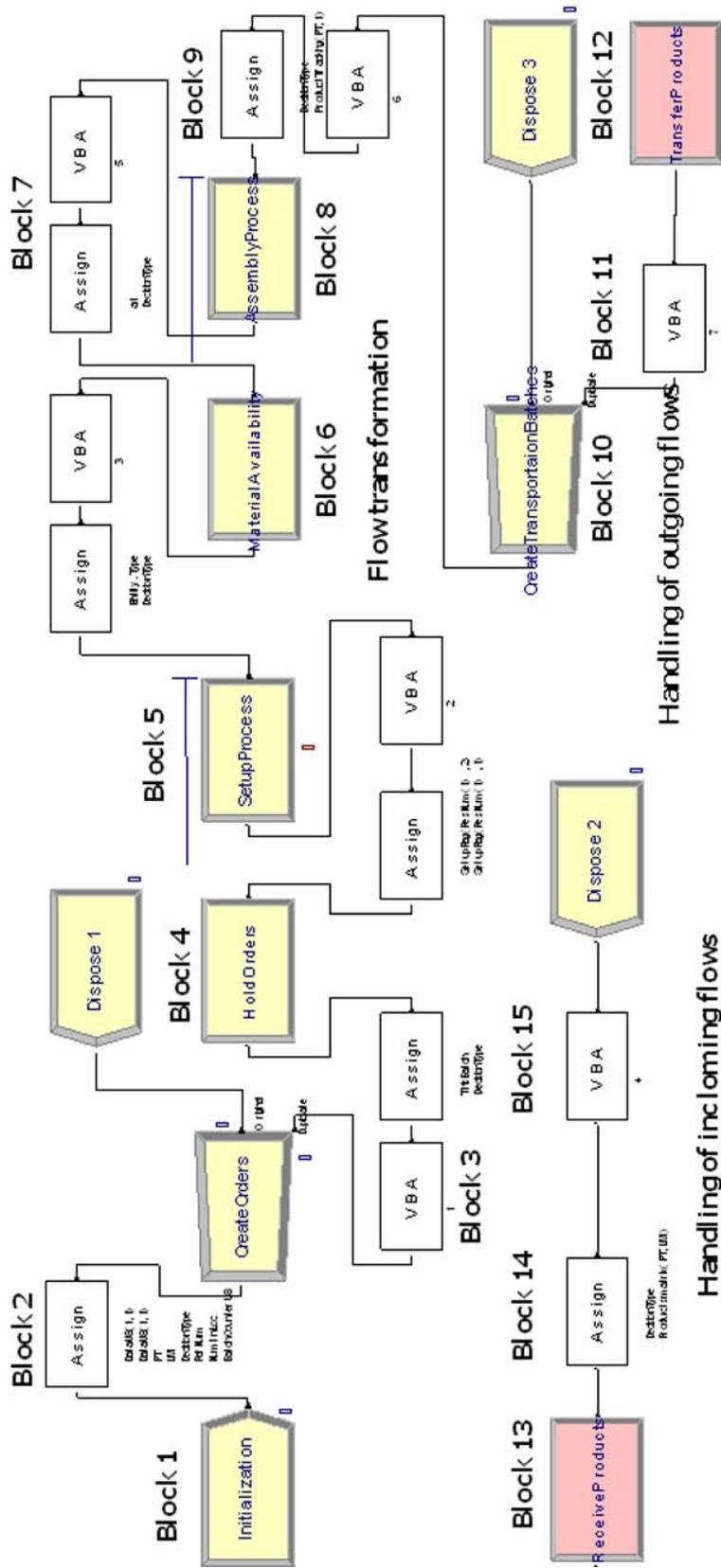


Figure 4: Representation of the Generic Supply Chain (ARENA implementation).

VisuaBasic. It creates ARENA objects using the ActiveX technology. Actually, the same data model can be used to create a simulation model in other simulation modeling environment supporting the ActiveX technology. A similar model generator has been developed for creating models in ProMODEL. However, ARENA appears to be more flexible mainly because of better support for integration with high level programming languages, easier generation of animation, higher level of openness to user editing, and more flexible input and output features.

The simulation model needs to be regenerated, if the structural data tables have been changed. Changes in the operational data tables can be captured just by updating the intermediate data.

Development of the model generator can also be a labor-intensive task. However, this model generator has been derived from a more general supply chain simulation model generator created by the authors. Modifications are introduced to represent some specific properties of the particular manufacturing system. Additionally, the usage of the model generator effectively eliminates syntactic and logical errors, which are likely to occur because of a large number of objects.

## APPLICATION

The elaborated model building approach and automatically generated simulation models are used in experimental studies of a multi-stage manufacturing system in order to evaluate applicability of the model building approach and to improve performance of the system analyzed.

### Case Description

Case studies using the elaborated model building approach are conducted based on a modeling problem experienced by an automotive company. The manufacturing system considered consists of raw material suppliers and a stamping plant (a similar steel processing supply chain at General Motors has been analyzed by Bhaskaran (1999)). The system is expected to meet strict delivery time requirements. Meeting of these requirements often force the plant to use a premium cost transportation mode. The modeling objective is to determine whether additional transportation costs are caused by:

- insufficiently coordinated deliveries of raw materials;
- production scheduling inefficiencies;
- faults in operations.

The manufacturing system produces approximately 300 end products using approximately 1000 components. Manufacturing operations are performed using approximately 100 work centers. The end-

products are delivered to about 30 customers, which are assembly plants and repair centers. Each customer places orders for multiple products.

The stamping plant consists of blanking, pressing and assembly departments. The blanking department cuts the raw steel into rectangular pieces. Work centers at this department are relatively flexible to process different products and setup times are insignificant. The pressing department stamps the blanks into parts. Work centers at the pressing department are partially specialized. There are substantial setup times. Welding and other operations are performed on stamped parts at the metal assembly department. Work centers at the assembly department are specialized where setup times are smaller than at the pressing department. Transportation times within the plant are assumed to be insignificant.

Production is initiated according to a production schedule. The production schedule is elaborated according to weekly customer demand. It specifies the quantity of products to be produced and resources (i.e., work centers) to be used in production of these products. The production schedule is implemented in the rolling horizon environment. The resource assignments can be dynamically changed to adjust for the actual state of the system.

Majority of costs in the system are fixed. Variable costs are the inventory holding cost and the transportation cost. The transportation cost consists of the cost for a standard mode of transportation and the cost for a premium mode of transportation. The standard mode of transportation is used for on time deliveries. The premium mode of transportation is used, if deliveries of ordered products are delayed.

The stochastic factors in the system are setup times, processing times and resource failures. Additionally, external demand used to elaborate the production schedule is stochastic. However, the demand for the current production period is fairly stable.

The company operates a relatively large number of similar manufacturing systems. In this paper, modeling is conducted only for one of them. However, the same data model populated with appropriate data and the simulation model generator can be used to automatically generate simulation models for other related manufacturing systems.

### Experimental Design

The third modeling objective on faults in operations is addressed. Particularly, the impact of setup time uncertainty on the production performance is analyzed. The setup time uncertainty is caused by a number of factors many of which are supposed to be avoidable. Three levels of the setup time uncertainty are considered. These levels include deterministic setup

time, the standard deviation of the setup time equal to 10% of the average setup time and the standard deviation of the setup time equal to 20% of the average setup time. In the cases with the stochastic setup time, it is modeled using the lognormal distribution.

The systems performance is measured by waiting time of customer orders. If the waiting time is zero then the customer orders are satisfied without relying on the premium transportation. However, if the waiting time is larger than zero, then the premium transportation is to be used. In the real system, the waiting time is not allowed to exceed certain threshold even when the premium transportation cannot assure timely deliveries. This aspect is currently ignored. The waiting time is a proxy measure for the premium transportation cost.

The system is modeled for one year and five replications are conducted for each level of the setup time variability.

The model does not represent a number of constraints and operations of the real system. For instance, the model does not represent the restriction that only a limited number of setups can be performed simultaneously. Impact of these constraints and operations are assumed to be insignificant. Additionally, many decisions such as prioritizing deliveries among customers are done in a non-formal manner, thus, making these decisions difficult to model and validate.

### Experimental Results

Figure 5 shows the average customer waiting time according to the setup time variability. The waiting time is expressed relative to the waiting time in the case of the deterministic setup time. The results indicate that the customer waiting time substantially depends upon the setup time variability. The relationship is approximately linear. However, the waiting time is larger than zero (i.e., the premium transportation mode is to be used) also for the case with the deterministic setup time.

Therefore, improving the setup time variability is not the only solution to the transportation cost reduction. There are multiple reasons for the setup time variability. The variability is caused by workforce resource limitations at the work floor level, qualification of workforce, quality of raw materials and precision of manufacturing tools. Dealing with these issues may require substantial organizational changes.

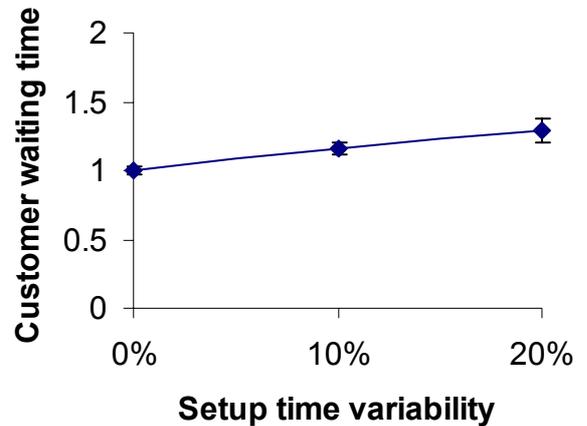


Figure 5: The Average Relative Customer Waiting Time According to the Setup Time Variability

Highly variable resource utilization is another problem faced by the manufacturer (Figure 6). Some of the resources are nearly overloaded, while others have low utilization rates. Achieving a more uniform distribution of the workload among resources would also facilitate reduction of the customer waiting time. However, this is constrained by inflexibility of resources, which are capable of processing only a limited number of products.

### CONCLUSION

The automated simulation model building approach

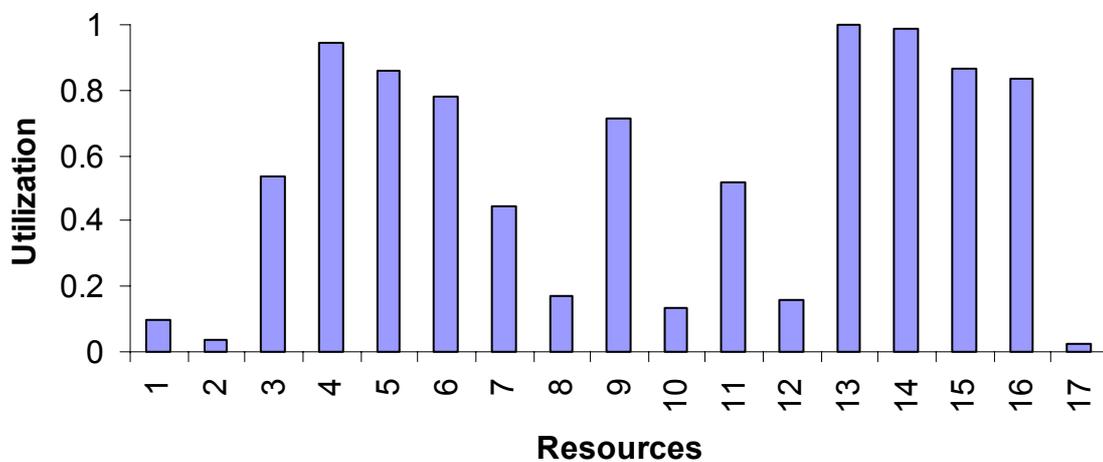


Figure 6: Average Utilization of Resources at One of the Manufacturing Units

has been elaborated. The approach is aimed at (a) reducing simulation model building efforts, (b) reducing model building errors, and (c) providing reusability. The simulation model is generated according to the problem definition provided by the data model which describes a multi-stage manufacturing system in the standardized manner. It is developed by assembling and standardizing raw data characterizing the system from multiple data sources. The data are standardized using the taxonomy and ontology concepts. Manufacturing units in the simulation model are represented using their generic approximations which describes the common functions of manufacturing units. The generated simulation model is open for customization.

Advantages of the elaborated model building technology are:

- Model building efforts are reduced by using the generic approximation of manufacturing units;
- The number of modeling errors is reduced because the large scale structure of the system is generated automatically instead of manual input;
- Editing of the simulation model is made more efficient because changes can be introduced by simple updating of the data model;
- The modeling process can be repeated for multiple similar manufacturing facilities without substantial model building efforts;
- Systematization of the problem analysis by establishing standardized definitions for subjects involved in the system.

The proposed model building approach and the generated simulation model have been applied to study the multi-stage manufacturing supply chain. Objectives of the studies are to identify factors reducing efficiency of manufacturing operations and to test alternative production schedules. The experimental results suggest that the setup time variability has substantial adverse impact on efficiency of the manufacturing system considered.

There are substantial problems associated with validation of the model. Two of the main obstacles are that the actual system heavily relies on judgmental decisions made by production managers and a lack of reliable data quantifying operational properties of the actual system. Accumulation of data needed for a thorough validation would incur substantial additional expenses. Currently, the discussion of the results with area specialists is the main validation approach. Additionally, validation should be performed with respect to the level of approximation provided by the model (an account on validation of models at different levels of abstractions can be found in Persson (2002)).

## REFERENCES

- Baker, G.S. 1997. "Taking The Work Out Of Simulation Modeling: An Application Of Technology Integration." In *Proceedings of the 1997 Winter Simulation Conference* (Atlanta, GA, Dec. 7-10), 1345-1351.
- Bhaskaran, S. 1998. "Simulation analysis of a manufacturing supply chain." *Decision Sciences* 29, 3, 633.
- Chandra, C.; J. Grabis; R. Marukyan; and A. Tumanyan. 2002. Supply chain taxonomy: Development and applications, *Research and Application Congress, SAP Institute for Innovation and Development* (Tampa, FL, Feb. 23-25), 2002.
- Ignall, R.G.; P. Kolesar; and W. Walker. 1978. "Using simulation to develop and validate analytical models." *Operations Research* 28, 237-253.
- Law, A.M.; and W.D. Kelton. 2000. *Simulation Modeling and Analysis*. McGraw-Hill, New York.
- Pater, A.J.G.; and M.J.G. Teunisse. 1997. "Use of a template-based methodology in the simulation of a new cargo track from Rotterdam harbor to Germany". In *Proceedings of the 1997 Winter Simulation Conference* (Atlanta, GA, Dec. 7-10). 1176-1180.
- Persson, J. F. 2002. "The impact of different levels of detail in manufacturing systems simulation models." *Robotics and Computer-Integrated Manufacturing* 18, 3-4, 319-325.
- Petrovic, D. 2001. "Simulation of supply chain behaviour and performance in an uncertain environment." *International Journal of Production Economics* 71, 1-3, 429-438.
- Rockwell Software. 2001. ARENA: User's Guide. Sewickley: Rockwell Software Inc.
- Son, Y.J.; A.T. Jones; and R.A. Wysk. 2000. "Automatic Generation of Simulation Models From Neutral Libraries: An Example." In *Proceedings of the 2000 Winter Simulation Conference*, 1158-1567.
- Swaminathan, J. M.; S. F. Smith; and N. M. Sadeh. 1998. "Modeling supply chain dynamics: A multiagent approach." *Decision Sciences* 29, 3, 607.
- Werner, S; and G. Weigert. 2002. "Process Accompanying Simulation – A General Approach For The Continuous Optimization Of Manufacturing Schedules In Electronics Production." In *Proceedings of the 2002 Winter Simulation Conference*, 1903-1908.

## AUTHORS BIOGRAPHY

**CHARU CHANDRA** is an Associate Professor in Industrial and Manufacturing Systems Engineering at the University of Michigan-Dearborn, U.S.A. Prior to this Charu was a Post Doctoral Fellow at Los Alamos National Laboratory, Los Alamos, New Mexico, and at the University of Minnesota, Minneapolis, U.S.A. He has worked in the industry as Information Technology Manager and Systems Analyst. He is involved in research in Supply Chain Management, and Enterprise Integration issues in large complex systems.

Specifically, his research focuses on studying complex systems with the aim of developing cooperative models to represent coordination and integration in an enterprise. He has published several papers and book chapters in leading research publications, in areas of supply chain management, enterprise modeling, inventory management, and group technology. He teaches courses in Information Technology, Operations Research and Supply Chain Management. His Ph. D. degree is in Industrial Engineering and Operations Research from the Arizona State University. He is a member of Institute of Industrial Engineers, Institute of Operations Research and Management Sciences, Decision Science Institute, Production and Operations Management Society, American Association of Artificial Intelligence, and Association of Information Systems.

**JĀNIS GRABIS** obtained his Ph. D. degree in Information Technology from the Riga Technical University in 2001. He spent two years with the University of Michigan-Dearborn and currently works in the Riga Technical University. His main research interests are supply chain management, simulation, forecasting and software project management.

# CONTROL ROOM INTERFACE UPGRADE OF AN OPERATING FULL-SCOPE TRAINING SIMULATOR

Endre VÉGH<sup>1</sup>, János BIRI<sup>2</sup>, Laura BÜRGER<sup>1</sup>, Gábor HÁZI<sup>1</sup>, László VARGA<sup>3</sup>

<sup>1</sup>KFKI Atomic Energy Research Institute  
1525 Budapest 114 P.O.Box 49, HUNGARY

E-mails: vegh@sunserv.kfki.hu, buerger@sunserv.kfki.hu, gah@sunserv.kfki.hu

<sup>2</sup>MTA-ITA\_LAI Foundation for Information Technology  
1525 Budapest 114 P.O.Box 49, HUNGARY

E-mail: jbiri@sunserv.kfki.hu

<sup>3</sup>EASTRON Trading & Development Agency  
1148 Budapest, Nagy Lajos király útja 20

E-mail: varga@eastron.hu

## KEYWORDS

Control Room Interface, Simulator upgrade, VMEbus.

## ABSTRACT

The Full-scope Block Simulator of the Paks Nuclear Power Plant has been operating since 1988. In 2001 the whole Control Room Interface of the Simulator was replaced by a new one. The upgrade was executed in a very tight time schedule, because the training on the simulator had to be maintained during the installation of the interface. This paper describes the operation of the new interface and the main organization aspects of the installation.

## INTRODUCTION

The full-scope training simulator of Paks Nuclear Power Plant (Hungary) has been operating since 1988. Six years ago the original VAX computer of the simulator was replaced by a DEC AlphaServer 2000 - 4/275, and – with this new hardware - the simulation cycle time was reduced from 1 second to 0.2 second. However, this simulation speed-up could not be seen in the Control Room, because its interface could operate only with 1 second refreshing time. The Control Room Interface (CRI) was designed in the mid-eighties and was built from CAMAC modules. Thus the interface became obsolete because

- its speed was not sufficient for the new requirements,
- spare parts for the electronics were hardly available,
- all of its reserve channels were practically spent.

For the above reasons a completely new interface was designed and installed in 2001. For the installations the following constraints had to be met:

- the cabling between the interface and the Control Room had to remain intact,
- the installation had to be performed in very short time steps, because the training on the simulator had to be maintained during the interface upgrade.

In the new interface VME electronics has been used, because of its reliability and we have had a proper development background.

## GENERAL CRI CONSIDERATIONS

In a Control Room there are several different instruments and their implementation in a simulator can be different: from the use of the very same devices, to the use of their graphical representations. In our simulator the faces of the instruments are the same as the real ones, but their operation is simulated. In this case again several approaches are possible. In our approach the instruments are divided into devices. A device is a simple entity in the Control Room having identical connections to the CRI, e.g.: digital input devices are pushbuttons, selector switches, relay contacts, etc. Thus every device has one or more input/output lines in the CRI and a value associated with the actual states of the lines (e.g.: a logical value in the case of a pushbutton, or an integer in the case of a thumbwheel switch). Most devices have very simple logic, but there are also some exceptions, e.g.: the synchronoscope, used to connect the electrical generator to the grid. In our approach the devices are handled in the CRI electronics, i.e. the input system calculates the values of the input devices, the output system drives the output lines associated to the values of the output devices.

The instruments are handled in the simulation server. Most instruments are modeled in the Control Room Communication System (CRCS), but the most complicated controllers (as e.g.: turbine controller, reactor power controller) are realized by independent model programs.

## HARDWARE CONFIGURATION

The Control Room Interface can be divided into five units, as digital input (DI), two identical digital outputs (DO1, DO2), analogue output (AO) and special output (SP). The SP subsystem drives the special devices, e.g. synchroscope. The units are connected to the simulation server by a dedicated Ethernet network (See Fig.1.).

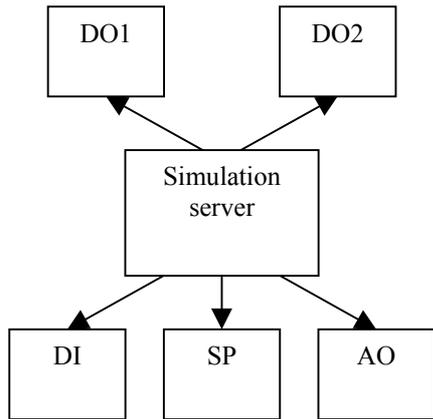


Figure 1: Basic hardware structure

Each unit has an identical structure: it contains a processor and identical peripheral cards arranged in crates. Because the peripheral cards need more than one crate, the crates are connected to each other by a transparent bus. The processor is a BVME4500 single board computer produced by BVM Ltd, Southampton, England. This board contains a 32-bit Motorola processor running at 33 MHz, a 10baseT Ethernet interface, a VMEbus system controller, 16 Mbytes RAM and 2 Mbytes Flash memory. The peripheral cards were developed and manufactured in the MTA-ITA-LAI. There are five different peripheral cards:

1. DI-64 receives 64 active low level TTL signals,
2. DO-64 provides 64 open collector Darlington drivers,
3. DR-64 contains 64 reed relays to convert logical signals to ground independent relay contacts,
4. DA-32 generates 32 current type (0÷5 mA, or 4÷20 mA, jumper selectable) analogue outputs with 8-bit resolution,
5. STP4 drives four stepping motors.

There is an additional analogue output board in the SP system (MVME 512-051) generating 12-bit resolution setpoint signals for controllers. The total number of input/output lines can be seen in the following table:

Unit	Line
Digital input	2880
Digital output	6912
Analogue output	768
Stepping motor drive	8

At present all lines are not used, but 10÷15% are reserve lines.

## SOFTWARE STRUCTURE

The basic software structure of each interface unit is identical:

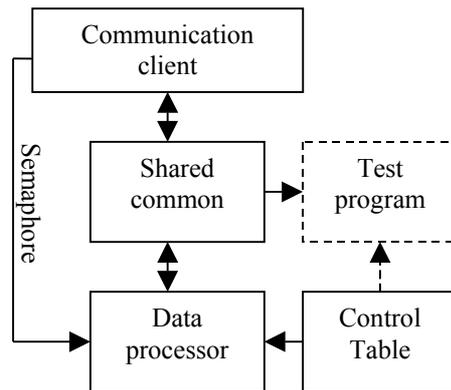


Figure 2: Basic software structure

An interface unit always replies to a command received from the server. The data processing is table controlled, i.e. a control table determines which lines form a device and how the lines have to be treated. The control tables are downloaded from the simulator server at the beginning of a simulation session. This solution is very versatile, since a new device can be added to the system without modifying the interface program.

The execution of the test program is optional, it is executed from a remote terminal connected to the network.

In the DI system there is an additional program running with its own 40 milliseconds timing. This program measures the input lines and detects the lines changed since the last data request.

All of the interface programs are developed in C language and they are executed in a diskless OS9 operating system.

## INTERFACE STATES

Every interface unit has the following system states:

- initial,
- downloading,
- downloading completed,

- setup,
- setup completed,
- normal.

When an interface is switched on, the unit goes to **initial** state. Downloading can be executed only from this state.

In course of **downloading** the control table is loaded from the simulator server. When the downloading is successful, the unit goes into the **downloading completed** state. Setup may be executed only from this state.

**Setup** exists when an initial condition is loaded into the simulator server. During setup the output image of the Control Room is sent to the interface, moreover, the state of the input lines are measured and compared to the input image stored in the initial condition file. So called lineup reports are generated until the measured and the stored input images are different. When the setup successfully terminates, the interface goes to the **setup completed** state. The unit may be controlled into **normal** state only from this state. The permitted state transitions can be seen in Fig. 3.

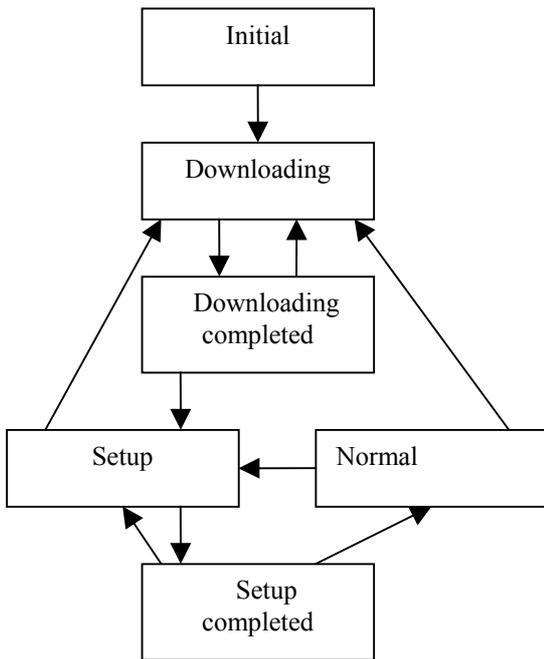


Figure 3: Permitted system state transitions

All of the state transitions are controlled by commands generated by the Control Room Communication System of the simulator server.

## CRCS SOFTWARE STRUCTURE

The Control Room Communication System (CRCS) is a subsystem in the simulator server. It contains several programs realizing the models of the Control Room and driving the different event-driven actuator handlers of the simulator (pump handler, valve handler, etc.). This system can be divided into two parts as logical- and physical ones. The Control Room Data Processor (CRDP) forms the logical level and it is connected to the actuator handlers. The physical level contains different Control Room Interprocess (CRIP) drivers. Each interface unit has its own CRIP driver. Each CRIP driver is connected to the VMESRV server program realizing TCP/IP data transfer among the clients and the server.

The basic CRCS program structure can be seen in Fig. 4.

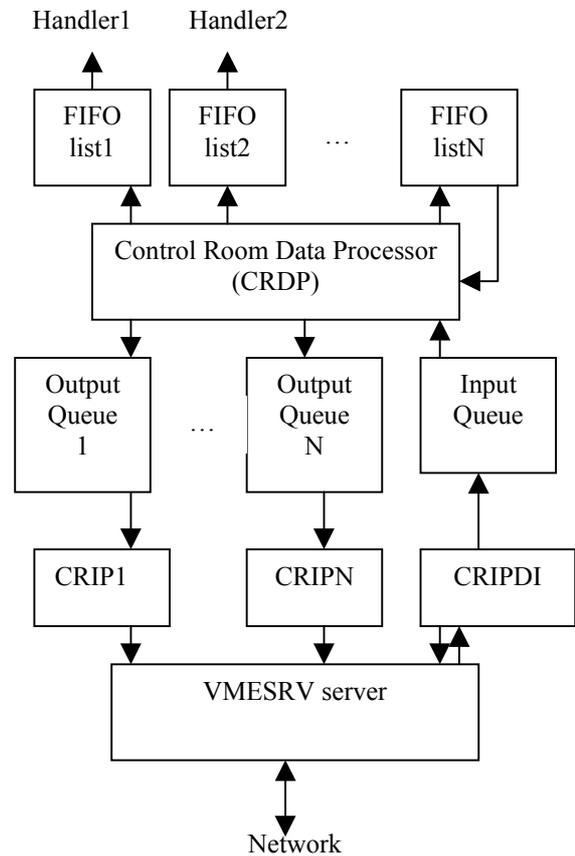


Figure 4

CRDP drives the whole interface system. This program is started every 0.2 second by its own timing.

CRDP has FIFO organized circular input/output buffers, namely 32 input- and 2 output buffers. From the 32 input buffers 24 are connected to different actuator handlers and 8 are used within the CRDP

itself. This CRDP feedback is used for the modeling of complex instruments.

CRDP drives the CRIP processes via output queues. Each CRIP has its own queue for storing both data and commands. The DI and the SP units have additional queues for transmitting to the CRDP level the input changes received from the CRI electronics. Each CRIP process has its own flag and they are awakened by CRDP every 0.2 sec.

## OPERATION MONITORING

Since the simulator is continuously used and consequently the Control Room Interface is permanently switched on, monitoring of its correct operation has utmost importance. Three different kinds of monitoring are needed, as

- monitoring the operation of the CRI units,
- monitoring the operation of the CRCS system,
- monitoring the data transfer in the network.

For monitoring the operation of the CRI electronics, each interface unit has a test card and a test program. The test card listens to the VMEbus and it generates an error signal if a peripheral board doesn't answer to the VMEbus system controller. In this way the erroneous cards are detected.

When an error is detected a test program can be started from a remote terminal. This menu driven program displays the following types of information on the terminal:

- the content of the last input/output data block,
- the content of the control table, which determines the operation of the given interface unit,
- the number of the communication errors registered since switching on,
- the actual values of the different control variables of the given unit.

Moreover, by means of the test program the value of any device can be overwritten.

Each unit checks the execution of the received commands and if an intolerable error occurs, a fatal error is reported to the CRCS system. In the CRCS a fatal error reception causes a break in the operation of the simulator and an error message is sent to the instructor.

In the CRCS there is also a monitor program. This monitor program can be started from any terminal of the simulator server. The monitor displays the following data:

- the content of any circular buffer of the CRDP program,
- the content of the input/output queues,
- states and error diagnostics of the CRIP processes.

A third program monitors the data transfer of the network. This program is executed in the simulator server and it has the following services:

- listing of the last data block of any interface unit,
- preparing input data blocks for testing purposes,
- displaying error counters of the different clients.

## UPGRADE ORGANIZATION

The CRI upgrade was complicated enough not to start the project without any experience. For this reason first a pilot project had been started, in which only the SP part of the interface was built and installed in the simulator. This subsystem was selected for pilot installation, because few cabling was affected, but at the same time the most complicated devices had to be programmed. During this project the communication server-client program pair and the main programming structure of the interface were worked out. The real upgrade project has started only after the successful completion of the pilot project.

The upgrade was executed in a very tight time schedule, because the simulator is practically permanently used. The whole CRI electronics was manufactured in our institute and first it was installed at us. In the Simulator Department we have a simulation server identical to the one operating in the power plant, and the new interface was connected to it. One piece from each Control Room devices was connected to the interface and the operation of the new interface was tested in detail. However, only the device treatment was tested in this way, some controllers could not be tested without the Control Room. When the development and testing was completed, an in-house testing was organized with the representatives of the Buyer. The interface was delivered to the final site after the successful in-house testing.

At the final site the new interface was installed beside the operating old CAMAC interface. The testing was organized during prolonged weekends in steps. First the analogue cables were connected to the AO unit and the digital input/output cables remained on the old interface. In the CRCS system the CRIP driver of the analogue subsystem was replaced by the new one. All analogue signals were individually tested before the next installation phase has started.

In the second phase the digital input part was replaced, because it needed less cabling work than the digital output subsystem. Again every input device was individually tested.

In the third phase the digital output cables were also connected to the new interface and every output line was tested seriously.

The last testing phase has started when the whole Control Room was driven by the new interface. During this one-week long testing experienced instructors performed different transients in the simulator. During this phase some fine tunings were done, because in the mixed operation – when both old and new interfaces operate – the CRDP timings were determined by the slower old one.

After completion of the installation the old interface remained at its original place for about half year, in order to use it again if a serious failure occurs. Fortunately, it was not the case and finally the old interface was removed from the computer room and the cabling was finalized. Since this time the new VME interface has been operating continuously.

#### **AUTHOR BIOGRAPHIES**

**JÁNOS BIRI** received his M.Sc.E.E. degree at the Technical University of Budapest in 1960. He is member of ESONE, member of IEEE, member of the New York Academy of Sciences. Since 1970 he is the leader of the department for laboratory automation in KFKI-MSZKI. For his leading role in the development of the CAMAC modular real-time system, he was awarded with the National Prize of Hungary in 1980. His research fields include modular instrumentation for experimental physics, special analogue devices (e.g. analog - to - digital converters), real-time systems for laboratory measurement and automation.

**LAURA BÜRGER** received her M.Sc.E.E. degree at the Technical University of Budapest in 1958. She is member of the European Nuclear Society. Her research field is the application of computer technique in the nuclear industry. Previously she was interested in computerized methods of disturbance analysis. She took part in the Jánossy price awarded team realizing a computer based closed loop reactor control system on the Budapest Research Reactor. Presently her main interests are: real time data collecting and processing systems, simulators, etc.

**GÁBOR HÁZI** received his B.Sc.E.E. degree at the Kando Kalman College in 1992, M.Sc.E.E. degree in 1995 and Ph.D. degree in 2000 at the Technical University of Budapest. He is member of the Hungarian Nuclear Society. For his research activity, he was awarded with the Prize for Young Researchers of the Hungarian Academy of Sciences in 2001. His research fields include noise diagnostics in nuclear power plants and different aspects of thermohydraulic modeling.

**LÁSZLÓ VARGA** received his M.Sc.E.E degree at the Technical University of Budapest in 1980. In the early 80'ies he was the developer of the CPU module and programmer of a PLC called EV-01. Later he became a research fellow-worker at MTA-SZTAKI and participated in the software development for the Central Control Room of the Nuclear Power Plant, Paks. In 1998 he together with some of his colleges departed the institute and first founded a private company, called AKRIBIA which was mainly devoted to the development of a special data acquisition and archiving software also for NPP Paks. He is now responsible for software quality assurance at a successor company, called EASTRON. His main areas of interest are process control and data acquisition, microprocessors and microcontrollers, real-time operating systems, distributed systems and communication.

**ENDRE VÉGH** received his M.Sc.E.E. degree at the Technical University of Budapest in 1961. At that time he joined the Central Research Institute for Physics (KFKI), where he was engaged in the construction of nuclear instruments and in the development of different computer systems for the nuclear industry (reactor information systems, power plant simulators). In the period of 1980-1995 he was the head of the Simulation and Reactor Control Department. He is a member of the Hungarian Nuclear Society. His research fields include reactor core surveillance, plant computers, simulation of nuclear power plants.

# CONTINUOUS COMPUTER SIMULATION MODEL OF THE MARINE GAS TURBINE

Ante Munitic

Mario Orsulic

Josko Dvornik

Maritime Faculty of Split

University of Split

Zrinsko-Frankopanska 38, 21000 Split, Croatia

e-mail: [munitic@pfst.hr](mailto:munitic@pfst.hr), [josko@pfst.hr](mailto:josko@pfst.hr)

## KEYWORDS

System dynamics, modelling, twin shaft gas turbine, continuous and discrete simulation and heuristics optimization.

## ABSTRACT

Simulation Modelling, together with System Dynamics and intensive use of modern digital computer, which mean massive application, today very inexpensive and in the same time very powerful personal computer (PC-a), is one of the most suitable and effective scientific way for investigation of the dynamics behaviour of non-linear and complex: natural, technical and organization systems.

The methodology of System Dynamics (Prof dr. J. Forrester – MIT), e.g. relatively new scientific discipline, in former educational and designer practice showed its efficiency in practice as very suitable means for solving the problems of management, of behaviour, of sensibility, of flexibility and sensibility of behaviour dynamics of different systems and processes.

System-dynamics computer simulation methodology have been used from 1991 to 2003 for modelling of dynamics behaviour of the large number of non-linear ship electrical, thermo-dynamical, hydraulically, mechanical and pneumatically systems. This methodology is used by students as a material for graduate these at Maritime faculty Split. Investigation of behaviour dynamics of the ship propulsion system, as one of the complex, dynamics, non-linear and technical system, requires application of the most effective modelling methods.

The aim of this paper is to show the efficiency of the application of the System Dynamics Simulation Modelling in investigation of behaviour dynamics, one of the complex marine system and process i.e. “gas – turbine”. Twin shaft gas turbine shall be presented with mental-verbal, structural and mathematical-computing modules, and will simulate working process of turbine.

## SYSTEM DYNAMICS SIMULATING MODELLING OF TWIN SHAFT GAS TURBINE

Basic equations of twin shaft gas turbine:

Most often dynamical analyze of gas turbine is based on observation of plan as accumulator of kinetic energy, while dynamics of thermal energy can be conditional ignored.

We presume that process of fuel combustion is momentarily, condition of pressure in turbine is constantly, engine air rate is equal as gas consumption, parameters of atmospheric pressure are relatively unchanged, and ideal heat transfers. Linear behaviour of plan is available when fuel consumption not depend on angular velocity of turbo compressor rotor.

Equations of shaft of turbo compressor with low pressure – consumer:

$$T_{a1}\varphi_{\omega 1} + k_1\varphi_{\omega 1} = \mu_G + k_{\omega 2}\varphi_{\omega 2} \quad (1)$$

$$\frac{d\varphi_{\omega 1}}{dt} = \frac{k_1}{T_{a1}} \left[ \frac{\mu_G}{k_1} + \frac{k_{\omega 2}\varphi_{\omega 2}}{k_1} - \frac{f(t)}{k_1} - \varphi_{\omega 1} \right] \quad (2)$$

$$\frac{d\varphi_{\omega 1}}{dt} = \frac{1}{T_{a1}} (\mu_G + k_{\omega 2}\varphi_{\omega 2} - f(t) - k_1\varphi_{\omega 1}) \quad (3)$$

Mental- verbal model:

When relative variation of fuel consumption  $\mu_G$  and product  $k_2*\varphi_{\omega 2}$  are increasing, relative angular speed variation is increasing also, resulting in positive cause-consequence relation UPV (+).

When relative variation of possible external act of cargo and product  $k_1*\varphi_{\omega 1}$  are increasing, relative angular speed variation is decreasing and observed UPV(-) is negative.

Further, when time of shaft running  $T_{a1}$  is increasing, relative angular speed variation is decreasing, resulting in negative sign UPV (-).

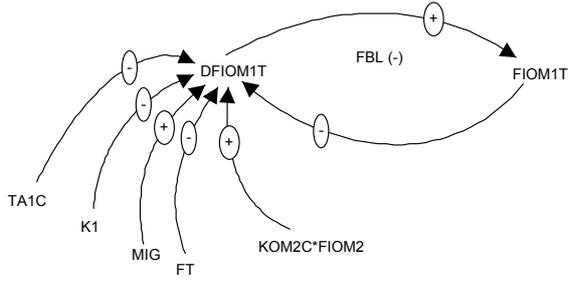


Figure 1. Structural diagram of turbo compressor with low pressure

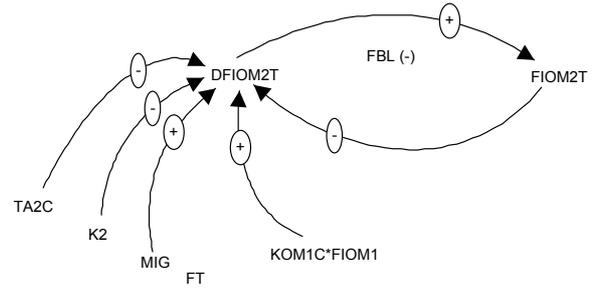


Figure 3. Structural diagram of turbo compressor with high pressure

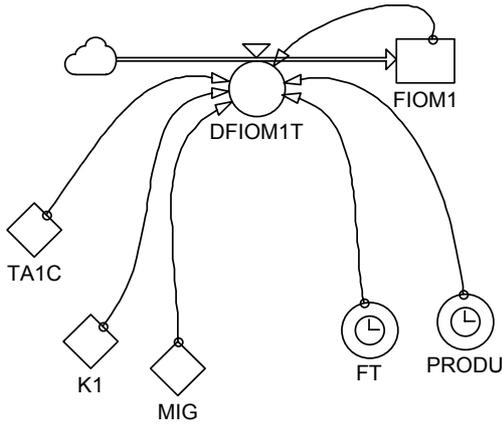


Figure 2. Flow diagram of turbo compressor with low pressure

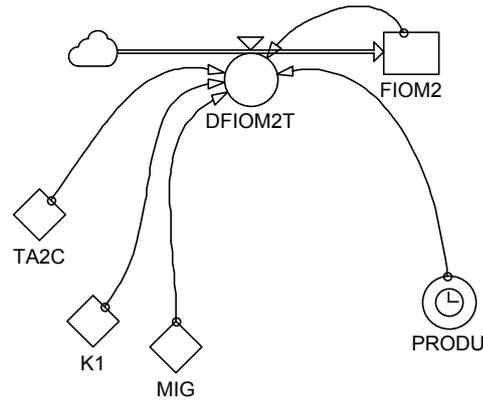


Figure 4. Flow diagram of turbo compressor with high pressure

In observed system there is one feed back loop (FBL):  
 FBL1(-): DFIO1M1T=>(+) FIO1M1T=>(+) FIO1M1T=>(-)DFIO1M1T; with self regulating dynamic character (-), because the addition of negative sign is odd number.

Equations of shaft of turbo compressor with high pressure:

$$T_{a2}\varphi_{\omega 2} + k_2\varphi_{\omega 2} = \mu_G + k_{\omega 1}\varphi_{\omega 1} \quad (4)$$

$$\frac{d\varphi_{\omega 2}}{dt} = \frac{k_2}{T_{a2}} \left[ \frac{\mu_G}{k_2} + \frac{k_{\omega 1}\varphi_{\omega 1}}{k_2} - \varphi_{\omega 2} \right] \quad (5)$$

$$\frac{d\varphi_{\omega 2}}{dt} = \frac{1}{T_{a2}} (\mu_G + k_{\omega 1}\varphi_{\omega 1} - \varphi_{\omega 2}) \quad (6)$$

Mental- verbal model:

When relative variation of fuel consumption  $\mu_G$  and product  $k_2*\varphi_{\omega 2}$  are increasing, relative angular speed variation is increasing also, resulting in positive cause-consequence relation UPV (+).

Further, when time of shaft running  $T_{a1}$  is increasing, relative angular speed variation is decreasing, resulting in negative sign UPV (-).

In observed system there is one feed back loop (FBL):  
 FBL1(-): DFIO1M2T=>(+) FIO1M2T=>(+) FIO1M2T=>(-)DFIO1M2T; with self regulating dynamic character (-), because the addition of negative sign is odd number.

The equations are:

Time of shaft 1. and 2. running:

$$T_{a1} = \frac{I_1\omega_{1n}}{\left(\frac{\partial M_{T1}}{\partial G_T}\right)_0 G_{Tn}}; \quad T_{a2} = \frac{I_2\omega_{2n}}{\left(\frac{\partial M_{T2}}{\partial G_T}\right)_0 G_{Tn}} \quad (7)$$

Self regulating coefficient of shaft 1. is:

$$k_1 = \frac{\left[ \left(\frac{\partial M_{K1}}{\partial \omega_1}\right)_0 - \left(\frac{\partial M_{T1}}{\partial \omega_1}\right)_0 \right] \omega_{1n}}{\left(\frac{\partial M_{T1}}{\partial G_T}\right)_0 G_{Tn}} \quad (8)$$

Self regulating coefficient of shaft 2. is:

$$k_2 = \frac{\left[ \left( \frac{\partial M_{K2}}{\partial \omega_2} \right)_0 - \left( \frac{\partial M_{T2}}{\partial \omega_2} \right)_0 \right] \omega_{2n}}{\left( \frac{\partial M_{T2}}{\partial G_T} \right)_0 G_{Tn}} \quad (9)$$

Coefficient of increasing of angular speed  $\omega_2$  upon angular speed  $\omega_1$

$$k_{\omega 1} = \frac{\left[ \left( \frac{\partial M_{T2}}{\partial \omega_1} \right)_0 - \left( \frac{\partial M_{K2}}{\partial \omega_1} \right)_0 \right] \omega_{1n}}{\left( \frac{\partial M_{T2}}{\partial G_T} \right)_0 G_{Tn}} \quad (10)$$

Coefficient of increasing of angular speed  $\omega_1$  upon angular speed  $\omega_2$

$$k_{\omega 2} = \frac{\left[ \left( \frac{\partial M_{T1}}{\partial \omega_2} \right)_0 - \left( \frac{\partial M_{K1}}{\partial \omega_2} \right)_0 \right] \omega_{2n}}{\left( \frac{\partial M_{T1}}{\partial G_T} \right)_0 G_{Tn}} \quad (11)$$

Relative variation of possible external act of cargo

$$f(t) = \frac{\Delta M_G [f(t)]}{\left( \frac{\partial M_{T1}}{\partial G_T} \right)_0 G_{Tn}} \quad (12)$$

Relative variation of angular speed  $\omega_1$  and  $\omega_2$

$$\varphi_{\omega 1} = \frac{\Delta \omega_1}{\omega_{1n}}; \quad \varphi_{\omega 2} = \frac{\Delta \omega_2}{\omega_{2n}} \quad (13)$$

Relative variation of fuel consumption

$$\mu_G = \frac{\Delta G_T}{G_{Tn}} \quad (14)$$

where are:

- $\omega_1$  - angular speed of first shaft ( $s^{-1}$ ),
- $\omega_2$  - angular speed of second shaft ( $s^{-1}$ ),
- $\Delta_{\omega 1}$  - absolute change of angular speed of first shaft ( $s^{-1}$ ),
- $\Delta_{\omega 2}$  - absolute change of angular speed of second shaft ( $s^{-1}$ ),
- $\omega_{1n}$  - nominal angular speed of first shaft

( $s^{-1}$ ),

$\omega_{2n}$  - nominal angular speed of second shaft

( $s^{-1}$ ),

$\Delta G_T$  - absolute variation of fuel consumption (t/h),

$G_{Tn}$  - nominal fuel consumption (t/h),

$I_1$  - moment of inertia of rotating masses of first shaft

$I_2$  - moment of inertia of rotating masses of second shaft

$M_{K1}$  - moment of low-pressure compressor (Nm)

$M_{K2}$  - moment of high-pressure compressor (Nm)

$M_{T1}$  - moment of low-pressure turbine (Nm)

$M_{T2}$  - moment of high-pressure turbine (Nm)

## COMPUTER SIMULATION MODEL OF THE MARINE GAS TURBINE

### Scenario:

Run of the turbine is triple-stage, which mean that in TIME=1 second we accelerate turbine by bringing the fuel. On 10% of nominal number of revolution in TIME=5 seconds we increase fuel consumption to the 35% of nominal number of revolution, and in TIME=10 we increase fuel supply, and in that way obtain "uniform" heating of turbine i.e. nominal number of revolution. In TIME=15 PID regulator is under the shock loading from gas turbine, in amount of 50% from nominal load, meaning that  $F(t)=0.5$ . In TIME=30 stochastically load occurs.

### Graphics results of the simulation:

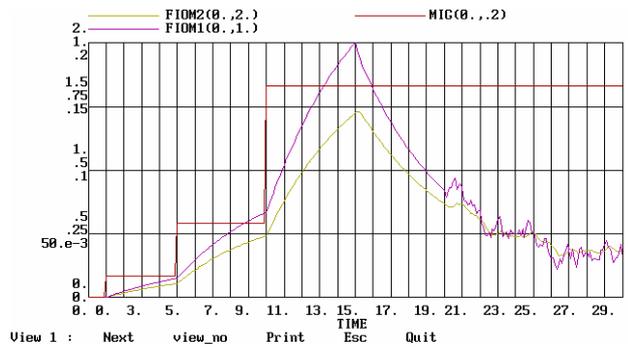


Figure 1. Relative angular speed variation, relative angular speed variation of second shaft, relative variation of fuel consumption

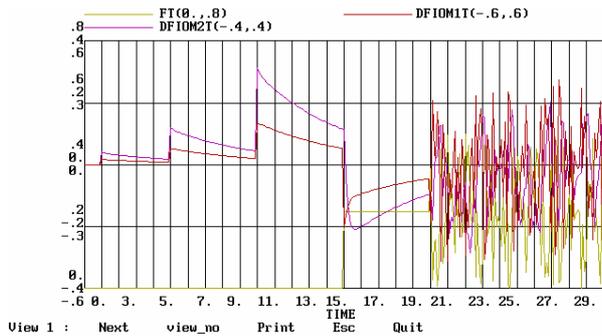


Figure 2. Speed variation of relative angular speed, speed variation of relative angular speed of second shaft

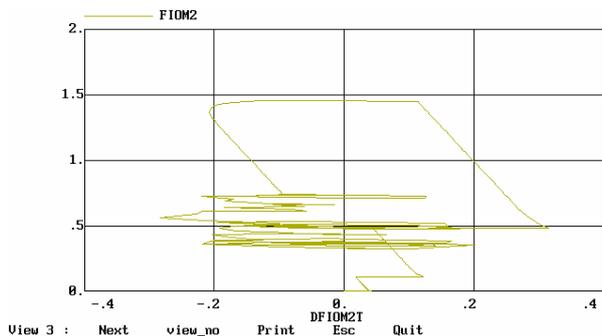


Figure 3. Dependence of relative angular speed of second shaft on its derivation (speed of change) diagram

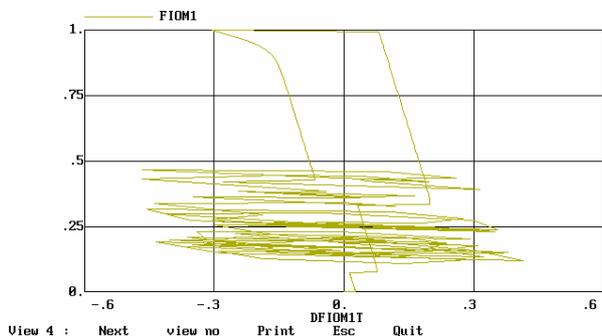


Figure 4. Dependence of relative angular speed of first shaft on its derivation (speed of change) diagram

## CONCLUSION

The application of System Dynamics Simulation Modelling Approach of the complex marine dynamic processes, which the authors, together with their graduate students, carried out at the Maritime Faculty University of Split - Croatia eleven years ago, revealed the following facts:

1. The System Dynamics Modelling Approach is a very suitable software education tool for marine students and engineers.

2. System Dynamics Computer Simulation Models of marine systems or processes are very effective and successfully implemented in simulation and training courses as part of the marine education process.

Finally, we may quote the Chinese proverb saying:

***“When I hear I forget. When I see I remember. When I work I understand”***

or we may express it a system dynamic way, i.e.:

***“WHEN I HEAR MENTAL-VERBAL MODEL OF DYNAMIC PROCESS, I FORGET”. “WHEN I SEE STRUCTURAL MODEL AND REALITY OF DYNAMIC PROCESS, I REMEMBER.” “WHEN I MAKE MATHEMATICAL OR COMPUTING SIMULATING MODEL OF DYNAMIC PROCESS, I LEARN”. “WHEN I MAKE SIMULATION OR EXERCISE EITHER ON SYSTEM DYNAMIC MODEL OR DYNAMIC PROCESS, I REFRESH MY TECHNICAL KNOWLEDGE WITH GAINED THEORETICAL AND PRACTICAL KNOWLEDGE ON DYNAMIC PROCESS”.***

## REFERENCES

- Forrester, Jay W. 1973/1971. "Principles of Systems", MIT Press, Cambridge Massachusetts, USA,
- Munitic, A., Milic M. and Milikovic M. 1997. "System Dynamics Computer Simulation Model of the Marine Diesel-Drive Generation Set", *Proceeding of World Congress on Scientific Computation, Modelling and Applied Matematchs*, Berlin,
- Munitic, A. 1989. "Computer Simulation with Help of System Dynamics", BIS, Croatia,
- Miler, J., 1955, "Parne i plinske turbine", Zagreb, Croatia,
- Richardson, George P. and Pugh III Aleksander L. 1981. "Introduction to System Dymanics Modelling with Dynamo", MIT Press, Cambridge, Massachusetts, USA,
- Munitic, A., 1989, "Application Possibilities of System Dynamics Modelling", *Proceeding of SCS Western Multiconference*, San Diego, California, USA,
- Нелепића, Р. А., 1975, "Автоматизација ендобних енергетических четвобок", Москва, Русија,
- Munitic, A., Kuzmanic, I., Krcum, M., 1998., "System Dynamic Simulation Modelling of the Marine Synchronous Generator", *Proceeding of Modelling and Simulation Conference, IASTED*, Pittsburgh, 372-375.
- Munitic, A., et al. 2002. "System Dynamics Modelling of Complex Electro Mechanical System", *IASTED, AMS 2002*, Cambridge, USA, 511-515.
- Munitic, A., Kulenovic, Z., Dvornik, J., 2003., "Computing Simulation of Driving System- "Ship/Piston Compressor-Electric motor", *IASTED MS 2003*, California, USA, 515-520.
- Munitic A., Antonic R., Dvornik J., 2003., "Computing simulation and heuristic optimization of ship anchor arrangement", *Proceeding of ICC'03*, SLOVAK, 353-357
- Munitic A., Antonic R., Dvornik J., 2003., "System dynamics simulation modeling of ship-gas turbine generator",

*Proceeding of ICC'03, SLOVAK, 357-360*  
Munitic A., Orsulic M., Dvornik J., 2003, "Computer Simulation of Complex Ship System "Gas turbine.Synhronous Generator" *Proceeding of ISC 2003, Valencia, Spain, 192-197.*

## BIOGRAPHY



**ANTE MUNITIC** was born 08. 26.1941. in Omis, near Split, Croatia! He received his first BSc. in Electrotechnics Engineering in 1968, and his second BSc. in Electronics Engineering in 1974; his MSc. degree in Electronics/Organization /Operational Research/Cybernetics Science in 1978, and his Ph.D. of Organization/Informatics Science

(exactly: System Dynamics Simulation Modeling) in 1983. He is currently a University Professor of Information/Computer Science at the University of Split, Croatia. Prof. Munitic has published over 100 scientific papers on system dynamics simulation modeling, operational research, marine automatic control system and The Theory of Chaos. He has published several books (as there are: "Computer Simulation with help of System Dynamics" and "Marine Electrotechnics and Electronics Engineering", .....). Today, he is professionally active university professor and scientist in the System Dynamics, Relativity Dynamics, System Dynamics Analogous Processes, Theory of Chaos and Informatics Scientific area.



**MARIO ORSULIC** received his B.Sc, M.Sc. and Ph.D. in mechanical engineering from Faculty of Mechanical Engineering University of Rijeka in 1968, 1984 and 1988 respectively. He is currently an associative professor at University of Split, College of Maritime Studies.

He is author or co-author of a number of bibliographical units (scientific and professional conference and journal papers, research projects, text books, etc.). His research interest is in Marine Engineering, auxiliary marine engines and technical mechanics theory and practise.



**JOSKO DVORNIK** was born 1978. at Split Croatia, were he finished elementary and high Maritime school. In school year 1996/97 he enrolled Maritime University in Split, Marine engineering Department, completed all theoretical and practical subjects

included in school program on time, and passed all exams. He graduated in 2000. year on theme "Application on computer simulation dynamics of behavior of ship propulsion system: windlass –

*asynchronous engine*", with very good degree as a first student in his class. Since December 2001. year he has worked as younger assistant at Maritime University in Split on scientific project titled "Computer simulation model of maritime educative system of Croatia". In June 2002. he has enrolled postgraduate study of engineering at Faculty of Mechanical Engineering and Naval Architecture. He has published 20 scientific papers on System Dynamics simulation modeling. He is a member of the SCS, The Society for Computer Simulation International.

# Prediction of Static Recrystallisation during Extrusion of Aluminium Alloy AA2024

Zhi Peng and Terry Sheppard  
School of Dec Bournemouth University  
S105, 12 Christchurch Road, Bournemouth, U.K.  
Email: Zpeng@bournemouth.ac.uk

## Keywords

Extrusion, Simulation, recrystallisation, aluminium

## Abstract

Extrusions experience large deformations at discontinuities when they traverse the die land, leading to considerable modifications to the average deformation parameters when compared to the remainder of the extrusion. The distribution of structure is therefore greatly inhomogeneous. Reference to both empirical and physical models of the recrystallisation process indicate that nucleation and growth will differ at these locations in those alloys that are usually solution treated and aged subsequent to the deformation process. In the work presented, a physical model based on dislocation density, subgrain size and misorientation is integrated into the commercial FEM codes, FORGE2<sup>®</sup> and FORGE3<sup>®</sup> to study the microstructure changes. Axi-symmetrical and shape extrusion are presented as examples. The evolution of the substructure influencing static recrystallisation is studied. The metallurgical behaviour of axi-symmetric extrusion and that of shape extrusion are compared. The predicted results show good agreement with experimental measurement.

## Introduction

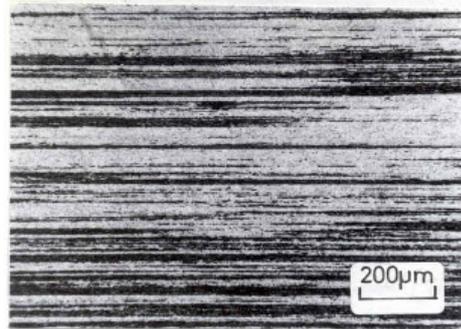
Aluminium and its alloys have good working characteristics in all the conventional metallurgical processes, such as rolling and extrusion. Since a range of commercial Al-Cu-Mg alloys has been developed, AA2024 and AA2014 are now the most widely used alloys of this system. The composition of AA2024 is shown below in table 1.

Table 1. Typical Composition of AA2024

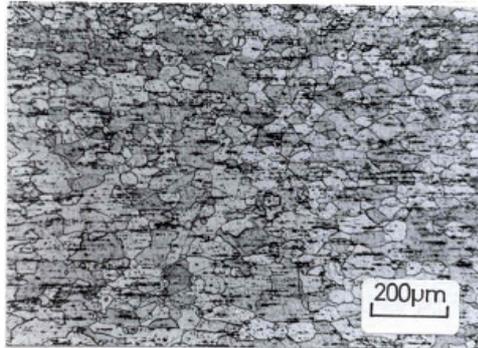
Alloy	Cu	Si	Mn	Mg	Fe
2024	4.5	-	0.6	1.5	0.2

In order to achieve the optimum mechanical properties in AA2024, it is usually necessary to solution heat treat and age the wrought product. Apart from the heavily worked outer region, the structure can be fully recrystallised, fibrous, or a combination of both. The structures may be classified into three main typical types as shown in Fig. 1. Frequently the structure was a

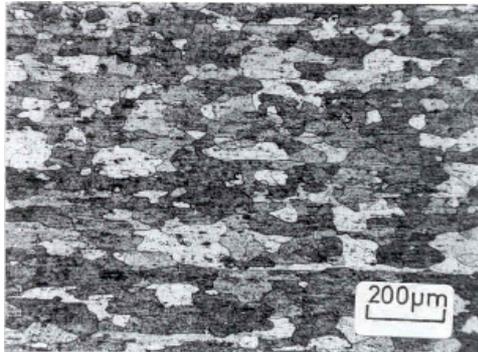
combination of one or more of these types (Sheppard 1993).



a)



b)



c)

Figures 1: Optical micrographs of typical structures

Micrograph (a) is of a typical fibrous structure with very long grains aligned in the extrusion direction. The similarity between this structure, formed at high temperatures, and that produced by a large reduction at

low temperatures, prompted much of the current interest in hot working and dynamic recovery.

In (b) an elongated structure can still be detected but recrystallization has occurred. Some of the recrystallized grains have their largest axis in the extrusion direction. The extent of the elongation is exaggerated by the aligned precipitates, they tend to make the grains appear longer than they are in reality. The fibrous structure occurs over a wide range of flow stress in the manganese-bearing alloy.

In (c) a fully recrystallized structure is seen. The only indication of the extensive deformation that has been used is the preferential orientation of the precipitates, which lie in the direction of extrusion. These precipitates appear to be much larger than those observed under the electron microscope. Two reasons for this may be given. One is the exaggeration of the size caused by the etchant. The other is that such large precipitate would be unlikely to thin down with the bulk matrix during the preparation of the thin foils.

In the extrusion of AA2024, due to the stored deformation energy within the extrudate, static recrystallisation usually occurs and extends to 100% of the material in some cases. The production of coarse grains is unbeneficial in subsequent heat treatment as it causes a reduction in mechanical property. Damage tolerance, fatigue crack propagation or corrosion, which are three very important technical indexes required by the aerospace industry, are significantly affected by the recrystallised grain size and the volume fraction recrystallised. It has also been shown that this problem becomes greater as the complexity of section shape increases.

Hence, knowledge of the variation of the recrystallised grain size with time and space assists optimisation of the extrusion process.

### Metallurgical Model

By empirical and physical means, a modest degree of prediction of microstructure can now be achieved. Excellent reviews on modelling of static recrystallisation (SRX) have been given by Gottstein et al. (Gottstein et al. 1999; Marx et al. 2000) and by Shercliff and Lovatt (Shercliff 1999). Some of the modelling work has been achieved in the field of hot rolling (Chen et al. 1992), and recently in the field of hot extrusion by Duan and Sheppard (Duan and Sheppard 2002). Some models introduce many tuning parameters, especially for the physically based models. These parameters depend mainly on the material. To estimate their real values, specific and numerous experiments would be required. Recently, the inverse method combined with FEM has been adopted to tune the values of these parameters. The FEM is run iteratively until the appropriate value is found to match the experimental measurement. Duan and Sheppard

(Duan and Sheppard 2003) have used the inverse method to give the parameters for alloy 5083 and 2014.

The relationship between the volume fraction recrystallised ( $X_v$ ) and the holding time ( $t$ ) is generally represented by the Johnson-Mehl-Avrami-Kolmogorov equation (JMAK), which predicts the relationship between the volume fraction recrystallised ( $X_v$ ) and the holding time ( $t$ ) and is generally represented as:

$$X_v = 1 - \exp\left\{-0.693\left(\frac{t}{t_{50}}\right)^k\right\} \quad (1)$$

where  $t$  is annealing time,  $k$  is the Avrami exponent with a commonly reported value of 2,  $t_{50}$  is the time to 50% recrystallisation. For the calculation of  $t_{50}$ , the physical model is commonly regarded as revealing the mechanics driving the transformation. Previous studies (Furu et al. 1999) have shown that the physical models describe the experimental results well for uniform processing conditions. The model was also successfully applied to tests in which the strain rate was increased (when microstructure transients were not observed). Recently, Sheppard and Duan (Duan and Sheppard 2002) have confirmed that the physical model will give better computed results than the empirical model in the simulation of aluminium extrusion. Only the physical model proposed by Furu and Zhu et al (Zhu and Furu 2000) has been used in this study.

In equation (9),  $t_{50}$  is calculated based on the stored energy ( $P_D$ ) and the density of recrystallisation nuclei ( $N_V$ ) (Furu 1999).

$$t_{50} = \frac{C}{M_{GB} P_D} \left(\frac{1}{N_V}\right)^{1/3} \quad (2)$$

where  $C / M_{GB}$  is a further calibration constant.  $N_V$  is defined as:

$$N_V = (C_d / \delta^2) S_V(\varepsilon) \quad (3)$$

where  $C_d$  is a further calibration constant,  $\delta$  is the subgrain size,  $S_V$  is the grain boundary area per unit volume

$$S_V(\varepsilon) = (2/d_0)[\exp(\varepsilon) + \exp(-\varepsilon) + 1] \quad (4)$$

The stored energy  $P_D$  is approximated by

$$P_D = \frac{Gb^2}{10} [\rho_i (1 - \ln(10b\rho_i^{1/2})) + \frac{2\theta}{b\delta} (1 + \ln(\frac{\theta_c}{\theta}))] \dots\dots\dots(5)$$

where  $G$  is the shear modulus,  $b$  is the burgers vector,  $\rho_i$  is the internal dislocation density,  $\theta$  is the misorientation and  $\theta_c$  is the critical misorientation for a high angle boundary ( $\sim 15^\circ$ ).

The evolution of  $\delta$ ,  $\rho_i$  and  $\theta$  has been explicitly expressed in differential form based on the most classical theories of work hardening and dynamic recovery.

$$d\delta = \frac{\delta}{\varepsilon_\delta \delta_{ss}} (\delta_{ss} - \delta) d\varepsilon \quad (6)$$

$$d\theta = \frac{1}{\varepsilon_\theta} (\theta_{ss} - \theta) d\varepsilon \quad (7)$$

$$d\rho_r = d\rho_r^+ + d\rho_r^- = (C_1 \rho_r^{1/2} - C_2 \frac{\sigma_f}{Z} \rho_r) d\varepsilon \quad (8)$$

where  $\delta_{ss}$  and  $\theta_{ss}$  are the subgrain size and misorientation at steady state deformation.  $\varepsilon_\delta$  and  $\varepsilon_\theta$  are characteristic strains,  $\rho_r$  is random dislocation density,  $C_1$  and  $C_2$  are constants. The internal dislocation density consists of two parts,  $\rho_r$  and  $\rho_g$  (the geometrical necessary dislocation density).

$$\rho_i = \rho_r + \rho_g \quad (9)$$

$$\rho_g = \frac{1}{b} \left( \frac{1}{R_g} - \frac{\theta}{\delta} \right) \quad (10)$$

where  $\rho_i$  is the internal dislocation density,  $1/R_g$  is the local lattice curvature.

For site-saturated nucleation, the recrystallised grain size is simply calculated from nucleation density as

$$d_{rex} = DN_V^{-1/3} \quad (11)$$

where  $D$  is a constant.

## Experimental Procedure

The material used for the current research work was supplied by Alcan Labs, Banbury, in the form of semi continuous logs of 86mm diameter. The billets were homogenised prior to extrusion at  $500^\circ C$  for 24 hours and furnace cooled. The homogenised billets were then machined to a diameter of 73mm and cut into billets of required length of 95mm. Specimens for mechanical testing, for heat treatment and for optical and electron microscopy were cut from a position one third along the extrudate in order to ensure steady state conditions. Specimens were also taken along the length of the extrudate to determine the range of properties. The final 60cm of the extrusion was never used since this region

remained unquenched. Transverse and longitudinal sections of 2-3 mm thickness were cut from the extrudates and grounded to a thickness of 0.25 – 0.3 mm on the silicon carbide paper. 3 mm discs were punched from these sections and electropolished in a commercial struers jet thinner. The solution was maintained at  $-30^\circ C$  and a potential of 13 volts was applied between the discs and solution (Sheppard 1993).

## FEM Simulation Setting

The FEM programs, FORGE2<sup>®</sup> and FORGE3<sup>®</sup> are used in the present study. It is a process simulation tool based on the Finite Element Method. The hyperbolic sine function was combined into the FEM to describe the material behaviour. The constitutive equation can be written as:

$$\bar{\sigma} = \frac{1}{\alpha} \text{Ln} \left[ \left( \frac{Z}{A} \right)^{\frac{1}{n}} + \left[ \left( \frac{Z}{A} \right)^{\frac{2}{n}} + 1 \right]^{\frac{1}{2}} \right] \quad (12)$$

where  $\alpha, A, n$  are temperature independent constants,  $Z$  is the Zener-Hollomon parameter,

$$Z = \dot{\varepsilon} \exp \left( \frac{\Delta H}{RT} \right) \quad (13)$$

where  $\dot{\varepsilon}$  is the strain rate,  $\Delta H$  is the activation energy and  $T$  is the temperature (Sheppard 1993).

For aluminium alloy AA2024,  $\Delta H = 148880 \text{ KJ/mol}$ ,  $\alpha = 0.016 \text{ m}^2 / \text{MN}$ ,  $n = 4.27$ ,  $\text{Ln}A = 19.6$ .

## Two-dimensional simulation results

An axi-symmetrical simulation was performed by Forge2<sup>®</sup> to check the effect of the metallurgical model. The simulation settings are shown in Table 2. The container temperature is 50 K lower than the initial billet temperature. The friction coefficients between the billet and the die and between the billet and the ram are set as 0.3. The extrusion ratios is 40.

The predicted recrystallized grain size and the subgrain size corresponded well with the experimental measurement along the transverse direction, as can be seen from figure 2,3. It is easy to see from figure 2 that the difference between the calculated subgrain size at the centre ( $2.27 \mu\text{m}$ ) and the experimental measurement ( $2.22 \mu\text{m}$ ) is no more than 1.0%. At the edge of the extrudate, the predicted subgrain size is  $2.52 \mu\text{m}$ , which is just 0.3% higher than the experimental result ( $2.45 \mu\text{m}$ ). It is clear from figure 2 that the predicted subgrain size increases as the temperature rises along the transverse direction of the extrudate. This phenomenon is the same as that observed previously (Sheppard 1993). From figure 3, the recrystallized grain size shows a sharp decrease at the surface of the extrudate, and it is easy to see the

Table 2. Two-dimensional Simulations

Extrusion mode	Material used	Billet temperature (Kelvin)	Die fillet radius	Ram speed (mm/s)	Friction coefficient
Direct	AA2024	683	0.5	3.0	1.0

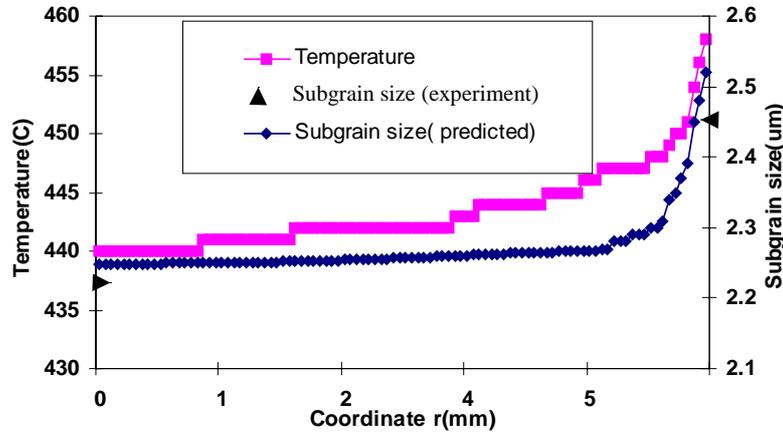


Figure 2. Subgrain Size and Temperature

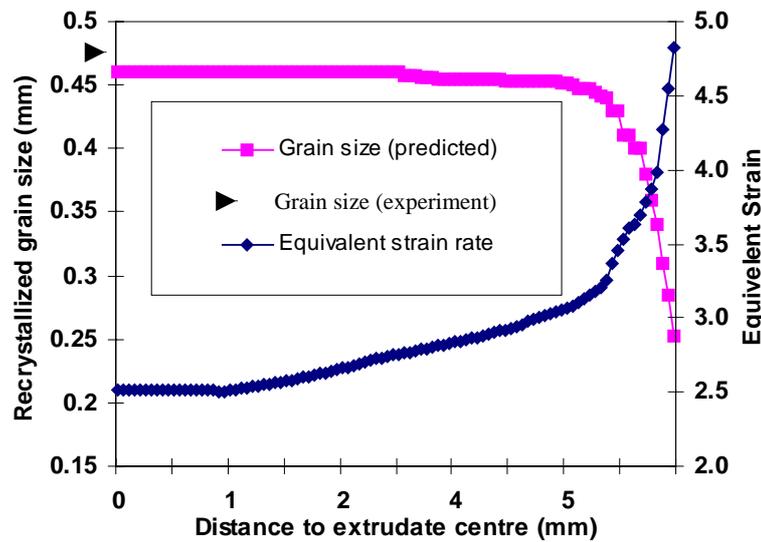


Figure 3. Recrystallized Grain Size and Equivalent Strain

grain size is in inverse proportion to the equivalent strain. This has also been observed before by Vierod and many others (Sellars and Zhu 2000). The predicted value is 0.455mm, which is 3.2% lower than the experimental results.

For the predicted result of volume fraction recrystallisation, the simulation give a slightly rising prediction along the extrudate surface, as can be seen from figure 4. This phenomenon has been observed in the previous experiment. It should be noticed that the experimental measurement, which is 27.37%, is an average value along the extrudate. It is difficult to compare the variation of the predicted  $X_v$  with the averaged experimental measurement. The method used in this study to solve this problem was to find a point, whose Y coordinate gave the best correspondence to the

experimental result in the given curve. At the same time, the X coordinate of this point and the running step of the simulation were picked out, and the predicted  $X_v$  of the other simulations were obtained from the point with the same coordinate and at the same time step. Because the rise of  $X_v$  during the extrusion is small, the value picked out from the point can be regarded as the mean value along the extrudate.

### Three dimensional simulation results

In this paper, two types of shape extrusion, the T shape and the U shape, are studied. The dimensions of these two shapes are shown in figure. 5. The simulation runs are shown in table 3.

Table 3 Shape Section Extrusion

runs	Type	ratio	temp (Kelvin)	speed (mm/s)
1	T	40	623	7
2	U	40	623	7

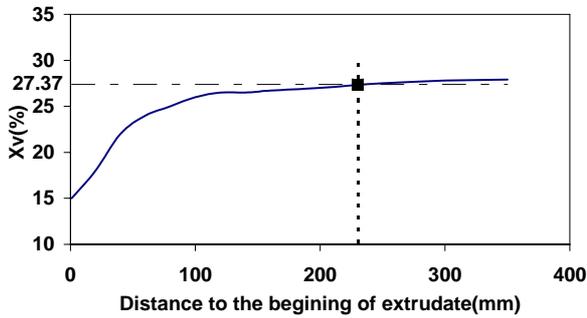


Figure 4. Predicted Xv along Extrudate Surface and the Selected Point

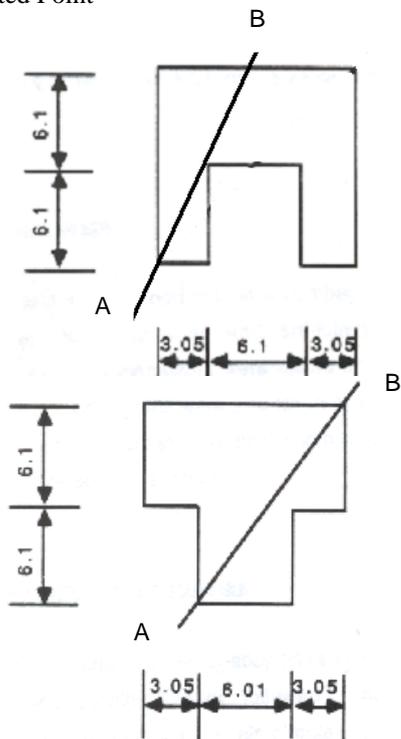
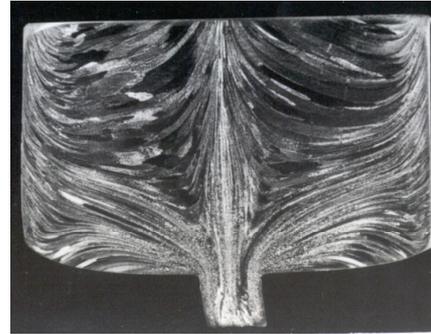


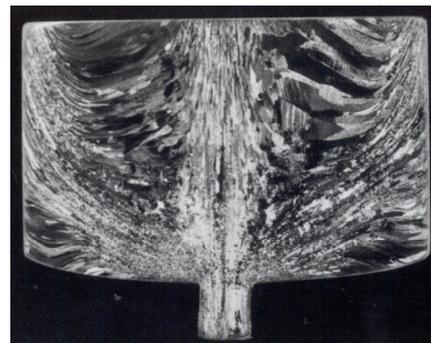
Figure. 5 Dimensions of Shape Sections

Compared with the axi-symmetrical extrusion, the material flow is inhomogeneous in shaped extrusion. The flow patterns corresponding to the extrusion of rod and shape sections are shown in figure 6. The shape sections are cut from face AB, as shown in figure 5. It can be seen from figure 6 that though the general flow pattern remains similar to that in rod extrusion, a certain amount of asymmetry about the billet axis can be envisaged, especially in the regions close to the die shoulders (dead metal zone). The asymmetrical material flow pattern has an important influence on the metallurgical behaviour during shape extrusion, which will be discussed below.

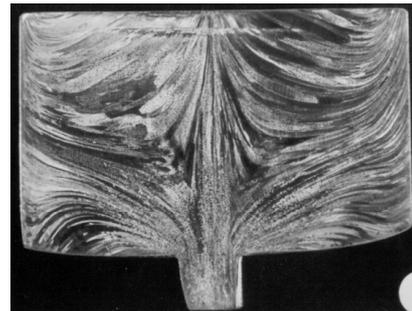
The recrystallised grain size and the equivalent strain distribution of the two sections are shown in figures 7-10. As can be seen in the figures 8 and 10, unlike the axi-symmetrical extrusion, the recrystallised grain size around the periphery of the shape section is inhomogeneous. The inverse relationship between the recrystallised grain size and the equivalent strain also exists in shape sections. In the area where sharp deformation occurs, the recrystallised grain size is smaller than the other areas.



a) Rod



b) U shape



c) Tshape

Figure 6 Macrosections of Partially Extruded Billets

Table 4 Fraction Recrystallised Factors of the T Shape Extrusion

Point	Equivalent strain	Xv	Point	Equivalent strain	Xv
1	2.1	0.61	8	3.19	0.7
2	3.37	0.6	9	3.36	0.78
3	3.04	0.65	10	3.38	0.65
4	2.97	0.60	11	3.40	0.72
5	4.06	0.69	12	3.17	0.62
6	4.15	0.82	13	3.4	0.66
7	4.81	0.72			

Table 5 Fraction Recrystallised Factors of the U Shape Extrusion

Point	Equivalent strain	Xv	Point	Equivalent Strain	Xv
1	2.62	0.95	6	4.19	0.99
2	3.42	0.99	7	4.92	0.99
3	4.62	0.99	8	3.9.3	0.99
4	4.4	0.99	9	5.37	0.99

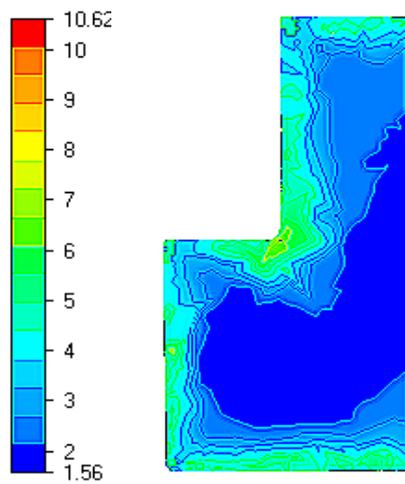


Figure. 7 Equivalent Strain Distribution

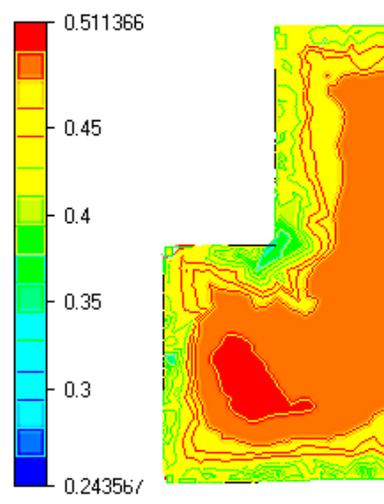


Figure. 8 Recrystallised Grain Size

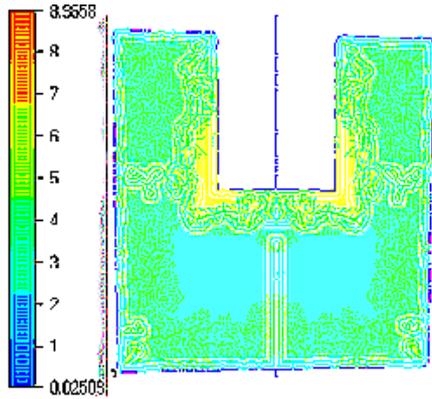


Figure. 9 Equivalent Strain Distribution

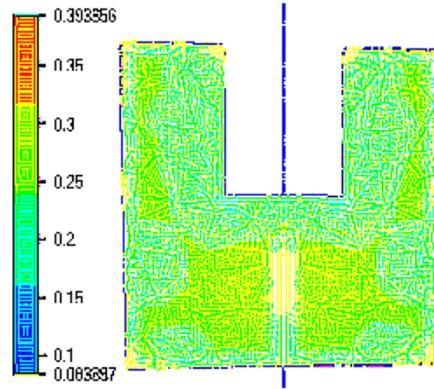
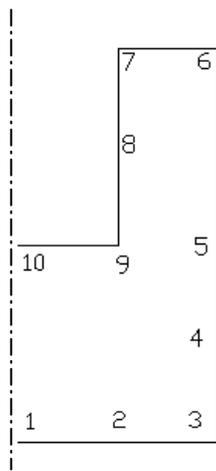
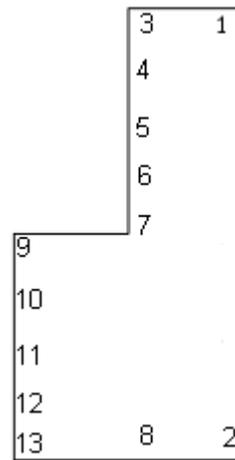


Figure. 10 Recrystallised Grain Size

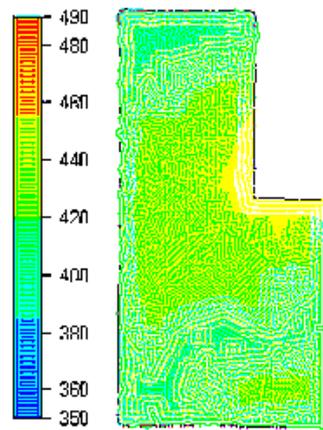


(a) U shape

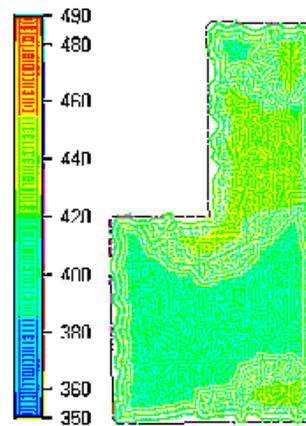


(b) T shape

Figure 11 Position of the Points in Table 4



(a) U shape



(b) T shape

Figure 12 Temperature Distribution

The predicted fraction recrystallised factors at the periphery of the two sections are shown in table 4 and table 5 respectively. The positions of the points used in these tables are shown in figure 11. For shaped sections, estimation of the depth of the recrystallised layer was difficult in previously experiments, since the

layer was no longer uniform. With FEM simulation, the distribution of the volume fraction recrystallised factors can be predicted and extracted more easily. It has been found that both in previous experiments and FEM simulations that the recrystallised layer was thicker for more complex sections due to larger

temperature rise. As can be seen in table 4 and 5, at the positions where the deformation is more complex, that is, where the strain and the temperature are higher, the volume fraction recrystallised is also higher. The temperature distribution across the shape extrudates is shown in figure 12. It can be seen that at the place where a sharper deformation occurs, the temperature and the volume fraction recrystallised are higher, but at the same time, it should be noted that at the corners, the temperature is lower than the other places. It is easy to see that the U shape section experiences larger deformation than the T shape. After the shape sections been extruded to the same distance, the temperature rise and the equivalent strain of the U shape are higher than that of the T shape. The volume fraction recrystallised is also much more significant in the U shape than the T shape.

It can be seen from the discussion above that, due to the different flow pattern to axi-symmetrical extrusion, the recrystallised grain size and the volume fraction recrystallised factor are inhomogeneous around the periphery of the sections.

## Conclusion

FEM simulation is effective in predicting the metallurgical behaviour happened during extrusion. The calculated recrystallised grain size, the subgrain size and the volume fraction recrystallised in this study are in reasonable agreement with the previous experimental results.

## Reference

- Aretz, H.; R. Luce.; M. Wolske.; R. Kopp.; M. Goerdeler; V. Marx.; G. Pomana.; and G. Gottstein. 2000. "Integration of physically based models into FEM and application in simulation of metal forming processes." *Modelling and Simulation in Materials Science and Engineering*, 8 (Nov), 881-891.
- Chen, B. K.; Thomson, P. F.; and Choi, S. K. 1992. "Computer modelling of microstructure during hot flat rolling of aluminium." *Materials Science and Technology*, 8 (Jan), 72-77.
- Duan, X. and T. Sheppard, 2002. "Influence of forming parameters on static recrystallization behaviour during hot rolling aluminium alloy 5083." *Modelling and simulation in materials science and engineering*, 10, 363-390.
- Duan, X. and T. Sheppard. 2002. "Simulation of substructural strengthening in hot flat rolling." *Materials processing Technology*, 125-126, (Sep), 179-187.
- Duan, X. and T. Sheppard. 2003. "Computation of substructural strengthening by the integration of metallurgical models into the finite element code." *Computational Materials Science*, 27, 250-258.
- Furu, T.; H. R. Shercliff; C. M. Sellars; and M. F. Ashby. 1996. "Physically-based modelling of strength, microstructure and recrystallisation during thermomechanical processing of Al-Mg alloys." *Mater. Sci. Forum.* 217-222, 453-458.
- Furu, T.; H. R. Shercliff; G. J. Baxter.; and C. M. Sellars. 1999. "Influence of transient deformation conditions on recrystallization during thermomechanical processing of an Al-1% Mg alloy." *Acta. Mater.* 47, 2377-2388.
- Johnson, W. and H. Kudo. 1962. *The mechanics of metal extrusion*, Manchester University Press, Manchester, 60-67.
- Marx, V.; F. R. Reher.; and G. Gottstein. 1999. "Simulation of primary recrystallization using a modified three-dimensional cellular automaton." *Acta Materialia*, 47, 1219-1230
- McLaren, A. J. 1994, Phd Thesis, University of Sheffield.
- Sellars, C. M. and Q. Zhu. 2000. "Microstructural modelling of aluminium alloys." *Microstructure and Processing*, 280, 1-7.
- Sheppard, T. 1993. "Extrusion processing parameter-mechanical property correlations in rapidly solidified Al-6.7Fe-5.9Ce and Al-6.2Fe-5.9Ce-1.63Si (wt-%) alloy powders" *Material Science and Technology*, 9(May), 430-440.
- Shercliff, H. R. and A. M. Lovatt. 1999. "Modelling of microstructure evolution in hot deformation." *Phil. Tran. R. Soc. Lond. A*, 357 1621-1633.

## AUTHOR BIOGRAPHIES

**Professor Terry Sheppard** has spend his working life in the metals industry. He occupied senior posts with Loewy Engineering and Tube Investment before moving to Imperial College, London where he developed research in the metal forming area for thirty years and was Professor of Industrial Metallurgy. He retired from Imperial College in 1991 and since then he has developed his consultancy work, especially applying his learning to practical extrusion. He also holds a part-time position as Professor of Production Technology at Bournemouth University. His e-mail is: [Tsheppar@bournemouth.ac.uk](mailto:Tsheppar@bournemouth.ac.uk)

**Zhi Peng** was born in China and now study in the University of Bournemouth, U.K. for his Ph.D's degree. His e-mail address is: [Zpeng@bournemouth.ac.uk](mailto:Zpeng@bournemouth.ac.uk)

# SUPPLY CHAIN SIMULATION: EXPERIENCES FROM TWO CASE STUDIES

Fredrik Persson  
Department of Production Economics  
Linköping Institute of Technology  
S-581 83, Linköping, Sweden  
E-mail: fredrik.persson@ipe.liu.se

## KEYWORDS

Supply Chain Management, Simulation, Supply Chain Simulation.

## ABSTRACT

Analysing supply chains utilising discrete event simulation allows the analyst to take on a dynamic approach to system analysis. This paper outlines the research area of supply chain simulation and reports on two case studies from the Swedish electronics industry. Starting with a descriptive case study of a company's transition into supply chain management, the case continues to study how the level of detail in simulation models affects the simulation results and also to analyse the upstream supply chain of the same company. In the other case study, a supply chain is analysed using two different approaches. First, the interaction between quality, cost, and lead-time is evaluated using simulation. Second, screening and robust optimisation is applied to the same simulation models.

## INTRODUCTION

One of the research topics covered by Production Economics is supply chain management. Supply chain management incorporates the use of analysis tools such as system dynamics, optimisation, and simulation. Theoretical models of the supply chain behaviour can be created by observing the supply chain's historical data or by collecting new data. Experiments that study the supply chain behaviour are useful in order to find causal effects and to test different or even extreme scenarios. Causal effects are however difficult to find if they are separated in time and space and extreme scenarios are hard to control in a supply chain. An alternative to conducting experiments in the system is of course to use a model of the system for experimentation.

Many of the supply chain models found in the literature are models that are used for optimisation. These models are used to answer questions about plant location, product mix, technology choice, means of distribution, inventory planning and control, vendor choice, configuration, and reverse logistics; see Goetschalckx *et al.* (2002) and Shapiro (2001) for extensive work on supply chain optimisation models. Optimisation models consider the supply chain at specific instances in time and

do not take on a dynamic view like in simulation. Optimisation models often lack the estimate of the variability or robustness of a solution in a stochastic environment. Metrics such as lead-time variability, percent of on-time delivery and so on, are hard to obtain in using an optimisation model. In a recent literature review, Goetschalckx *et al.* (2002) examine seven different modelling approaches for global logistics systems using mathematical programming. Only one model in the review utilised stochastic lead-times and only a few included other stochastic characteristics. Stochastic characteristics are an important factor of supply chains. Especially a stochastic demand is regarded as having great impact on financial performance (Chwif *et al.* 2002).

Bekker and Saayman (1999) distinguish between time based and non-time based modelling techniques in logistics. They define time based as time driven and characterise simulation as a time-based technique. The advantage of time-based techniques, such as simulation, is the ability to include the stochastic nature of a system over time, while the non-time based techniques, such as optimisation often exclude this behaviour.

There is a methodological and practical difference in the way optimisation and simulation finds optimal solutions. In optimisation, the solution is dependent on the scenario that defines the experimental domain (cf. Zeigler *et al.* 2000). The optimal solution is only valid for that scenario and will become invalid if the scenario changes. In simulation it is possible to experiment with a set of scenarios in order to find a robust solution. The robust solution is not optimal but minimises/maximises the objective function that is subject to a set of scenarios. This solution is not as sensitive for environmental changes as the optimal solution obtained through optimisation.

Simulation in supply chain management offers a complement to the more prevailing modelling using optimisation models since simulation is more suited for representing random effects and predicting the dynamic behaviour of supply chains. This paper outlines the research area of supply chain simulation and reports on two case studies from the Swedish electronics industry. Starting with a descriptive case study of one company's transition into supply chain management, the case continues to study how the level of detail in simulation models affects the simulation results and also to analyse

the upstream supply chain of the same company. In the other case study, a supply chain is analysed using two different approaches.

## SUPPLY CHAIN SIMULATION

Forrester (1961) developed industrial dynamics (also known as systems dynamics) as a tool for systems analysis. Thus, systems dynamics have been used for supply chain simulation in over 40 years (Towill 1996). Later, supply chain simulation developed to include other simulation methodologies as well such as discrete event simulation.

Supply chain simulation defined as the use of simulation methodology, incorporating discrete event simulation technology, to analyse and solve problems found relevant to supply chain management.

The main reasons to use discrete event simulation for supply chain management related problems are (i) the possibility to include dynamics and (ii) the simplicity of modelling. Discrete event simulation is suited for these kinds of studies where time-dependant relations are analysed. Simulation also has a capability of capturing uncertainty and complexity that is well suited for supply chain analysis (Jain *et al.* 2001). Manivannan (1998) provides different examples of supply chain simulation including warehousing and distribution systems and trucking operations, among others. In another example of supply chain simulation, Bhaskaran (1998) provides a technique to analyse supply chain instability and inventory levels. Simulation is used to link the dynamic behaviour of a supply chain with the cost calculations possible in a mathematical programming model.

Jain *et al.* (2001) point out the model's level of detail as being one of the major difficulties in supply chain simulation. It is not uncommon to simulate at a level of detail that does not match the objective of the analysis. The choice of the level of detail is therefore an important issue in supply chain models. The model must also be credible in order for the results to be useful. Validation of a supply chain model can be a difficult task because of lack of data and lack of system experts. Manivannan (1998) points out the complex nature of supply chains as one of the main obstacles in supply chain simulation. Other modelling challenges that Manivannan highlights are the missing support for logistic processes in simulation software and unfamiliarity to simulation in the logistics industry. The wide use of optimisation methods in logistics and the fact that many problems have a closed-form solution are other challenges for supply chain simulation.

Banks *et al.* (2002) discuss what makes supply chain simulation different from other simulation applications. One major difference from e.g. simulation of manufacturing systems is that supply chain models contain information flows together with the flow of materials. The importance of handling different levels of detail is made

apparent in the case of supply chain simulation. Different actors in the supply chain store data in different ways, which makes data collection harder. It is therefore difficult to model the whole supply chain at the same (desired) level of detail. The difficulty with different levels of detail together with the size of supply chain models tend to make the model building process take a longer time in supply chain simulation. To experiment with supply chain simulation models often include a large number of alternative scenarios demanding efficient experimental planning. Validation is another field where supply chain simulation meets difficulties. Subjective methods such as walkthroughs are hard to accomplish on the supply chain level. Sub-model validation is a way around the problem with huge systems since the problem is to get system experts with detailed knowledge about the *whole* supply chain.

## CASE STUDY I: SUPPLY CHAIN TRANSITION AND MODELLING

The first case study concerns a company in the mobile communications industry. Olhager *et al.* (2002) deals with the impact of supply chains on operations management at the case company. It specifically deals with the transition of one of the plants from being a production unit to the role of a supply unit in a supply chain. This change has had a large impact on many factors related to supply chain structure and flexibility, reengineering of the information flow, the management of the supply process, and on performance measurement.

The role in the supply chain emphasises that a proactive approach for the operations at the supply units is necessary. Before introducing new technology, new solutions, and new supply paths, these must be tested, debugged, and corrected. This calls for simulation approaches especially with respect to new supply chain or supply network structures.

The change experienced at the case company has also been a change towards mass customisation. When mapping this move relative the product-process matrix by Hayes and Wheelwright (1984) and product profiling by Hill (1995) a misalignment between markets and manufacturing is seemingly appearing. However, the line flow process with a short set-up time is able to accommodate multiple product variants. Total product volumes are high and increasing. Consequently, rate-based and JIT/pull approaches can be utilised even though products are built to order. The role of the operations manager in the supply chain setting is to build structures and to empower the organisation. Flexibility must be built into these structures. The delivery team managing the order process embodies this empowerment.

For the same case company, Persson (2002) investigates how the choice of level of detail influences the outcome of a simulation study. To investigate the impact of a varying level of detail, a part of the manufacturing system is modelled using three different levels of detail.

The first model is built at a high level of detail containing all elements in the system. The second model contains aggregations of some of the processes in the system and the third model consists only of the main process. The experiments with the models aims at finding differences between the three models' outputs that originates from the choice of the level of detail. The results show that there are significant differences between the models.

The credibility of the results clearly depends on how a study is carried out. In simulation, this is done in the activities of validation and verification. The models outputs show great differences although the models are validated and verified. Essentially this difference in modelling originates from the chosen level of detail. Validation of simulation models must therefore include all simulation outputs used in a study. Output data must also be analysed using aggregates instead of single outputs.

Following the same case study, the analysis in Persson (2003) focuses on the supply chain's dynamic behaviour. The case company defines three different upstream routes for supply of electronic components and mechanical parts for mobile communications manufacturing. The first route concerns traditional purchasing, where components are shipped to the manufacturing plant and delivered into stock. This concept is here called direct supply, DS. The second route includes the use of vendor-managed inventory, VMI, at the manufacturer's plant. The components are called off when they are needed in production. The third route is a supply logistics centre, SLC, run by a third party. At the SLC components are held in stock by the company's suppliers and shipped in sets to the manufacturing plant at predefined time intervals. In this case, the SLC supplies two manufacturing plants with around the clock shipments.

Persson (2003) analyses the sensitivity of the three supply routes with respect to cost, lead-time, and lead-time variability. The results of the study indicate that the influence from a decreasing yield in the manufacturing process can be limited to the inventory at the manufacturing line and the use of an SLC relaxes the supplier from influences of manufacturing process yield. Managerial implications of the findings in this study are both specific and general. In the specific situation of the case company, the model provides useful data on the concepts of the different supply routes. The SLC concept is supported by the simulation results since it provides a buffer towards the suppliers of the SLC products. Thereby, the effect of yield level on supply is reduced. The company predefine the choice of supply route for each product in this study. In general, products that use the VMI and direct supply routes are characterised by the level of co-operation with the suppliers. Suppliers with high volume standardised products use the VMI mode while more customised products, such as inte-

grated circuits, make use of the direct supply route. Products that are assembled late in the process and that have a high degree of customisation make use of the SLC concept. The SLC provides opportunities for coordination of supply to the manufacturing line and quality tests at the SLC. While relaxing the suppliers from the changes in yield, no increase in work in process can be detected. However, the transportation cost with hourly transports between the SLC and the manufacturing plant can not be neglected.

In more general terms, the model shows that the supply chain under the influence of fluctuating demand due to the changing process yield can handle most of the fluctuations in all investigated supply routes. However, the route that takes care of the fluctuations in the most effective way is the longest route, i.e. the route that contains the most intermediate inventory. In the case of the SLC route, the supply chain is expanded with an extra inventory holding stage compared with the DS and VMI routes. Therefore, the fluctuating demand of supply is limited to the manufacturing line. This means that the concept of a third party solution for the SLC provides the company with an independent upstream time buffer. This can be seen in Figure 1, the closer the inventory is stored to the manufacturing line, the more influenced by changing yields it gets.

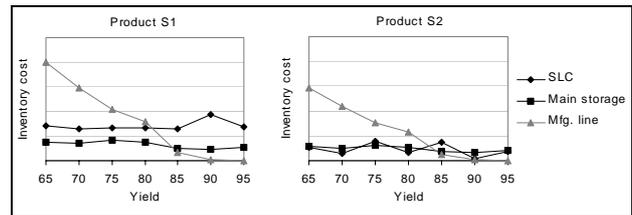


Figure 1: Inventory cost for the two products that utilise the SLC concept

Olhager et al. (2002) describes the transformation process from production units to demand-driven supply units. The transformation included a change in material supply routes. Vendor-managed inventory was already in use at the company and the benefits of that system was incorporated in supply logistics centres (SLC). The SLC concept is supported by the results in Persson (2003). The results of the simulation study show that the SLC provides a buffer towards the supply unit, such that environmental disturbances such as yield changes have little effect on the supply from the SLC. At the same time the SLC concept can incorporate the use of VMI and direct supply to supply the SLC. As the results from this case study the case company points out the advantage with having a single global provider for the SLC instead of local providers. This allows for materials to be re-routed between the SLCs as the demands for specific components change with the product mixes at different supply units. The SLC handles material used for packaging and mechanical components used in assembly.

## CASE STUDY II: SUPPLY CHAIN SIMULATION ANALYSIS

In the second case study, Persson and Olhager (2002) reports on a supply chain simulation analysis. The purpose of this paper is twofold. First, alternative supply chain designs with respect to quality (product yield is equal to quality in this case), lead times and costs are evaluated. Second, the interrelationships among these and other parameters are analysed. The study concerns three different instances of the same supply chain, the old, the current, and the next generation supply chain design.

The results in terms of performance measures such as total cost, inventory holding, quality, lead-time, and lead-time variability shows some interesting interactions. Lead-time variability increases between the old and the current supply chain, even though lead times are reduced. However, for the next generation supply chain design, both lead-times and lead-time variability are expected to reduce considerably, thereby providing both shorter and more reliable lead times. The other performance measures improve with an improvement of quality and supply chain structures. The model capturing the relationships among total cost, quality and lead-time, indicates that total cost increases more than linearly with lead-time, see Figure 2. Also, the level of non-linearity increases with reduced quality levels. Consequently, low quality in supply chains with long lead-times is devastating to supply chain performance. Inversely, good quality and short lead-times in integrated and synchronised supply chains will lead to superior performance. The payoff in terms of total cost is more than proportional to the improvements in quality and lead-times, the latter largely a result of improved supply chain designs.

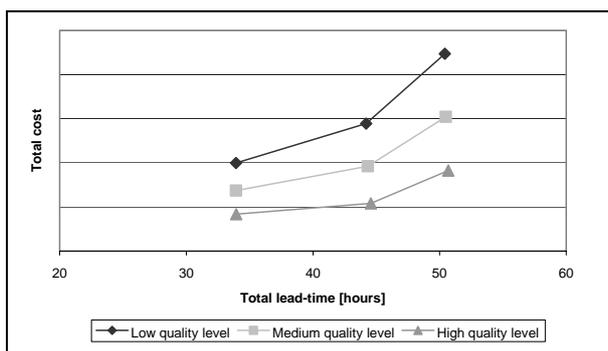


Figure 2: Relationship between total cost, lead-time, and quality level (cost levels are confidential)

To capture the influence of different yield levels on lead-time, the use of “scrap-inflated” lead-times as a performance measure is introduced in Persson and Olhager (2002). When a product is scrapped its lead-time is added to the next product that is entered into the simulation model, thereby adding to the lead-time cal-

culations. These lead-times get shorter as the yield level (or quality level) improves. Scrap-inflated lead-times correspond to the time that capital is tied up in work in process, and, therefore, provide a better interpretation in terms of time for work-in-process inventories. Lead-times that ignore the scrap effect would underestimate the true work in process levels.

Kleijnen, Bettonvil, and Persson (2003) expands the analysis of Persson and Olhager (2002) by including screening and robust optimisation. In the paper, the three alternative designs from Persson and Olhager (2002) are subject to a screening process using sequential bifurcation. This group screening strategy is previously used for deterministic simulations. The number of experimental factors is reduced from 92 factors to 11 in the most extreme case. This is achieved in only 42 simulations (including mirror observations of each run)

An experimental design is also applied to the important factors in the model in order to find a robust optimal solution for the case company. The factors are divided into two groups, one consisting of all controllable factors and one consisting of all environmental factors. The case company can directly control controllable factors while the environmental factors are outside of control and can be considered as disturbances. In the experimental design, a central composite design for the controllable factors is crossed with a Latin hypercube sampling design for the environmental factors.

The results show that all of the controllable factors, that are important according to the screening process, show significant effects in the regression analysis. As was suspected, several two-way factor interactions also show significant effects. Most interesting is the presence of interactions between the controllable and environmental factors. These interactions indicate that the case company can counteract eventual changes in environmental factors by changing their controllable factors.

The next generation supply chain design, the design with the shortest lead-time and lowest cost, proves also to be the structure with the most robust solution. The variability of lead-time and cost are smallest for the next generation supply chain design compared with the other two.

The simulation studies of the second case company focuses on performance measures such as quality (in terms of yield), lead times and costs. Costs include costs for work in process levels, rework time, and scrap. The study shows how these performance measures are interlinked with each other. The studies show that the performance measures improve with improved quality and improved supply chain designs. The model capturing the relationships among cost, quality and lead-time, indicate that cost increases more than linearly with lead-time. Also, the non-linearity increases with worse quality. Consequently, bad quality in long lead-time supply

chains is devastating to supply chain performance. Inversely, better quality and shorter lead-times in integrated and synchronised supply chains will lead to superior performance, see Figure 2.

The screening used in the second case study shows that demand, yield and transportation time are the most important input variables in the simulation models. In the context of supply chain management this means that product quality, in terms of yield, and supply chain design, in terms of transportation times, are important parameters in the studied supply chain designs. Demand is the single most important input variable in all the investigated models and is, of course, crucial to any supply chain. The results of the experiments, designed to find two-way interactions and quadratic effects, shows that two-way interactions exists between factors that can be controlled by the case company and factors describing the surrounding environmental disturbances. Disturbances such as low yield levels affects both the overall supply chain cost and the lead-time in this particular supply chain setting. Decreasing the transportation time in the supply chain can, however, counteract these disturbances. The optimal setting found through the experiments, suggest that the shortest supply chain design (the next generation supply chain) is the one with lowest cost, see Figure 3. The shortest supply chain will also provide the most robust solution since both the standard error and average of the weekly costs are minimised.

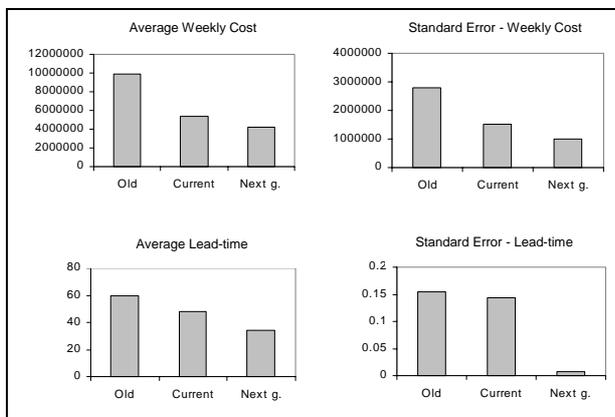


Figure 3: Optimal solutions for the three supply chain designs

## CONCLUSION

The two reported case studies highlights the usability of simulation for research related to supply chain management. The first case study reports both on the implications of a transition of the company to become a supply unit in a supply chain and on methodological impacts of simulation for supply chain related research questions. The second case study further expands the methodological development in screening and simulation output analysis. Together, these two case studies

provides a balanced picture of both insights in supply chain management and in simulation methodology.

## REFERENCES

- Banks, J., Buckley, S., Jain, S., Lendermann, P., and Manivannan, M. 2002. "Panel session: Opportunities for simulation in supply chain management", *Proceedings from the 2002 Winter Simulation Conference*, pp. 1652-1658.
- Bekker, J. and Saayman, S. 1999. "Drawing conclusions from deterministic logistic simulation models", *Logistics Information Management*, Vol. 12, No. 6, pp. 460-466.
- Bhaskaran, S. 1998., "Simulation analysis of a manufacturing supply chain", *Decision Sciences*, Vol. 29, No. 3, pp. 633-657.
- Chwif, L., Barretto, M.R.P., and Saliby, E. 2002. "Supply chain analysis: Spreadsheet or simulation", *Proceedings from the 2002 Winter Simulation Conference*, pp. 59-66.
- Forrester, J. W. 1961. *Industrial Dynamics*, Cambridge: MIT Press.
- Goetschalckx, M., Vidal, C .J. and Dogan, K. 2002. "Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms", *European Journal of Operational Research*, Vol. 143, Issue 1, pp. 1-18.
- Hayes, R. H. and Wheelwright, S. C. 1984. *Restoring Our Competitive Edge - Competing Through Manufacturing*, Wiley, New York.
- Hill, T. 1995. *Manufacturing Strategy - Text and Cases*, MacMillan, London.
- Jain, S., Workman, R. W., Collins, L .M., and Ervin, E. C. 2001. "Development of a high-level supply chain simulation model", *Proceedings of the 2001 Winter Simulation Conference*, pp. 1129-1137.
- Kleijnen, J. P. C., Bettonvil, B., and Persson, J. F. 2003., *Robust Solutions for Supply Chain Management: Simulation and Optimisation*, Working paper WP-303, Department of Production Economics, Linköping Institute of Technology, Linköping, Sweden.
- Manivannan, M. S. 1998. "Simulation of logistics and transportation systems", in Banks, J. ed.. *Handbook of Simulation*, John Wiley & Sons, New York, pp. 571-604
- Olhager J., Persson, J. F., Parborg, B. and Rosén, S. 2002. "Supply chain impacts at Ericsson: From production units to demand-driven supply units", *International Journal of Technology Management*, Vol. 23, Nos. 1/2/3, pp. 40-59.
- Persson, J. F. 2002. "The impact of different levels of detail in manufacturing systems simulation models", *Robotics and Computer Integrated Manufacturing*, Vol. 18, Issues 3-4, pp. 319-325.
- Persson, J. F. 2003. *Simulation Analysis of Supply Routes*, Working paper WP-301, Department of Production Economics, Linköping Institute of Technology, Linköping, Sweden.
- Persson, J. F. and Olhager, J. 2002. "Performance simulation of supply chain designs", *International Journal of Production Economics*, Vol. 77, No. 3, pp. 231-245.
- Shapiro, J. F. 2001. *Modelling the Supply Chain*, Duxbury, Pacific Grove.
- Towill, D. R. 1996. "Industrial dynamics modelling of supply chains", *International Journal of Physical Distribution & Logistics Management*, Vol. 26, No. 2, pp. 23-42.
- Zeigler, B. P., Praehofer, H., and Kim, T. G. 2000. *Theory of Modelling and Simulation*, 2<sup>nd</sup> Ed., Academic Press, San Diego.

## **AUTHOR BIOGRAPH**

**FREDRIK PERSSON** is an Assistant Professor in the department of Production Economics at Linköping Institute of Technology, Sweden. His research interests include modelling and simulation of manufacturing systems and supply chains. Of special interest are simulation methodology and validation methods.

He is a member of the Society for Computer Simulation International (SCS) and the Swedish Production and Inventory Management Society (SWEPIMS, also known as PLAN).

# A STUDY OF CONTROL VIA ON-LINE SIMULATION USING STOCHASTIC PETRI NETS

Matthias Becker

Thomas Bessey

Helena Szczerbicka

Institute of Systems Engineering, University of Hannover

Welfengarten 1, 30167 Hannover, Germany

{xmb,tby,hsz}@sim.uni-hannover.de

## KEYWORDS

On-Line Control, Decision Making, On-Line Simulation, Transient Dynamics

## ABSTRACT

Complex systems such as flexible manufacturing systems and traffic systems typically evolve with alternating periods of transient and nearly steady-state behavior; such systems often show suboptimal performance. Thus, it is desirable to optimize the system's performance on-line by adjusting the system's parameters properly before a performance drop is to occur. To this end, the system's future evolution is assessed in advance repeatedly by means of on-line simulation. However, there are several problems accompanying this approach, particularly the demand of real-time decisions, that have not been sufficiently solved yet.

Aiming at studying the dynamics of on-line control as well as its impact on the system's operation, we built a stochastic Petri net model that simulates on-line control of a simple open queueing network as it performs by means of on-line simulation. The system under control is easy to study since it has known properties and can be considered as part of a manufacturing system; jobs arriving at the system have to be dispatched to one of two machines, each providing a queue for jobs waiting to be processed. The processing times of the machines are deterministic or stochastic, while the jobs' arrival times are stochastic. With on-line simulation, the system's future performance is assessed by virtually dispatching a new job to either of the machines, based on the system's current state; the results are compared and thus lead to the real decision concerning to what machine the new job should be dispatched in order to minimize the work in progress.

In this work, we compare the quality of on-line control with that of other policies such as random choice and join the shortest queue.

## INTRODUCTION

### On-Line Simulation

Complex systems such as flexible manufacturing systems and urban traffic systems are usually planned through simulation before their operation actually starts. However, such systems typically evolve with high dynamics, that is, there are alternating periods of transient and nearly steady-state behavior. This is partially due to short-term changes of the requirements of their environment. Additionally, unexpected events such as machine breakdowns or accidents blocking certain routes in the network for some considerable time are also responsible for periods of transient behavior due to congestions. Thus, such systems often show suboptimal performance.

In order to overcome this problem, the idea is to optimize the system's performance on-line by repeatedly adjusting the system's parameters properly. This is referred to as on-line control. In general, on-line control is either reactive or proactive. The former type is characterized by adjusting the system's parameters only after a considerable performance drop is observed.

Proactive on-line control tries to adjust the system's parameters before a performance drop is to occur, in order to avoid this drop. To this end, the system's future evolution is assessed in advance repeatedly. The instants of time at which the assessment of the further evolution and adjustment of the parameters are done are called decision points. The assessment is done as follows: First, the system's current state is copied to several identical system models. For each of these models, certain values of the system's parameters are set, according to some appropriate policies that alternatively could control the system. Once the initialization is done, the models are analyzed in order to assess the future evolution under each policy. As the system under control typically is complex, simulation is the only feasible analysis method. This method is referred to as on-line simulation. With the results, the policy that leads to the optimal future performance of the system under control is chosen

to be implemented next, that is, the system is controlled by the chosen policy until the next decision point. This process is referred to as decision making.

There are several problems encountered with this approach, such as setting of the decision points, repeated validation of the system model (the search space for the parameters may vary over time) and proper analysis of the simulation results [2]. As simulation runs consume much time, the number and the length of the simulation runs become crucial. Since the system under control continues to evolve while the next policy is sought by on-line simulation, further problems arise [2]. For a detailed discussion of on-line simulation and associated proactive on-line control, see [4].

For some applications of this approach for traffic systems and (flexible) manufacturing systems, see [7] and [12, 6, 10], respectively. However, these applications are by far not suitable for widely adoption to the real world; they merely employ classical off-line simulation techniques for on-line use. By now, no strict theoretical research has been done.

### **This Work**

This work is intended as being a first step towards a characterization of the dynamics of proactive on-line control as well as of its impact on operation of the system under control. We believe that such characterization is an important and yet open problem that has to be solved in order to be able to assess the impact of applying on-line simulation on the system's performance before actual application. With the results of this and future work, we hope that we will get hints towards theoretical aspects in the field of on-line simulation. For this purpose, we built a stochastic Petri net model that simulates on-line control of a simple open queueing network as it performs by means of on-line simulation. The system under control is easy to study, since it has known properties; it can be considered as part of a manufacturing system. The on-line control and its associated on-line simulation are modeled as a stochastic Petri net as well. Note that both the system under control and its on-line control are integrated into one single stochastic Petri net model.

Stochastic Petri nets are widely accepted and applied as a powerful and concise modeling formalism for modeling of concurrent systems [11, 5], enabling qualitative as well as performance analysis. Depending on the state space of the model, performance analysis may be conducted by means of analytical methods, while simulation of a stochastic Petri net is possible in any case. Since Petri nets have a concise graphical notation, we decided to use stochastic Petri nets for our work; the resulting model is considered to be much clearer than many lines of code

in any programming language.

Furthermore, we believe that Petri nets might be a powerful tool for application in the field of on-line simulation [1]; however, to our best knowledge, no such application has been reported in the literature yet.

The remainder of this paper is organized as follows: In the next section, the stochastic Petri net model is presented in detail. Following this, we discuss the results of experiments conducted with the model. Finally, we conclude with a review of this work.

## **THE PETRI NET MODEL**

The stochastic Petri net employed in this work is modeled and simulated using the tool TimeNET [8]. However, the nets depicted in this paper are redrawn using another tool for sake of readability, since TimeNET does not show marking dependent arc weights. For the use of marking dependent arc weights, see the subsection on on-line control.

### **The System under Control**

The system under control consists of two queueing systems, that is, two queues Q1 and Q2 with a single server each. The two servers denote machines M1 and M2 with service rates  $\mu_1$  and  $\mu_2$ ; the firing times of the associated transitions may be deterministic or exponentially distributed. As soon as a token arrives to the system, as modeled by the transition arrival with exponentially distributed firing times and arrival rate  $\lambda$ , it will be dispatched to one of the two queueing systems, where it will be processed. The dispatching is accomplished by a policy such as random choice or JSQ (Join the Shortest Queue). Once the token is dispatched to one queueing system, no further jockeying is allowed.

### **Random Choice**

A stochastic Petri net which models the system under control as it implements random choice is shown in figure 1.

The arriving tokens are dispatched to either server with equal probabilities due to equal weights associated with the transitions choice1 and choice2.

### **JSQ**

A stochastic Petri net which models the system under control as it implements JSQ is shown in figure 2.

The places count1 and count2 denote the difference of the queue lengths regarding Q1 and Q2; they influence the transitions choice1 and choice2 by inhibitor

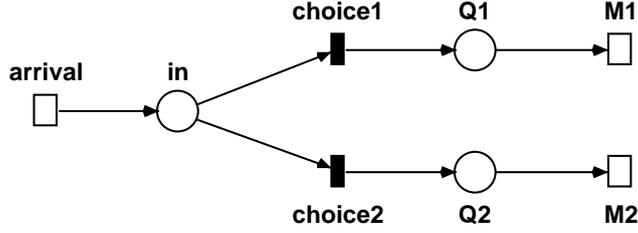


Figure 1: The System under Control Implementing Random Choice

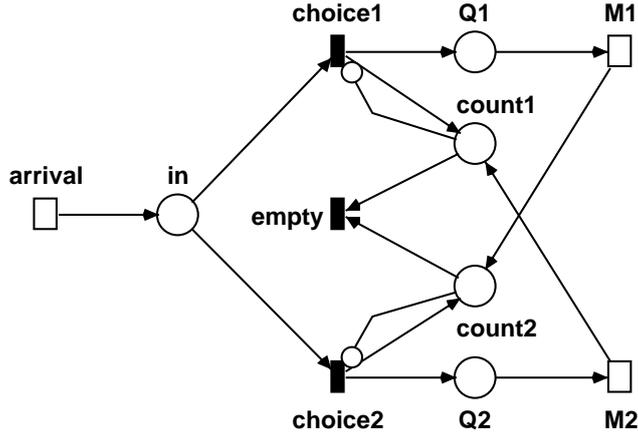


Figure 2: The System under Control Implementing JSQ

arcs according to JSQ. Since we are not interested in the queue lengths but only in their difference, we consider a job leaving server M1 by adding one token to place count2 instead of removing one token from place count1; note that the latter approach fails in the presence of the transition empty, while the former approach leads to a concise model of JSQ.

### On-Line Control

A stochastic Petri net which models the system under control as it is controlled by means of on-line simulation is shown in figure 3.

The stochastic Petri net works as follows:

Each newly arriving token triggers the on-line control; i.e., the token is not dispatched before the appropriate choice has been made by on-line control. To this end, the current system state as denoted by the queue lengths  $(\#Q1, \#Q2)$  is copied to the two alternative systems as initial states  $(\#S1Q1, \#S1Q2)$  and  $(\#S2Q1, \#S2Q2)$  for the on-line simulation, respectively; the copying is accomplished using marking dependent arc weights according to the transitions copyQ1 and copyQ2. Note that the alternative systems are reset to empty queues before copying,

since former on-line simulation may have left the systems dirty.

Following this, the on-line simulation is started by firing of transition start, which adds a new token to the places S1Q1 and S2Q2, respectively, according to the two alternatives of dispatching to be evaluated. In terms of the introduction, the on-line simulation evaluates two policies, where the first one always chooses the machine M1 for dispatching arriving tokens, while the second one always chooses the machine M2. Note that further arrivals to the on-line simulated systems do not occur. The alternative of which simulation has completed first (in that the respective queues are empty) is considered to be the best choice. This consideration is due to the limited tractability of the stochastic Petri net formalism with respect to performance evaluation as it is integrated as a stochastic Petri net itself. Regarding the transitions choice1 and choice2, the stop condition of the on-line simulation is modeled with inhibitor arcs, while the concurrency of both transitions is considered by the place P4. The transitions transfer the token of which arrival triggered the on-line simulation to either queue Q1 or queue Q2.

The service rates of the transitions  $SxMx$  are greater

than that of the transitions M1 and M2 by a factor of  $s > 1$ , which is the speedup of the on-line simulation. Actually, instead of simulating on-line simulation (it would be a difficult task to do so since we would have to specify for each control decision its required "thinking time" in advance), we simply operate the system under control at a higher speed. Hence, the definition of speedup that we use in this work is quite different from the usual one, which states that speedup is the factor by which the on-line simulation is faster than real-time. But since both definitions are related by some kind of mapping, we expect our approach to be valid.

While the on-line simulation is being performed, arriving tokens according to the system under control have to wait in the place in to proceed until the choice for dispatching is made for the pending token that triggered the on-line control (as controlled by the place *ready*). Thus, the mean number of tokens in the place in can be used as a measure for the delay of the system's operation due to the execution time of the on-line simulation needed for each token.

Note that the stochastic Petri net model presented here is simplified for sake of comprehension in that just one single simulation run is performed with respect to the on-line simulation. In fact, for the experiments we conducted, we used an extended stochastic Petri net model that employs  $N$  simulation runs in order to make the choice for dispatching according to one token. For this purpose, the extended model comprises two additional places (one for each possible choice) that count the results of each simulation run concerning what choice is considered to be the best (just as described above); the place that contains  $\frac{N}{2} + 1$  tokens first causes the associated choice to actually be made.

## EXPERIMENTAL RESULTS

We use the mean work in progress (WIP) as the performance measure regarding the system under control; in all experiments, the WIP is computed by performing steady-state simulation with a confidence level of 95 percent. The simple system presented here reaches steady-state even under on-line control, so our experimental environment is justified. Since the mean WIP observed in the queues Q1 and Q2 is directly related to the mean waiting time through Little's Law [3], smaller values of the WIP imply better performance of the control policy.

In the following, we consider four cases regarding the processing time distributions of the two machines M1 and M2; these cases differ in the variance (i.e., deterministic vs. exponential processing times) and in the ratio of the mean processing times  $\mu_1^{-1}$  and  $\mu_2^{-1}$

(equality vs. inequality). Note that in case of deterministic processing times, only one simulation run is needed in order to perform the on-line simulation, thus  $N = 1$  in these cases.

### Balanced Deterministic Case

The results are shown in table 1, where  $\lambda = 1.75$  and  $\mu_1 = \mu_2 = 1.0$  (with the machines having constant processing times).

Note that in the case of balanced deterministic processing times, JSQ is the optimal policy with respect to the mean waiting time of the system under control, since for each machine, the waiting time can be computed by multiplying the number of waiting jobs in the queue with the constant processing time for each job. However, in the experiment, the WIP resulting from applying on-line control is somewhat smaller than that resulting from JSQ; this is due to the simulation error. Basically, the on-line control leads to optimal performance in this case, just as JSQ does. In fact, studying the simulation traces of the experiment, it turns out that the on-line control just behaves like JSQ. Note, however, that while JSQ considers the system's state explicitly in that it compares the two queue lengths, the on-line control just selects one of the two simplest policies possible, which do not consider the system's state explicitly. In the latter case, the information about the system's state is implicitly considered when initializing the on-line simulation to the current state of the system under control.

In addition, the experiment yields  $WIP(in) = 0.003$ , given that  $s = 100$ ; thus, in this case, the execution time of the on-line simulation has virtually no impact on the system's operation, compared to the WIP of Q1 and Q2.

### Balanced Exponential Case

Here, the machines have exponentially distributed processing times, where, again,  $\lambda = 1.75$  and  $\mu_1 = \mu_2 = 1.0$ . The results are shown in table 2 regarding the random choice and JSQ policies and in table 3 regarding the on-line control under different settings with respect to  $N$  and  $s$ .

It can be observed that generally, the results obtained by means of on-line control get better, the more simulation runs are performed regarding the on-line simulation. Quite clearly, this is because of the stochastic nature of the processing times associated with the machines. However, with  $N$  increasing, the time needed for performing the on-line simulation becomes crucial to the quality of the on-line control. At this point, the speedup of the on-line simulation becomes important, where larger speedup allows for performing more simulation runs according to a fixed

Table 1: Results for the Balanced Deterministic Case

	Random Choice	JSQ	On-Line Control
WIP(Q1)	3.88	2.42	2.35
WIP(Q2)	3.95	2.42	2.35
$\Sigma$	7.83	4.84	4.70

Table 2: Results for the Balanced Exponential Case (1)

	Random Choice	JSQ
WIP(Q1)	7.02	3.95
WIP(Q2)	6.99	3.98
WIP(in)	—	—
$\Sigma$	14.01	7.93

time period.

The results show that also in this case, on-line control is able to perform as optimal as JSQ does [9], although a considerable number of simulation runs concerning the on-line simulation is needed, requiring an appropriate speedup. However, the WIP according to the place in is not negligible anymore, implying that the on-line simulation has a serious impact on the system's operation in that it causes a considerable number of tokens to wait before processing.

Note that in case of  $N = 5$  and  $s = 100$ , the WIP regarding the place in is about two third of the WIP regarding Q1 and Q2 (with this ratio being considerably greater than in other cases); however, the total WIP is similar to the WIP resulting from applying JSQ. Thus, the on-line control performs quite well despite of its specific dynamics as measured by observing WIP(in). In addition, in case of  $N = 15$  and  $s = 1000$ , the total WIP is similar to the WIP resulting from JSQ, too, while the speedup is considerably larger.

However, large speedup may lead to several problems concerning statistical analysis of the simulation results [1]: While the execution time needed for on-line simulation is reduced as the speedup increases, concurrent validation of the employed simulation models may still be necessary; however, certain events that will possibly occur while the system under control continues to operate, causing the real-time data to be updated, may become rare with respect to the on-line simulation, while the data updates have to be considered by the validation process.

Thus, small speedup leads to the problem of incorrect decision making due to the system's evolution while the on-line simulation is being performed, while large speedup may lead to incorrect decision making

due to incorrect simulation results; quite clearly, this stresses the need for research on a trade-off regarding the speedup and the accuracy of on-line simulation.

### Unbalanced Deterministic Case

The results are shown in table 4, where  $\lambda = 1.75$  and  $\mu_1^{-1} = 1.5$ ,  $\mu_2^{-1} = 0.5$ . In this case of unequal mean processing times, we additionally give results for the throughputs  $\tau$  of the machines M1 and M2, since they imply the ratio of the jobs' splitting between the machines.

Note that in the case of random choice, we adjusted the weights associated with the transitions choice1 and choice2 (see figure 1) in that arriving jobs are now dispatched to the slower machine M1 with a smaller probability and vice versa. With this experiment's settings, the probability of choosing the first machine is 0.25, whereas that of choosing the second one is 0.75, corresponding to the reciprocal of the ratio of the mean processing times. (Note that while this is an obvious adjustment, it is not optimal with respect to the sum of the mean waiting times.) As result, the throughputs' ratio is that of the dispatching probabilities, while the WIP of Q1 and Q2 is evenly split.

When comparing the results regarding JSQ to those regarding the on-line control, the following observations can be made: First, the ratio of the throughputs in case of on-line control is greater than that in case of JSQ by a factor of four; second, the main portion of the WIP in case of JSQ is related to Q1, while it is to Q2 in case of on-line control. This is due to the fact that JSQ only relies on the queue lengths; however, longer queues in front of faster servers may be processed in less time than shorter queues in front of slower servers, depending on the ratio of the mean serving times as compared to the ratio of the queue

Table 3: Results for the Balanced Exponential Case (2)

On-Line Control	$N = 1$ $s = 100$	$N = 5$ $s = 100$	$N = 15$ $s = 100$	$N = 15$ $s = 1000$
WIP(Q1)	4.99	3.20	1.90	3.96
WIP(Q2)	4.98	3.00	1.90	3.73
WIP(in)	0.12	2.15	18.00	0.26
$\Sigma$	10.09	8.35	21.80	7.95

Table 4: Results for the Unbalanced Deterministic Case

	Random Choice	JSQ	On-Line Control
WIP(Q1)	1.25	1.31	0.29
WIP(Q2)	1.26	0.84	1.62
$\Sigma$	2.51	2.15	1.91
$\tau$ (M1)	0.43	0.54	0.18
$\tau$ (M2)	1.30	1.18	1.56

lengths. While JSQ cannot consider this fact, on-line control can thanks to the on-line simulation. This leads to the on-line control performing better than JSQ, thus being the optimal policy in this case.

### Unbalanced Exponential Case

Here, the machines have exponentially distributed processing times, where, again,  $\lambda = 1.75$  and  $\mu_1^{-1} = 1.5$ ,  $\mu_2^{-1} = 0.5$ . The results are shown in table 5 regarding the random choice and JSQ policies and in table 6 regarding the on-line control under different settings with respect to  $N$  and  $s$ . (Note, however, that these settings are slightly different from that of the balanced case.) Again, the dispatching probability regarding the machine M1 is three times smaller than that regarding the machine M2 in case of random choice.

The results show that in case of stochastic processing times, performing one single simulation run with respect to on-line simulation is senseless; in this experiment, the on-line control even causes the system under control to become unstable, which is due to the fact that the on-line control randomly overloads one of the two machines because of insufficient future projection accuracy. (In this case, the steady-state simulation that we performed is senseless.) With several simulation runs, however, the on-line control performs nearly as good as JSQ. Note that also in this case, the WIP of the place in becomes significant as the number of simulation runs increases, given a fixed speedup; in fact, the WIP regarding in is about one fifth of the total WIP, although the speedup is  $s = 1000$ .

### CONCLUSION

In this work, we built a stochastic Petri net that models on-line control as it performs by means of on-line simulation; for sake of conciseness, both the system under control and its on-line control were integrated into one single stochastic Petri net model. As the system under control, we employed a simple open queueing network consisting of one arrival process and two service processes, where the dispatching of arriving jobs to one of the two servers is subject to one of the policies random choice, JSQ and on-line control. Since our aim was to study basic aspects of on-line control, we confined ourselves to a simple system under control with known properties. We conducted several experiments with this model, varying the processing time distributions associated with the two machines. With these experiments, we studied the impact of the speedup and the number of simulation runs employed in the on-line simulation on the performance of the on-line control.

It turned out that JSQ always outperforms random choice (as expected), while the on-line control performs quite as good as JSQ. In case of unbalanced deterministic processing times, the on-line control even turned out to be optimal, despite of the simple evaluation method of emptying the queues. This is considered as a strong hint towards the profitable applicability of on-line simulation to manufacturing systems, although the system under control that we used here is indeed very simple.

Despite of this simplicity, our experiments lead to an important observation that may turn out to reveal a major issue in the application of on-line simulation:

Table 5: Results for the Unbalanced Exponential Case (1)

	Random Choice	JSQ
WIP(Q1)	1.91	1.79
WIP(Q2)	1.92	1.16
WIP(in)	—	—
$\Sigma$	3.83	2.95

Table 6: Results for the Unbalanced Exponential Case (2)

On-Line Control	$N = 1$ $s = 100$	$N = 5$ $s = 100$	$N = 5$ $s = 1000$	$N = 15$ $s = 1000$
WIP(Q1)	(unstable)	1.01	1.19	0.56
WIP(Q2)	(unstable)	1.65	1.79	2.04
WIP(in)	(unstable)	0.32	0.03	0.64
$\Sigma$	—	2.98	3.01	3.24

While on-line simulation generally leads to higher performance of the on-line control as it is performed with larger speedup (since the system's ongoing evolution becomes negligible), it may still perform quite well with smaller speedup (or similarly, with more simulation runs). Facing the risk of statistical instability in case of large speedup due to rare events of the system under control, one may benefit from small speedup compared to fast on-line simulation. However, small speedup (or similarly, a large number of simulation runs) may affect the system's operation; in our experiments, the WIP related to the incoming place considerably increased. Thus, the experimental results give hints towards the need for a trade-off for on-line simulation regarding its execution time and its accuracy, where great care has to be taken in case of large speedup with respect to statistical stability.

Furthermore, the results give hints that it is not suitable to trigger on-line control for every single job, since the impact on the system's operation may become undesirably great; instead, the on-line simulation should cover the short-term future of the system's evolution in that it includes arrival processes as well. In this approach, the policies subject to decision making would be assessed regarding their impact on the system's performance every time a certain decision point is reached, as discussed in the introduction.

## REFERENCES

- [1] T. Bessey. Needs and proposals for theoretical research on on-line simulation. In *Proc. Summer Computer Simulation Conference (SCSC)*, 2003.
- [2] T. Bessey. On-line simulation: Towards new statistical approaches. In *Proc. Summer Computer Simulation Conference (SCSC)*, 2003.
- [3] G. Bolch, S. Greiner, H. d. Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains*. Wiley, 1998.
- [4] W. J. Davis. On-line simulation: Need and evolving research requirements. In J. Banks, editor, *Handbook of simulation*, chapter 13. Wiley, New York, 1998.
- [5] A. A. Desrochers and R. Y. Al-Jaar. *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*. IEEE Press, 1995.
- [6] G. R. Drake and J. S. Smith. Simulation system for real-time planning, scheduling, and control. In *Proc. Winter Simulation Conference (WSC)*, 1996.
- [7] J. Esser, L. Neubert, J. Wahle, and M. Schreckenberg. Microscopic online simulation of urban traffic. In *Proc. 14th International Symposium on Transportation and Traffic Theory*, 1999.
- [8] R. German, C. Kelling, A. Zimmermann, and G. Hommel. TimeNET — A toolkit for evaluating non-Markovian stochastic Petri nets. *Performance Evaluation*, 24:69–87, 1995.
- [9] H.-C. Lin and C. S. Raghavendra. An approximate analysis of the Join the Shortest Queue (JSQ) policy. *IEEE Transactions on Parallel and Distributed Systems*, 7(3), 1996.

- [10] S. Manivannan and J. Banks. Real-time control of a manufacturing cell using knowledge-based simulation. In *Proc. Winter Simulation Conference (WSC)*, 1991.
- [11] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs, 1981.
- [12] A. I. Sivakumar. Optimization of cycle time & utilization in semiconductor test manufacturing using simulation based, on-line near-real-time scheduling system. In *Proc. Winter Simulation Conference (WSC)*, 1999.

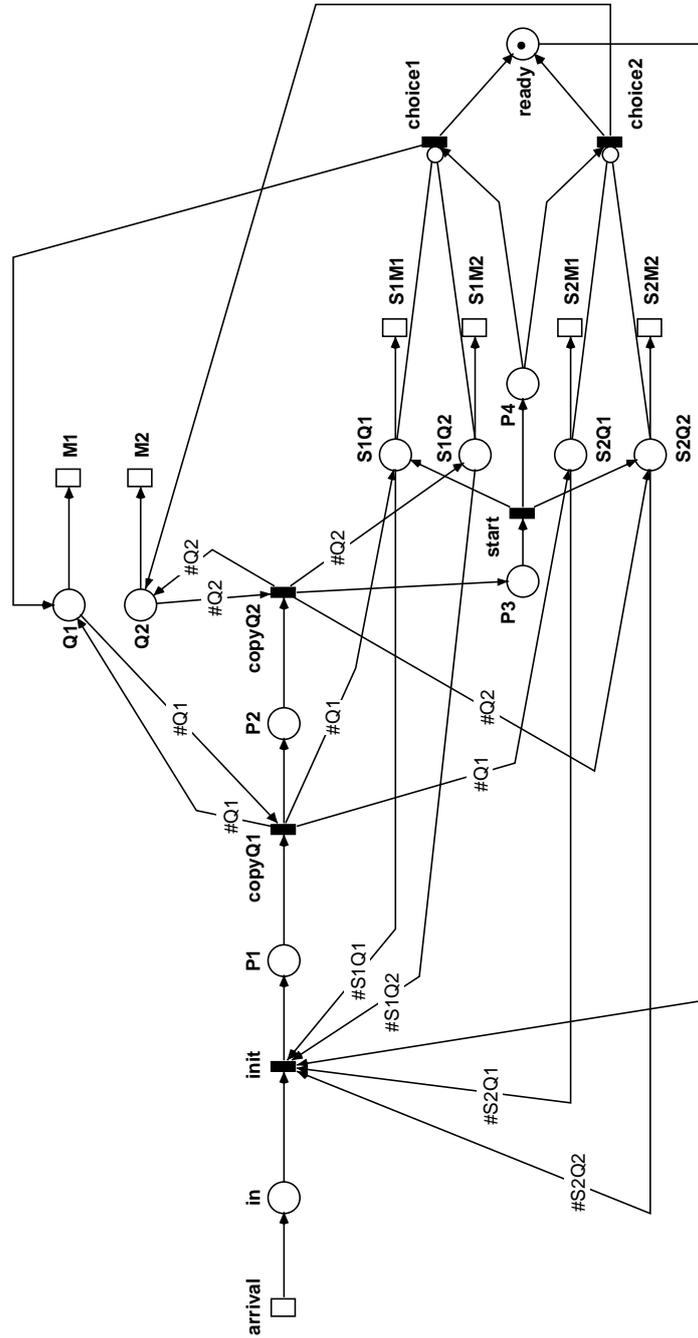


Figure 3: Control via On-Line Simulation

# MATHEMATICAL MODELLING AND IDENTIFICATION OF THE FLOW DYNAMICS IN MOLTEN GLASS FURNACES

Jan Studzinski  
Systems Research Institute of Polish Academy of Sciences  
Newelska 6  
01-447 Warsaw, Poland  
E-mail: studzins@ibspan.waw.pl

## KEYWORDS

Navier-Stokes equations, mathematical modeling, computer simulation, dynamic systems identification.

## ABSTRACT

In the paper a new method for computer aided modelling and identification of flow dynamics in molten glass is presented and numerically analysed. The process of the construction of the model occurs in several steps on which the sub-models with differentiated mathematical descriptions (with distributed or lumped differential equations) and dynamical features (with inertial and oscillatory characteristics and with slow and fast changeable dynamics) are setting up. This method makes possible to prepare the models of glass tank furnaces of high degree of accuracy, described with the equations of high orders. The models are suited well to estimate technological parameters of glass tank furnaces and to control the glass melting process.

## INTRODUCTION

The glass production is a very complex technological process. For this reason there is very difficult to set up its mathematical models that could be useful for practical applications, such as computer simulation, control or estimation of technological parameters. The modelling of glass tank furnaces occurs usually under separated application of Distributed or Lumped Parameter Equations (DPE or LPE models) and the result is that they are very complicated (DPE models) or very simplified (LPE models). That is why their practical usefulness is very limited. This situation is caused by lacking of adequate identification methods. Thus in the following a numerical algorithm is presented for setting up molten glass models under consideration of both arts of mathematical description. This way their drawbacks could be eliminated and their advantages retained. The algorithm presented consists of two general stages. On the first stage a DPE model is formulated with the quasi-linear Navier-Stokes and energy equations and with an equation added that describes the glass mass composition change in the molten glass. On the second stage a complex LPE model is prepared using the DPE model previously identified. All computations are done using real data from a industrial glass tank furnace. The fitting of the DPE model to the data occurs by using static

optimisation methods. To estimate the structure and the parameters of the LPE model an indirect identification method is used, developed especially for setting up continuous dynamic models of higher orders.

## DPE MODEL FORMULATION

To model the glass mass flow in a tank furnace by the partial differential equations the following description is used (Studzinski 2002a):

$$\begin{cases} \mu(T)(D_1^2 v_1 + D_2^2 v_1) = D_1 p \\ \mu(T)(D_1^2 v_2 + D_2^2 v_2) = D_2 p - \rho g \beta (T - T_o) \end{cases} \quad (1)$$

$$\lambda(T)(D_1^2 T) + \lambda(T)(D_2^2 T) = \rho c_v (v_1 D_1 T + v_2 D_2 T) \quad (2)$$

$$D_1 v_1 + D_2 v_2 = 0 \quad (3)$$

$$\frac{\partial z}{\partial t} + e_1 v_1 \frac{\partial z}{\partial x_1} + e_2 v_2 \frac{\partial z}{\partial x_2} = D(T) \left( e_3 \frac{\partial^2 z}{\partial x_1^2} + e_4 \frac{\partial^2 z}{\partial x_2^2} \right) \quad (4)$$

where the parameters mean:  $v_1, v_2$  – longitudinal and vertical glass mass velocities in  $x_1$  and  $x_2$  directions,  $p$  – pressure,  $T$  – temperature and  $T_o$  – reference temperature,  $z$  – chemical composition of the melt,  $t$  – time,  $\mu$  – dynamic viscosity,  $\rho$  – density,  $g$  – gravitational acceleration,  $\beta$  – thermal expansion,  $\lambda$  – thermal conductivity,  $c_v$  – specific capacity,  $D$  – diffusion coefficient,  $e_1$  to  $e_4$  – some fitting coefficients (to fit the model to an object).

Equations (1), (2), (3) are known in the classical fluid mechanics as the Navier-Stokes (or motion), energy and continuity equations, respectively, and they are formulated on the base of the momentum, energy and mass conservation laws. These equations describe the distributions of the temperature and the glass melt velocities in a tank furnace induced by the free and forced convections currents in the molten glass. Equation (4) describes the glass mass composition changes induced by the convection currents and the diffusion. While setting up the equations several simplified assumptions were made that took into consideration the specific properties of the glass mass flow and also the hypothesis that the glass melt is an incompressible and Newtonian liquid (Studzinski

2002a). The scheme of a glass tank furnace modelled and the main convection currents occurring in the molten glass are shown in Fig. 1.

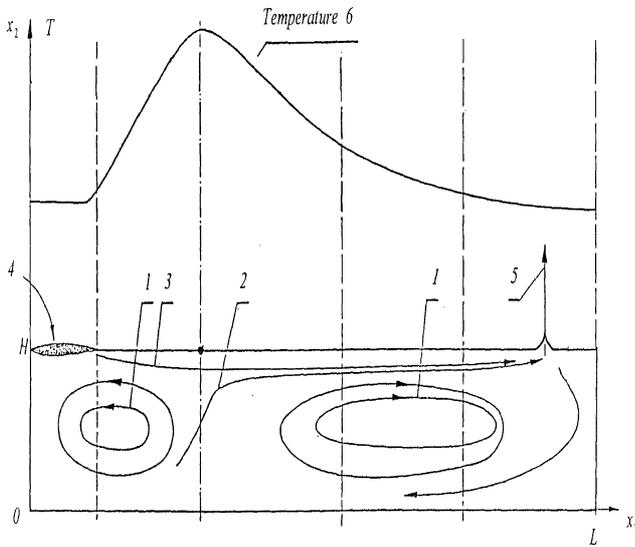


Figure 1: Longitudinal section of the glass tank furnace and the main currents occurring in the melt; 1,2,3 - rotating, withdrawal and surface current, respectively, 4 - raw materials input, 5 - glass take-out, 6 - temperature distribution on the free surface of the glass melt.

### DPE MODEL IDENTIFICATION

Equations (1-3) make together a two-dimensional DPE model of the glass mass flow in a tank furnace. After some boundary conditions are given and the temperature and velocities values are calculated from equations (1-3), one can calculate subsequently the glass melt composition at each point of the tank by solving equation (4). To get the numerical solution of the model equations the finite difference method is used and a theoretical analysis of the numerical solvability of the model is made (Studzinski 2002a). On the first step of the model computing equations (1-3) are solved. The boundary conditions for the function  $p$  are unknown and this makes necessary to transform the equations. It is done by replacing the velocities  $v_1, v_2$  by the current function  $\psi$  what results in a new model form consisting of only two equations contrary to the four ones in (1-3). The reduction of the number of equations causes in general a better convergence when solving the model numerically. A discrete approximation of the model equations occurs by the help of difference quotients. The use of standard difference quotients leads, however, in the case of high order derivations of equations to a bad stability of the resulted difference schemes at the edges of the knotted grid. To improve the approximation some new central difference quotients have been developed for the high order derivations of  $\psi$ . The difference schemes resulted from equations (1-3) are solved by means of the

relaxation method using an iterative algorithm. For the numerical calculation the values of the physical coefficients and of the space dimensions of the model were chosen according to those ones of a real tank furnace. The convergence of the iterative algorithm was relatively fast with highly satisfactory accuracy of the calculation. Some results of the temperature and current fields computed are shown in Fig. 2. One can see in Fig. 2 that only the rotating and withdrawal currents but not the surface current (as it is shown in Fig. 1) are determined after the simulation of model (1-3) was made. This current could not be obtained with a two-dimensional DPE-model.

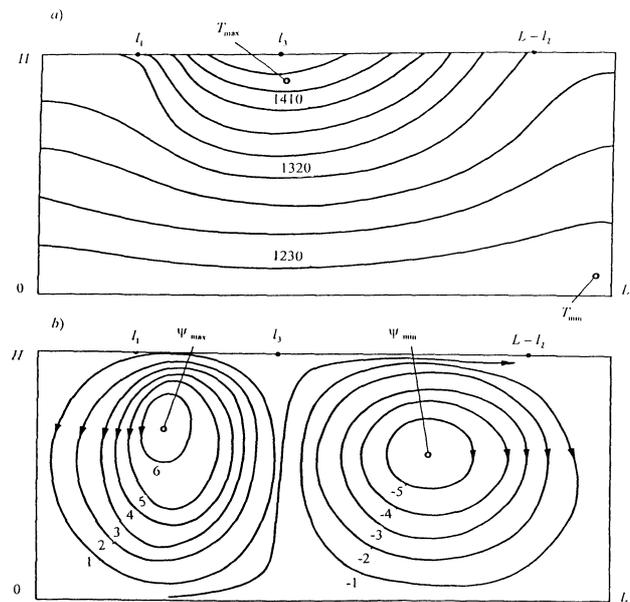


Figure 2: Computed temperature (figure a) and current distribution (figure b) in the glass melt for the longitudinal section of the glass tank furnace.

The numerical solution of equation (4) occurs on the second step of the modelling. To approximate (4) some central difference quotients of the finite difference method are used and as a result a new difference scheme with some fitting coefficients is obtained. The glass tank model described by equations (1-4) constitutes an approximation of a real object. Such an approximation is usually not exact although the parameters and dimensions of the model correspond to those ones of the tank. The possible inaccuracies occur while the model equations and boundary conditions are formulated and the parameter values are determined. Also the numerous simplifications made during the setting up the model are responsible for many inaccuracies and this is practically unavoidable. Then the fitting of the model to the object can be realised by the help of equation (4) and some measurements data obtained from the tank furnace under investigation (see Fig. 3). To do it the following identification problem is formulated:

$$\min_{e_i} Q(e_i) = \min_{e_i} \sum_{k=0}^K (z^k - z^k)^2 \quad (5)$$

where  $z^k$  and  $z^k$  mean the measured data and the discrete values of the model output that is calculated by solving equation (4) (the glass composition  $z$  is here considered as the radioactivity of the glass melt that has been measured while realising an isotope experiment on the tank furnace). To solve problem (5) a static non-gradient optimisation method is used (Studzinski 2002b). The criterion function  $Q(e_i)$  is strong non-linear relating to  $e_i$ . Because of that the start points for the optimisation runs had to be chosen very carefully and close enough to the optimum. The model output obtained from the calculation is shown in Fig. 3. One can see that the output fits well to the data in the farther section of the curve where the influence of the rotating and withdrawal currents on the glass mass flow is the strongest. The approximation of the data with the model output in the initial section of the curve is much worse but there the surface current determines the data which is noticeable through the high oscillations of the curve. This situation can be explained through the omission of the surface current in the DPE model. This current could be considered in a three-dimensional DPE model but unfortunately such a model would be hardly possible to identify because of its great complexity.

## LPE MODEL AND ITS IDENTIFICATION

The glass mass flow in a tank furnace can be described using also LPE models. Their parameters have no physical meaning and this gives interpretation troubles when comparing the models and objects. On the other side the setting up of such the models is easier than PDE models regarding the work complexity and the computing time needed for simulation and identification. Usually the non-linear regression methods are used for developing the lumped parameter models. These methods are generally successful if models of lower orders have to be set up but they are not effective in more complicated cases. The main problems then are connected with the choice of an adequate model structure and with the fixing a start point possibly closely to the optimum while making the identification. The methods of non-linear regression converge usually to the local optimal points if the start points are not right.

To overcome these problems an indirect identification method was developed to model linear dynamic objects of higher orders from their sampled impulse responses (Nahorski et al. 1985). This method has been adopted for setting up the LPE models of glass tank furnaces by using a multistage modelling approach (Studzinski 2002b). The mathematical description of an object modelled is now in the form of the homogeneous ordinary differential equation:

$$\frac{d^R z}{dt^R} + a_{R-1} \frac{d^{R-1} z}{dt^{R-1}} + \dots + a_0 z = 0 \quad (6)$$

with the non-zero initial conditions added:

$$\begin{cases} z(0) = b_{R-1} \\ z^{(1)}(0) = b_{R-2} - a_{R-1}z(0) \\ \dots \\ z^{(R-1)}(0) = b_0 - a_1z(0) - \dots - a_{R-1}z^{(R-2)} \end{cases} \quad (7)$$

and with the following analytical solution function:

$$z(t) = \sum_{j=1}^J \sum_{l=0}^{m_j-1} t^l \exp(\alpha_j t) (c_{jl} \cos(\varphi_j t) + d_{jl} \sin(\varphi_j t))$$

$$\sum_{j=1}^J m_j = R \text{ and } m_j > 0 \quad (8)$$

The continuous equation (6) can be approximated by the following discrete equation:

$$z_k + s_{R-1}z_{k-1} + \dots + s_0z_{k-R} = 0 \quad (9)$$

with:  $z_k = z(k\Delta)$ ,  $k=1,2,\dots,K$ ,  $\Delta t$  – sampling step, and with the following analytical solution function:

$$z_k = \sum_{j=1}^J \sum_{l=0}^{m_j-1} k^l \sigma_j^k (f_{jl} \cos(\psi_j k) + g_{jl} \sin(\psi_j k)) \quad (10)$$

By comparison (10) and (8) one can convert very easy the coefficients of function (10) into the coefficients of (8).

The numerical algorithm realising the indirect identification method is as follows:

1. Fitting the difference equation (9) to the impulse response obtained from the object, using a standard time series identification method.
2. Estimation of the coefficients in the time discrete function (10) using a standard optimisation method (e.g. the linear regression) and the parameters identified in (9).
3. Calculation of the coefficients in the time continuous function (8) converting the coefficients of function (10) with the help of some simple algebraic formulas.
4. Calculation of the parameters of equations (6) with the help of the parameters of (8).

The main idea of the indirect identification method is that at first a discrete model is found and afterwards it is converted into the time continuous one. In this way the search for a continuous model is realised „indirectly”, i.e. using a discrete model that is much easier to develop from the numerical point of view. In the case of

complex objects it is well-advised to divide the modelling process into several stages at which sub-models with different dynamics features are constructed and afterwards put together to one overall model. On each stage of modelling different data sequences must be used for identification and they are to be isolated from the original measurements. The currents distribution occurring in the glass melt (see Fig. 1) suggests that the features of the melt mixing dynamics in a tank furnace depend in a different way on the character and velocities of the currents. The slow-running withdrawal current decides on the dynamics of the slow-varying inertial character and the fast-running surface current, as well as the rotating currents decide on the dynamics of the different-varying oscillatory characters. Also the isotope data for identification display both the inertial and oscillatory characters (see Fig. 3). The above remarks justify the application of the multistage approach for modelling glass tank furnaces. The choice of the best („optimal”) sub-models as well as of the best overall model occurs by means of the residual sums.

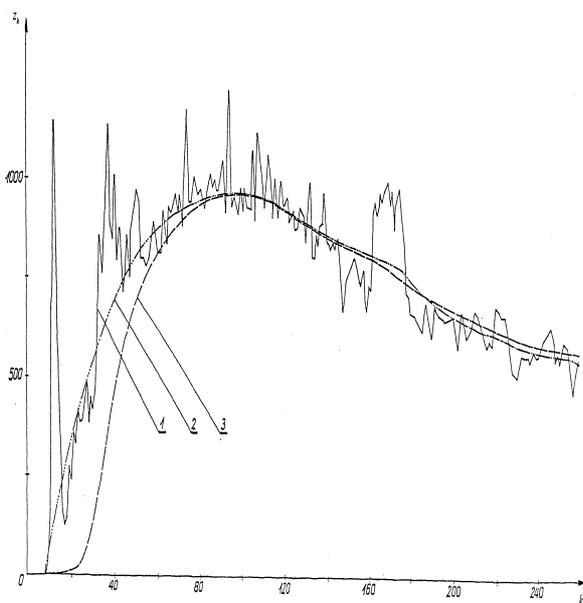


Figure 3: Isotope data for modelling the glass mass flow dynamics; 1,2- noisy and smoothed data, respectively, 3 - output of the DPE model.

Some models have been developed for the glass tank furnace under consideration using this multistage approach. They fit well to the farther part of the data curve (where the „slow” dynamics of the object dominate) but their adaptation to the initial phase of the curve (where the oscillatory components dominate) is much worse. The modelling of this initial data section depends considerably on the division of the whole data sequence into the components which are used for setting up the sub-models. This makes the main trouble when

using the multistage modelling approach with the LPE description of the models. Since the runs of the data components are not known from the beginning, they can be guessed only in general and the appropriate data curves are obtained using various smoothing algorithms. This leads, however, to great inaccuracies of the proceeding.

### COMBINED ALGORITHM OF MOLTEN GLASS MODELING

To avoid the disadvantages of the above modelling approaches a combined algorithm for modelling glass tank furnaces has been developed. The final models obtained by means of this approach are described by ordinary differential equations but a DPE model is used at the first level of the modelling. The conception of this combining modelling resulted from the experience which was gathered after the models with distributed and lumped parameters were developed separately. In the latter case the modelling of the oscillations appearing in the isotope data (and caused by the surface current) is not exact. The only use of smoothing algorithms does not allow to determine exactly the initial run of the data curve which is used later to set up the „slow” inertial sub-model and because of that there is not possible to get the right data for farther stages of the modelling. But these difficulties can be surmounted by help of the DPE model. It allows to isolate correctly the surface current component of the data from the component which is responsible in the main for the glass mass transport in a tank furnace. This component is caused by the withdrawal and rotating currents and it is approximated correctly by the DPE model.

The two-level modelling approach is as follows:

1. Formulation of the partial differential equations describing the DPE model and its computer simulation.
2. Identification of the DPE model with the help of the measurements data and by means of an optimisation method.
3. Developing of the slow-varying LPE sub-model using the output of the DPE model as the data for the indirect identification method.
4. Preparation of the data for setting up the fast-varying LPE sub-model by subtracting the output of the DPE model from the original measurements and by smoothing the results (this sub-model will describe the contribution of the surface current in the measurement data).
5. Developing of the fast-varying LPE sub-model using the indirect identification method.
6. Putting together the sub-models into one overall model and the subsequent estimation of its parameters by means of the non-linear regression methods.

After using the combined modelling algorithm a complex LPE model of the glass mass flow dynamics was finally set up. The model has the eleventh order and

it consists of two sub-models of sixth and fifth orders, respectively. The 6th order sub-model has the inertial-oscillatory character and owns two real and four complex roots in its transfer function. It fits very well to the output of the PDE model. The 5th order sub-model has either the inertial-oscillatory character and it has one real and four complex roots in its transfer function. It fits very well to the oscillations caused by the surface current. The overall LPE model fits well to the original measurements and it approximates exactly the oscillations occurring in the initial section of the data (see Fig. 4).

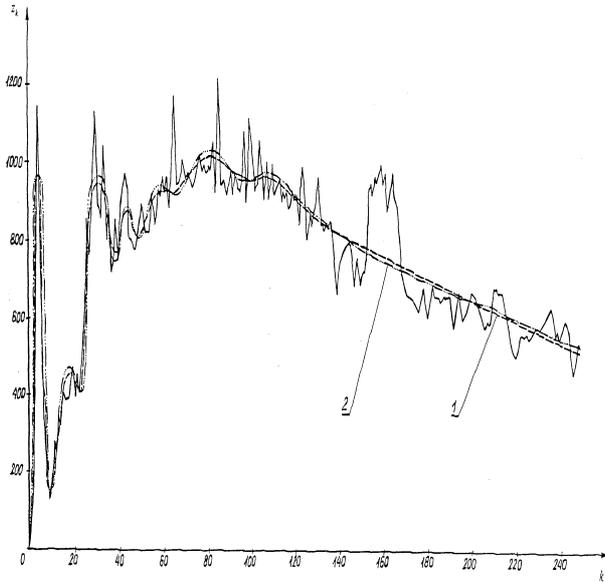


Figure 4: Overall LPE models obtained by means of the combined modelling approach without and after using the non-linear regression method (curve 1 and 2 respectively).

## CONCLUSIONS

The problem of mathematical modelling of the glass mass flow dynamics in a glass tank furnace is solved and three numerical approaches of modelling are presented, tested and discussed. The first approach develops two-dimensional DPE models that describe the slow-varying dynamics of the glass tanks in which the withdrawal and rotating currents occur and no surface current appears. The second approach allows the development of LPE models of relatively small order that do not describe exactly the complex dynamics of the objects in which all kinds of the currents occur. The troubles arise while modelling by means of this approach the initial section of the isotope data where the simultaneous effects of the slow and fast varying currents are particularly strong. There is no effective algorithm to divide the data curve into the components for there is not known *a priori* in which way the individual currents influence the measurements. The

third approach is a combination of the two and it makes possible to develop complex LPE models of the high order that have got the inertial-oscillator features and very differentiated parameter values.

The models computed by help of this approach describe very well the dynamics of the glass mass flow and all the same they are simple and convenient enough for numerical treatment. They can be used for the development of control or stabilisation algorithms with reference to the chemical composition of the glass as well as for the calculation of the technological parameters of glass tank furnaces.

## REFERENCES

- Nahorski Z, Bogdan L. and Studzinski J. 1985. "Estimation of system structure and parameters from noisy sampled impulse response". In: Proceedings of 7. IFAC/IFORS Symposium on Identification and System Parameter Estimation, York, 1747-1754.
- Studzinski J. 2002a. "Identification of the glass mass flow dynamics in glass tank furnaces". Report of XXIX Summer School on Advanced Problems in Mechanics APM'2001, Saint Petersburg, 525-542.
- Studzinski J. 2002b. "On the solution of a nonlinear Navier-Stokes problem using the finite difference method". Report of XXIX Summer School on Advanced Problems in Mechanics APM'2001, Saint Petersburg, 543-561.

# USING OPC DATA EXCHANGE IN SIMULATION ASSISTED AUTOMATION TESTING

Jyrki Peltoniemi, Matti Paljakka, Tommi Karhela  
VTT, Technical Research Centre of Finland  
P.O.Box 1301, FIN-02044 VTT  
email: Jyrki.Peltoniemi@vtt.fi

## KEYWORDS

Process simulation, automation testing, OPC data exchange, software design, performance

## ABSTRACT

Dynamic process simulation models can be used for testing automation - both control and logic - before the commissioning. This activity requires a flexible, fast and robust connection between automation software and process simulation engines.

OPC Data eXchange (DX) specification provides an open and standardized means for configuring connections and exchanging data between various kinds of automation components, e.g. dynamic process simulators and automation software.

This paper first presents an analysis of the DX specification from the perspective of large-scale simulation use. Then, a DX server design is introduced that provides high performance, without compromising component reuse and portability. Then, the throughput of a prototype DX server is evaluated. A performance test is carried out to demonstrate the applicability of DX-based communication in simulation assisted automation testing purposes.

## INTRODUCTION

Process simulation can be used in various phases of an automation delivery project e.g. to verify process and automation design in specification phase and to validate automation implementation in factory acceptance tests (FAT) at the end of the implementation phase. By using simulation in FAT, one can rehearse the commissioning of the system and to catch the flaws in the application that would normally be caught on the site. This leads to shorter commissioning times and better quality.

In simulation assisted FAT, the automation application reads the measurement values from a simulation model and writes the control values to the model. Typically, the automation application and the simulation model run in separate systems. The connections that are configured between these two systems form the basis for the low-level communication, required to exchange data during the testing procedure.

For obvious reasons, there is a need for a vendor-independent standard for both configuring and

executing data exchange in the testing environment. If open data exchange specifications are used, applications that have been implemented on any platform that conforms to the standard can be tested in the same environment. Similarly, simulation tools by different providers can be connected to the same environment, which may be desirable in case the model comprises submodels that represent different domains, or in case different parts or the model are provided by different companies. Open interfaces to configure connections and exchange data between the applications in this kind of testing environment is hence desirable from several perspectives.

Open interfaces, however, are not enough for successful simulation aided testing if the implementations cannot provide reasonable performance. The main objective of this paper is to introduce a design that can provide large throughput without compromising component reuse or portability. This design and the suitability of the DX specification are then evaluated using simple performance tests that are carried out using a prototype implementation.

Figure 1 presents an example on the co-use of multiple automation and simulation products. The distributed simulation system is controlled by a Simulation Manager application, through which one can define data connections and control simulation. For automation testing use, the system may naturally also include a database for test case definitions and software for the analysis of test run data.

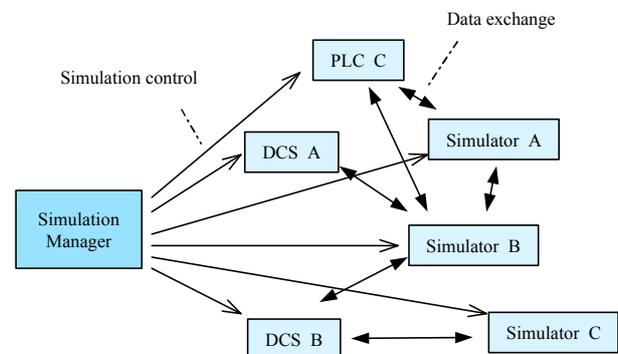


Figure 1: an Example System for Simulation Assisted Testing

## DATA EXCHANGE SPECIFICATION

The aim of the OPC Foundation is to promote open connectivity in industrial automation. The Foundation has developed and released a number of interface specifications for the exchange of data, alarms and events etc., and as these specifications have been implemented in most automation products, OPC has become a popular way to integrate data across the enterprise.

To standardize the horizontal data exchange between automation components, OPC Foundation has released a new specification, Data eXchange (DX) (OPC Foundation 2003a). Earlier, special OPC clients had to marshal the data between two OPC data-access (DA) servers (Laakso 2003; Karhela 2002). The setup had basically two problems. Firstly, each client had a product-specific way to configure the connections, and the connections made with one product could not be re-used with another one. Secondly, due to the client, the communication architecture was unnecessarily complicated and far from optimal in performance. The new specification in principle solves both problems as it removes the need for a special client.

The Data eXchange specification defines how to make connections between a number of DX and DA servers.

Configuring a bi-directional connection between two servers requires that both participants conform to the DX interface specification. Information about connections is as well distributed as each server has a database of its own. The runtime communication is based on the earlier DA specification or the newer XML-DA specification.

A DX server mainly consists of a database for connections, a COM (Component Object Model) or WSDL (Web Services Description Language) interface for updating the database, source access components for DA or XML-DA servers, runtime target item components and support for monitoring and controlling the existing connections. These concepts are quite clearly covered in the DX specification.

A DX database consists of DX connections that can be arranged under DX branches. A DX connection is composed of a target item id, a source item id, a reference to the source server and a number of other attributes. DX clients can modify each connection separately or affect on the behavior of several connections by modifying attributes of both branches and source servers.

The structure of the DX database is quite complicated. All servers have to support e.g. vectors of strings and

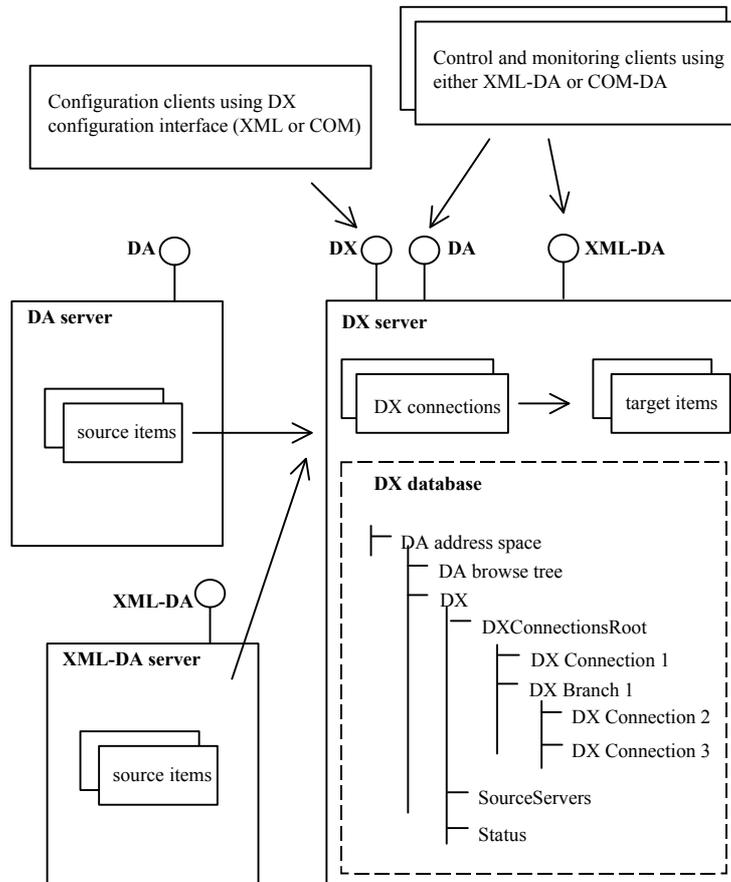


Figure 2: Architecture of DX System (OPC Foundation 2003a)

branches that can simultaneously be items. The database also includes some redundant information, as some structural data can be accessed through the composed string items or through individual simple items. Also the fact that connections may have several parent-branches may be problematic. One feature that clearly makes the server-side implementation rather complex is that there are essentially two distinct methods to affect on the run-time behavior of connections. Using the configuration interface is an obvious way, but controlling the run-time behavior can also be made by writing to some special items that exists in the DX database. Both methods are available either through the COM or the WSDL interface. The run time activity can thus be configured using also plain data access clients. Figure 2 illustrates the DX architecture.

One important requirement from the server is that the operations that use the services of source servers have to be asynchronous. The operations first update the database, and then return the control back to the client. After that, the DX-server asynchronously makes the required operations by calling appropriate source servers. Depending on the status of the source server, the results of the operation may be reflected to one or more items in the database of the corresponding DX server. The client that made the original operation may see this if it is currently monitoring proper items.

From the server-side design perspective the specification is quite complicated, although the configuration interface is quite simple.

A compliant DX server provides a rich set of operations to clients that establish and monitor connections. There can be several clients simultaneously making and monitoring connections. The connections are not client-specific i.e. it is not relevant which client has made the connection. Furthermore all clients may see the whole database. The clients use the standard DA and XML-DA interfaces for monitoring purposes. There is a wide set of operations that clients can make and the effects of the operations greatly depend on the structure and complexity of the database on the server side. The idea is that by creating a meaningful structure in the connection database, clients can conveniently and effectively observe and control data exchange. However, a client has no means for preventing other clients from changing any parts of the database.

The DX specification provides a sufficient set of operations to configure data exchange for co-use of automation and simulation software. The extra complexity originates of the support for two different interface technologies, COM-IDL and WSDL.

Simulation control and synchronization interfaces, that are needed to administrate non-real time simulators, are still lacking standardized approach.

## DX SERVER DESIGN

Adding the features that allow DA servers to act like OPC clients, needs not to be a complicated task. Also the interface that allows making such connections can be very simple. This task became more complicated mainly because the DX working group under the OPC Foundation wanted that connections could also be configured using a web-services (WSDL) interface. The XML-DA interface (OPC Foundation 2003b) is also provided for data exchange purposes. Some features and extra complexity also reflects the fact that DX specification extends the concepts already defined in the earlier DA specification.

### Requirements for Design

Before the DX server design is introduced, the basic requirements for the DX server are listed. An obvious requirement is to implement specification as precisely as possible. For the interoperability reasons, also the optional parts should be implemented whenever it is possible. The second important requirement is to be able to make such server deployments that allow porting web-services part for non-Windows platforms also, particularly to Linux. This is probably the most restrictive requirement. Old COM-based DA components that have already been developed and tested should be able to be re-used as effectively as possible. The performance of the COM-based data exchange should be sufficient for large-scale use. The performance should not suffer considerably e.g. because of the support for connection monitoring or because of the persistence requirements for the DX database.

The implementation should also provide an opportunity to configure and monitor connections simultaneously using the web-services interface and the COM interface. This kind of functionality is, of course, required only on Windows platform.

Essentially, there are three major requirements, that design of DX server should reflect: portability, component reuse and performance of data exchange.

### Overview of the Component Design

Figure 3 illustrates the component view of design under study. It consists of seven components, which build up one executable. The *simulation engine* uses the data-access framework interface (*framework*) to link components that provides standardizes means for external connections for the simulation model.

The *COMKit* exposes standard DA-interfaces and DX-configuration interface implemented as a COM-interface. The *SOAPKit* handles SOAP requests and exposes both the OPC XML-DA interface and the DX configuration interface declared using WSDL.

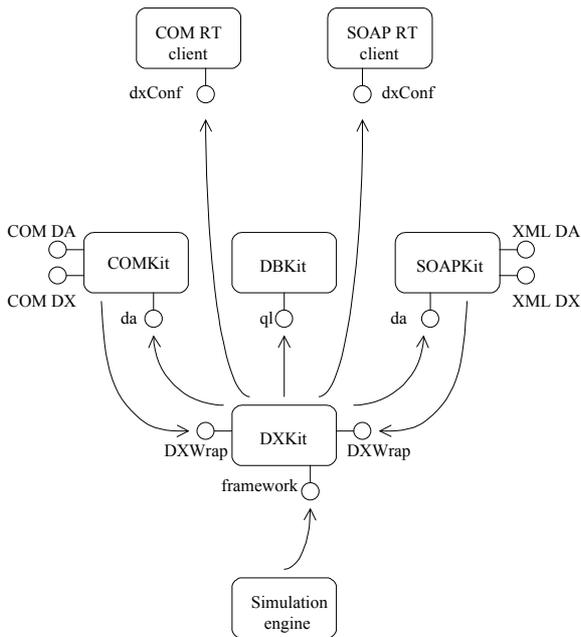


Figure 3: Component View of DX Server Design

The DXKit (*DXKit*) contains major parts of the DX server functionality and marshals data to and from the simulation engine using the framework interface. The COMKit and the SOAPKit marshal DX configuration requests to the DXKit by using a C++ interface *DXWrap*. This internal interface is a one-to-one mapping of the standard OPC DX interface. The DXKit uses a Database Kit (*DBKit*) to build up the persistent database for DX connections. The DBKit provides simple XML-based query language (*ql*) interface to access database entries that consist of DX connections and various simulation models.

The COM run time client (*COM RT client*) is responsible for subscribing data from appropriate source servers if source items are located in a COM DA server. Similarly, the SOAP runtime client (*SOAP RT client*) builds up the appropriate SOAP requests if source items are found on an XML-DA aware source server. The DXKit uses the C++ interface *dxConf* to command these two components. The *dxConf* interface is also a bi-directional interface. These two runtime client components notify the DXKit whenever they get new data from source servers.

### Core DX Functionality

After a brief overview on the component design, a closer discussion about functionality and interactions between these components is examined.

The most central and most multifunctional component in the DX-server design is the DXKit. It is a portable component and one of its responsibilities is to marshal data to underlying simulation engine or to the DBKit. The DXKit hides the actual location of the data from the

COMKit and the SOAPKit. There are essentially three different types of items that can be monitored either through the COM DA or the XML-DA interfaces. The first and most evident type of items is those that are currently loaded to the solver of the underlying simulation engine. The second type of items is those that are not currently loaded to the simulation engine but rather exist in the persistent database of simulator, i.e. in the DBKit. The DBKit can contain several models that can be simulated. These currently not simulated items should also be observable and connectable using standardized techniques. Thirdly, the DX specification defines that every DX database has a similar kind of a structure and e.g. each DX connection can be observed also using standard data-access interfaces. The DXKit hides these details from the COMKit and the SOAPKit using the *da* interface (C++ data access interface) that provides a means for browsing and transferring data. This simplifies the structure of the COMKit and the SOAPKit. It also improves the reusability of the DA server implementation available in the COMKit.

For all active DX connections the DXKit keeps up volatile run time objects as well. A particularly interesting special case is the status information that each connection has. This status information consists of e.g. quality information, timestamps and source item value. The connection status information changes constantly during the run time data exchange. Persisting these values during run time would definitely compromise performance. Depending on the status of the DX connection some data items that the connection has created can be found either in the database (DBKit) or in the volatile memory. This kind of behavior is also hidden from the COMKit and the SOAPKit. Neither component can see the actual location of any individual data access item.

The DXKit hides the actual location of data from monitoring and controlling clients, and it also hides it from the COM RT client and SOAP RT client components. The target item of each active connection may exist either in the solver of simulation engine or in the persistent database (DBKit). Typically the target data of the active connections can be located in the solver of underlying simulation engine, rather than in the persistent database.

Although the DX configuration interface does not contain many operations, the overall result of each operation heavily depends on the structure of the DX connection database. A single operation made by using either the DX configuration interface or through the control items may affect the status of several connections. A good example of this kind of behavior is a case, where a configuration client modifies the attributes of some DX branch that has several child connections. The source items of connections may exist in several separate source servers. Furthermore, each of these source servers may be either COM-DA or XML-

DA servers. The DXKit component resolves this kind of dependencies and commands either the COM RT client or the SOAP RT client component using the simple C++ interface dxConf. Through the dxConf interface the DXKit can create new connections, remove connections and modify the status of each connection. However, the dxConf interface is much simpler than the standard DX configuration interface. Ideally RT components should be as lightweight as possible, and their sole purpose is to get data and to marshal data from source servers to the DXKit. Whoever implements these run time client components, does not need to be aware of the constructs defined in the DX specification.

It is also required that DX connections can be modified and controlled using SOAP and COM clients simultaneously. Centralizing all intelligent functionality in the single component is the easiest way to fulfill this challenge. Otherwise there may arise troublesome inconsistencies and synchronization problems. In this kind of a design all decisions that can affect the run time behavior are made in the same portable component irrespective of the interface type that the configuration and monitoring clients are using.

Finally, the most important justification to centralize the main functionality to the DXKit is to avoid coding similar functionality twice. Clearly, if only e.g. a COM-based DX implementation is needed, tighter integration of a DA server, a DX configuration component and a run time client part, would result in a more compact realization. Similarly, if platform independence of the SOAP-based solution is not an issue, different kind of solutions may be reasonable and more effective.

Because of the challenging requirements, the overall design consists of rather many components. As interfaces between these components are basically bi-directional, the overall structure and the control paths during the operation take quite a complicated form.

### Runtime Data Flow during Data Exchange

The data flow from source items to targets can begin after the necessary data structures have been created to the DXKit as well as to either of the run time components, and to the persistent database of the DBKit. The appropriate run time client component marshals data to the DXKit. The DXKit marshals the data to the appropriate location, typically to the simulation engine. Neither the COMKit nor the SOAPKit participates in data exchange.

In addition to marshalling data, the DXKit is responsible for updating necessary status items that are associated with each DX connection. These status items can be used to observe the current state of the connections and the data flow. Although the status items are contained in the DX database, the status items cannot be persisted during data exchange. Doing such persisting operations at run time would drastically lower

the data exchange performance. The RT clients marshal the connection status information and make appropriate processing only if the DXKit requires that. This behavior is essential to optimize the throughput for the most demanding data exchange needs.

### PERFORMANCE METRICS

Throughput is the most critical performance quantity, when large automation applications are connected with process simulators. An obvious test case consists of two DX servers that are connected symmetrically using COM-based communication. Similar tests have earlier been done for DA-based communication. (Peltoniemi 2001)

A PC with a 1.2 GHz processor and 512-MB RAM was used to carry out the test case. Both DX servers were located on the same computer using Windows 2000 operating system. Creating equal numbers of DX connections in both DX servers created a bi-directional connection between two simulators. Event based (DA2) data exchange was used to retrieve data from the source server. All double precision source items in both servers were continuously changing and the update rate that was used during data exchange was 200ms (Figure 4).

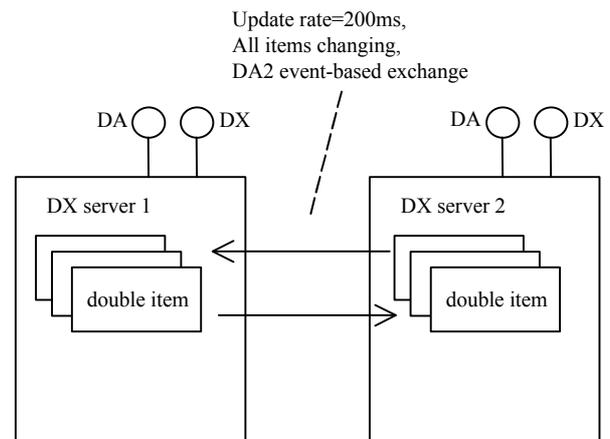


Figure 4: Arrangement of Test Case

None of the connections or items was observed during data exchange. If monitoring clients are simultaneously observing plenty of connection status data, this significantly affects the performance, depending on the needed data type conversions and other properties of item set under monitoring. A particularly heavy load may be generated if plenty of complex DX connection items or connection status items are observed simultaneously. Hence, when performance aspects are critical, also the behavior of monitoring clients is important.

As discussed preceding study (Peltoniemi 2001), it is expected that the throughput depends linearly on the number of connections. This seems to be a valid assumption also in this case, see Figure 5.

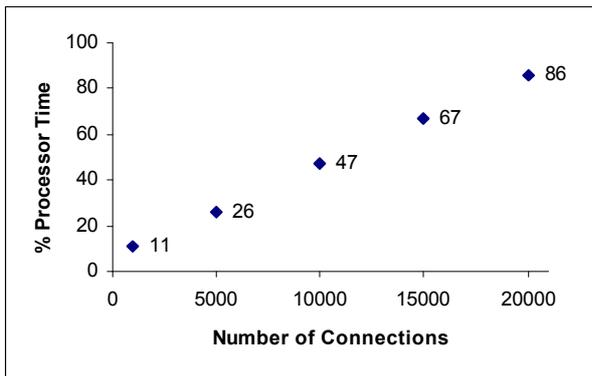


Figure 5: Total Processor Time Using Different Number of Connections

The throughput is significantly better compared to the performance that was achieved in earlier tests, where a separate cross-connector client application marshaled data between two OPC DA servers. This earlier test case was a little bit different, as connections were created for one direction only. The processor load was 81% when 11000 items were transferred from one server to another. The throughput of the DX-based communication is over three times better than the throughput that was achieved using a separate client application to marshal data.

If extremely large-scale models have to be connected with external applications using e.g. Linux platform, the web-services-based communication may not provide a reasonable performance. An insufficient throughput may be a problem for the COM-based communication as well. Proprietary communication could be done e.g. by using an optimized socket-based solution. In that case the configuration of connections could still be made through the standard DX interfaces, only data exchange being done using run time socket components.

## CONCLUSIONS AND FUTURE PROSPECTS

The discussion above concentrated those aspects that are relevant when the components based on the OPC Data eXchange specification are used in simulation assisted automation system testing. A detailed design was illustrated and the design philosophy was justified. A design that provides support for the entire OPC DX specification, including both SOAP-binding and COM-binding was introduced. Component-based design allows reasonable throughput without compromising portability of SOAP-based solution and reusability of existing COM DA server components.

The performance of the data exchange was studied to have a better understanding about the suitability of the DX-based communication in large-scale simulation aided automation testing purposes. The performance of COM-based data exchange should be reasonably good if

both server components are designed in a manner that allows a large throughput.

COM-based data exchange forms the basis for the data exchange between automation system and process simulator in the foreseeable future. However, SOAP-based data exchange defined in the OPC XML-DA specification may become a more attractive choice in forthcoming years. The suitability of XML-DA -based communication for large-scale use will be studied using a similar arrangement.

The ability to effectively and flexibly exchange data between automation software and process simulation models is a fundamental requirement for successful simulation aided automation testing. Using OPC Data eXchange specification, simulation systems can be built that meet this requirement. However, solving the purely information technology related challenges is only the first step in the take-up of simulation in automation testing. In addition, defining proper working methods for simulation aided automation testing and building tools to support the new working methods are essential research challenges in near future.

## REFERENCES

- Karhela, Tommi. A Software Architecture for Configuration and Usage of Process Simulation Models. Software Component Technology and XML-based Approach. Espoo, 2002, Technical Research Centre of Finland, VTT Publications 479, 129p.
- Laakso, Pasi; Peltoniemi, Jyrki; Karhela, Tommi; Paljakka, Matti. The use of OPC in Simulation Systems – experiences and future prospects. The Proceedings of the 3<sup>rd</sup> International Symposium on Open Control Systems 2003, Helsinki, September 9-10 2003.
- OPC Foundation. OPC Data eXchange Specification, 1.0, March 5, 2003.
- OPC Foundation. OPC XML-DA Specification, 1.0, July 12, 2003.
- Peltoniemi, Jyrki; Karhela, Tommi; Paljakka, Matti. Performance Evaluation of OPC-based I/O of a Dynamic Process Simulator. The Proceedings of the 2001 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Orlando, Florida, July 15-19 2001, p. 231-236, SCS, ISBN: 1-56555-240-7.

# SIMULATION OF LARGE STANDARD STILLAGE PLACEMENT ON A DIESEL-ENGINE ASSEMBLY

Robert Steringer

E-mail: Robert.Steringer@profactor.at

Martin Schickmair

E-mail: Martin.Schickmair@profactor.at

PROFACTOR Produktionsforschungs GmbH  
Wehrgrabengasse 1-5, A-4400 Steyr, Austria

Johann Prenninger

E-mail: jprenninger@bmw.co.at

Maximilian Bürstmayr

E-mail: mbuerstmayr@bmw.co.at

BMW Motoren GmbH  
Hinterbergerstr. 2, A-4400 Steyr, Austria

## KEYWORDS

Discrete-Event-Simulation, logistics, forklift, order dispatching strategies, assembly line supply

## ABSTRACT

BMW Motoren GmbH in Steyr, Upper-Austria is the most important manufacturer of diesel engines in the whole BMW automotive group. A new strategy to enhance the capacity on one of the two assembly lines and avoid problems in room limitation as well an expected potential for optimization of forklift utilization has lead to the development of a new set of modules to simulate the supply of the assembly lines. Within the simulation study, different strategies for order assignment (order dispatching) could be examined. The most important parameters could be identified and their influence on the number of line stops due to missing parts, which means low productivity, could be shown.

## INTRODUCTION

The optimisation of mixed model assembly lines by means of discrete event simulation is a well known art. (Banks98), (Fishwick95). Assembling is only one part of the whole manufacturing process. There are many other supporting processes like ordering, delivery, commissioning, supply and finally disposal of waste goods. This is known as the logistics chain. It can be distinguished between administrative tasks (keep all suppliers informed, manage the time table of transports etc.), the so called business-process and the physical processes itself (commissioning, transportation ...).

The processes of manufacturing and assembling are, at least in the case of mass production, well defined. Building a model for these processes is therefore a straight forward task, maybe challenging because of complexity or quantity, too. Within the logistics chain many tasks are not described into detail or depend on the human being, who has to make its decisions based on its individual experience and skills. Therefore, accomplishing a task may result in big differences in the way it is done and in the time needed for it.

Some of this uncertainty can be solved by traditional means of modelling, such as using a random value with a suitable distribution instead of an exact value.

On the other hand you have to make more assumptions to keep the model simple and to avoid to get lost in details.

## SITUATION ANALYSIS

BMW Motoren GmbH is the main manufacturer of 6 cylinder petrol and all 4 and 6 cylinder diesel-engines in the whole BMW automotive group. In 2002 about 561.000 engines have been produced, spread on 326.000 diesel and 235.000 petrol engines.

Diesel engines are produced on two assembly lines. Line 3 is responsible for assembling 20 different types of 6-cylinder diesel engines. It consists of four sections. The first section is built using conventional conveyors. All other sections rely on AGV's to transport the engines to the following station. Line 4, built just in the beginning of this century, uses conventional conveyors for the whole assembly line. On this line, 6 different types of diesel engines are built. Because of the limitation of available room, this line is spread over two floors. It is said, that, in terms of room efficiency, it is one of the most efficient engine assembly lines worldwide.

Within this paper (and the project) the focus is laid on the supply with large standard stillage (LSS) only. Both lines are supplied with material by means of 2 independent lifts. Each of them can deliver goods to both lines. Each line relies on different strategies both for reordering and for delivering requested parts. There is only a minor difference in process flow in the disposal of empty LSS-boxes.

Line 3 has a small intermediate storage in a close range of each assembly station. When nearly empty, a LSS will be replaced with a fresh one from the intermediate storage by a requested forklift. The forklift removes the empty LSS and places the full one at the station. On some stations the forklift has to wait until the last part is assembled and the LSS is entirely exhausted. On the other stations the remaining parts can be reloaded into the new LSS by the worker. This allows the forklift to change immediately without waiting. When the replacement of the LSS is finished, the forklift dumps the empty onto a waste goods trailer. Additionally, the forklift driver is responsible for reordering from the ware-

house. When, the new full LSS is delivered by the lift it is placed again in the intermediate storage.

Line 4 has two central intermediate storages instead of the individual nearby storages. Filled LSS's are brought to a commissioning zone close to a group of assembly stations. The forklift places the LSS on a special low level vehicle and is immediately ready for the next job. When the old LSS is completely exhausted, it is brought to the commissioning zone by the responsible assembly operator. When returning, he is pushing the LSS on the vehicle to its destination at the station. Now, the forklift has to return to the commissioning zone and dump the empty LSS as described above.

This strategy has been necessary to overcome troubles with limited space for manoeuvring the forklift within the assembly zone. It has the advantage (from the logistical point of view), that the forklift has no need to wait until the last part of the LSS is assembled. The disadvantage is that it depends on free vehicles, whose maximum number is limited due to the restricted space within the commissioning zone. In addition, the assembly operator complains about the extra task of changing the LSS. When requesting a forklift to supply a new LSS from the intermediate storage an order to the central warehouse is placed automatically.

The lines are, although situated side by side in a common building and using common resources like warehouses and lifts, completely differently and independently structured. The assembly and also the logistics personnel are divided into different departments. There is no direct cooperation between these departments. Also the placement of material on the line is performed by different groups of forklifts, who do not support each other (at least officially). Each forklift has a certain job, e.g. maintain a certain section or unload the lift and take the parts to the intermediate storage.

### Goals

It is evident, that combining both groups of forklifts can improve security of supply, maybe even with fewer resources. BMW decided to make a simulation study to evaluate possible benefits. The study should show whether:

- It is possible to unite these two forklift groups
- What organisational structures are needed?
- What strategy for task distribution is best?
- How much resources are needed to supply both lines?
- How many vehicles are needed at the various commissioning zones to minimize forklift waiting times and insure zero line stops due to missing parts?

Because of the corporate wide strategy in discrete event simulation, it was a preliminary condition to use the

simulator eM-Plant™ (Tecnomatix Technologies Ltd., <http://www.eMPlant.com/>) for building the model. The model had to rely on only the eM-Plant standard components, and was not allowed to use any AOL's or other commercial libraries.

### Assumptions

As stated above, there are many uncertainties and exceptional cases in modelling these processes. Many data needed was not available and even not collectable (e.g. average speed of forklifts) because of legal circumstances. In many situations the staff is able to change the standard processes, e.g. reroute the path to a station because of traffic. The decision of taking another route or wait, maybe just a few seconds until the traffic jam is over, is complex to model. However, for an experienced forklift operator it is simple and rather intuitive. For implementation of a simulation model you need concrete specifications. So we decided to make some preliminary assumptions:

- When working, a forklift does not disturb other driving forklifts.
- Forklifts can only carry one LSS at a moment in time
- All forklifts have the same velocity. There is no need for taking care for acceleration/deceleration because of the usage of a low average velocity.
- There is a strict priority in the timetable-schedule of performing jobs:
  - Refuelling (change batteries)
  - Unload lifts
  - Maintain stations with necessary parts
  - Dump wasted material
- Within the priorities the sequence of orders is strictly chronological
- There is a possibility to order a certain forklift for a given task
- There is no focus on the assembly stations itself. We did not model any failures or maintenance periods. The cycle time is not type dependent. There are no extra buffers between stations. The only failure a station may have is if there is at least one missing part. All failures of this type are to be logged.

### THE MODEL

The model was built using the discrete event simulator eM-Plant™ in version 4.6. As we started from scratch, we had to implement not only components for stations, forklifts with order dispatcher, roads, crossings, stores, lifts, dumping, reordering, battery changing but also "helper" components to increase comfort and efficiency in performing experiments with the model.

According to the assumption, that forklifts do not disturb other forklifts, we built a system of bi-directional roads. We implemented several types of crossings with various numbers of crossing roads. They are needed not only for the road net itself but also in order to approach the stations. We decided to use the standard “automatic routing” capability. Thus it was necessary to implement “Forward destination Lists” for every track within a crossing. Because we had to implement 20 crossings and additionally 120 approaches it was important to create a tool to fill in these lists automatically. As a secondary effect we got a matrix of (shortest) distances between all destinations. This matrix is also used by the order dispatcher when the “find nearest forklift” strategy is used.

Our central order deployment component, the so called “dispatcher” was built as the most “intelligent” part of our model. It can be extended with new dispatching strategies very easily. We implemented simple *First Come First Serve (FCFS)* strategies as well as a more sophisticated “*Find nearest Forklift (FNF)*” strategy or a strategy to ensure almost “*Perfectly Balanced Utilization (PBU)*” of forklifts. These strategies are used for dynamic order assignment. In contrast to these strategies we also implemented a static order assignment. This strategy causes the dispatcher to assign all tasks of a given station to one designated forklift. We used this configuration to model the “traditional” order assignment.

Every assembly station knows about all parts it assembles. It needs type dependent knowledge of part consumption as well as information about the LSS. (e.g. the number of boxes within a station and in the intermediate storage) and about the number of parts within a box. When the content of a LSS is falling short of an adjustable limit, a new order is placed. If there is less than the required amount of parts, a failure is raised. Immediately, the “guilty” part and the duration of the failure are logged. It has been necessary to implement an “Emergency Order” strategy to prevent forklifts of deadlock situations.

For statistics evaluation the forklift utilization is divided into the states *Driving, Waiting and Loading (Unloading)* and *Free*.

We implemented an “ActiveX” based interface for comfortable key-in in of data with MS-Excel™. This interface is also used for building an “Auto-Experimenter” – a tool which automates the parameterisation and execution of experiments. Tecnomatix delivers newer versions of eM-Plant with a similar tool.

## RESULTS

The simulation proofed that the new concept of maintenance for line 4 with vehicle based LSS exchange will work. The necessary amount of vehicles on the commissioning zones could be defined.

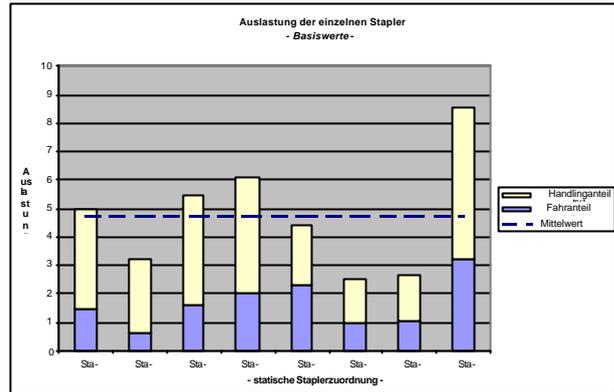


Figure 1: Utilization of forklifts (2 groups, static order assignment)

In our first series of experiments, we examined the influence of different strategies of order assignment. We found that the average value of forklift utilization in standard order assignment (Figure 1) is only marginally higher than in the FNF strategy (Figure 2). (All results shown rely on special, hypothetical conditions).

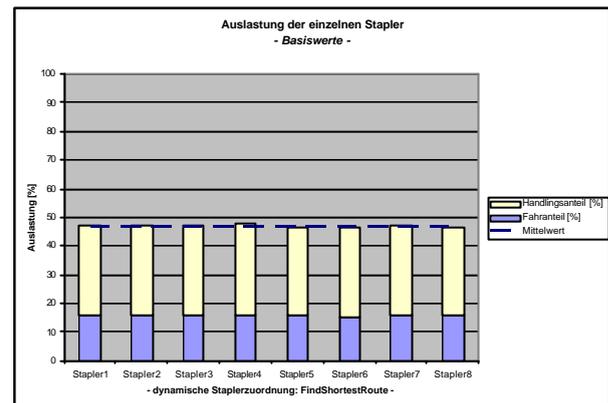


Figure 2: Utilization of forklifts (1 groups, dynamic order assignment (FNF strategy))

This is especially true, when there are few forklifts with an already high utilization. Although it was not in our special intention, all forklifts are almost equally balanced.

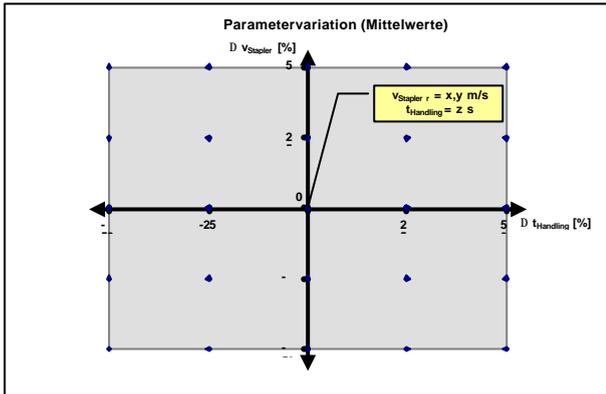


Figure 3: Range of examination: parameter speed and load time

This first experiment showed very different forklift utilization. The diagram reflects the original static order assignment. In reality, there is not such a rigid “demarcation line” between the tasks of the individual forklift. The operators will help each other. This makes the differences in utilization among all forklifts smoother. Furthermore we did not make any effort to improve the balance by reassigning stations properly.

The *FIFO algorithm* takes care for good balancing too, but causes a slightly higher utilization.

The *PBU algorithm* also resulted in higher utilization, but decreased the differences among all forklifts.

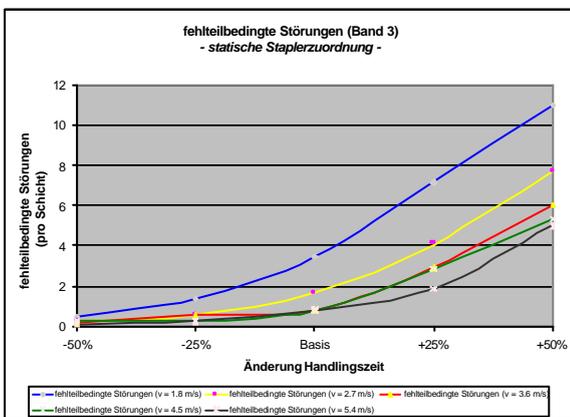


Figure 4: Number of line stops due to missing parts (2 groups, static order assignment)

Because of the measurement of data, such as forklift speed or load time being prohibited, we had to rely on approximations and theoretical computations of an average speed.

Therefore, we made an experiment series to get a feeling for the influence of these parameters. We made an analysis about sensitivity for these parameters by performing experiments on different parameter combinations in a range of +/- 25% and +/-50 % (Figure 3) of

the estimated values. We also measured the influence of these parameters on the number of line stops.

In cooperation with the affected foreman, consensus values for forklift velocity and load / unload time finally could be found.

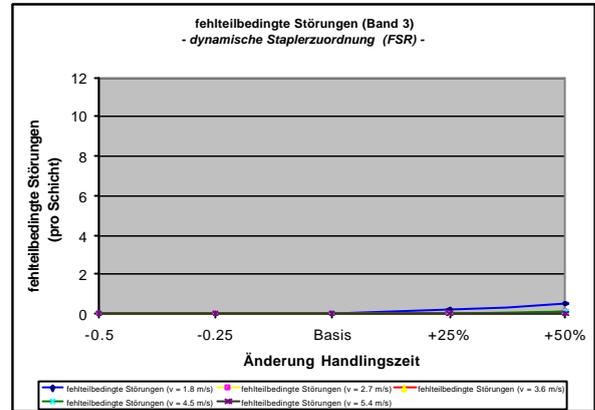


Figure 5: Number of line stops due to missing parts (1 group, dynamic order assignment (FNF strategy))

As expected, the number of line stops increased with the forklift utilization (Figure 4). There was a legible improvement when using the dynamic order assignment (Figure 5). There are almost zero line stops in the whole parameter range.

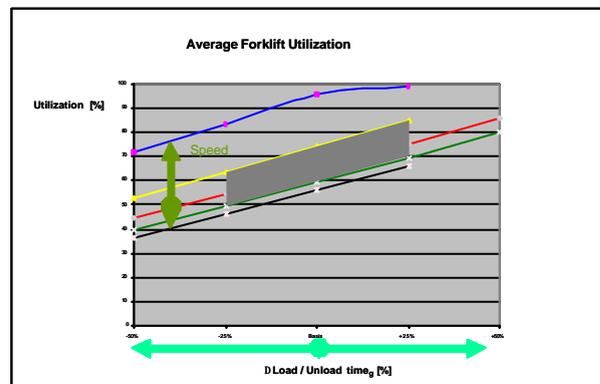


Figure 6: Average utilization (1 group, “FNF” strategy, 6 forklifts)

In addition, we checked the forklift utilization for a different number of forklifts. With an optimistic choice of both parameters (speed and load time) the number of forklifts could be reduced from 8 down to 4. Due to safety regulations and the rather narrow lanes, a more pessimistic set of parameters have been chosen. The computed average utilization with 6 forklifts in use and different values for load/unload time and forklift velocity is shown in Figure 6.

## CONCLUSION

Building a model for assembly line supply needs a lot of process knowledge. Many tasks are not defined into detail. The forklift operators usually are very free in determining which task should be performed next. They have little assistance in planning their next tasks. The duration, for both transporting and handling of large standard stillage depends very much from the skills of the operator. Therefore a lot of assumptions must have been made to keep the model "as simple as possible".

In tight cooperation with the forklift operating foreman, feasible and plausible parameter values for forklift speed and load / unload time could be found. The course of events, though simplified, is validated and accepted by the experts.

The experiment results showed that there is no major difference in forklift utilization, when comparing dynamic order dispatching strategies with (well balanced) static strategies. Dynamic strategies have an advantage in scalability and supply guarantee. Of course, there is a need for organisational measures like an additional order dispatching system including technical equipment for communication with all forklift operators.

The study we made lead to (until now) three new projects and each of them extended and improved our set of modules and our know-how in modelling assembly line supply.

## REFERENCES

Banks Jerry (ed.): *Handbook of Simulation – Principles, Methodology, Advances, Applications, and Practice*; John Wiley and Sons, 1998; ISBN 0-471-13403-1.

Fishwick Paul A.: *Simulation Model Design and Execution: Building Digital Worlds*; Prentice Hall, 1995;

Kosturiak, J. Gregor, M.: *Simulation von Produktionssystemen*. Springer Verlag, 1995

Heidenblut, Volker: *Software-Simulation im Vorfeld der Inbetriebnahme von Logistik-Projekten*. Logistik für Unternehmen, Heft: 11, 2000, S. 28-33, ISSN 0930-7834

Benecke, Carlo: „*Simulation von Materialfluss- und Lagerprozessen*“. Zeitschrift für Logistik, Heft: 5, 1990, S. 30 - 32

B. Schmidt: *Die Modellierung menschlichen Verhaltens*. SCS-Europe BVBA, Ghent, Belgium, 2000

## AUTHOR BIOGRAPHIES

**ROBERT STERINGER** is with Profactor Produktionsforschungs GmbH, a non-profit manufacturing research institute located in Steyr, Austria. He received a degree of a Dipl.-Ing. in Computer Science from Vi-

enna University of Technology. Since 1999 he is working at the company's simulation department. Besides his activities in simulation he is also interested in software engineering.

**MARTIN SCHICKMAIR**, born in Wels, Austria studied electrical engineering at the Technical University Vienna and received a degree of a Dipl.-Ing.. He is also with Profactor Produktionsforschungs GmbH where he joined the company's simulation department in 2000. His special interest is the integrated simulation of technical and business-processes.

**JOHANN PRENNINGER**, born in Linz, Austria studied electrical engineering at the Technical University Vienna. In 1992 he received a Ph.D. in Flexible Automation and robotics from TU-Vienna. In 1992 he received the JIRA award for the best scientific work in robotics from the Japan Industrial Robotics Association. From 1993 - 1999 he was working as executive manager of a research company working in the field of production technologies and process automation. Since 2000 he works for BMW engine plant in Steyr as head of the logistic planning department. He can be reached by email: Johann.Prenninger@aon.at

**MAXIMILIAN BÜRSTMAYR** is with BMW Motoren GmbH since 1999. He is responsible for planning of logistics and material flow on all of the companies final engine assembly lines. Furthermore he is the contact person for all simulation activities within the department.

# NON-LINEAR MODEL REFERENCE CONTROL OF pH PROCESS: AN EXPERIMENTAL STUDY

Nayeem N. Karnachi and Geoff Waterworth  
Faculty of Information and Engineering Systems  
Leeds Metropolitan University  
City Campus, Leeds LS1 3HE  
United Kingdom  
n.karnachi@lmu.ac.uk

## KEYWORDS

Non-linear, LabVIEW, experimental study, pH control, simulation and control design.

## ABSTRACT

The control of pH is important in many processes including wastewater treatment, chemical processes and biological processes. This paper considers a model reference non-linear controller developed by Jayadeva et al. (1990a). The method is tested using a 7-litre continuously stirred tank reactor to neutralise a strong acid using a strong alkaline solution. The method is first realised using a simulation of the process. Subsequently it is demonstrated on an experimental rig using real-time control. Experimental results confirm that a robust control of the process is achievable.

## INTRODUCTION

In many processes, pH neutralisation is a very fast and simple reaction. In terms of practical control, it is recognised as a difficult control problem (Shinsky 1973; Pishvaie et al. 2000; Wright et al. 1991). The difficulties arise from high process nonlinearity (the process gain can change tens or hundreds of times over a small pH range) and from changes in the pH characteristics due to changes in influent concentration. Various techniques have been developed to control process pH. Young and Rao (1986) presented a variable structure controller ("sliding mode control for a neutralisation process") involving strong acids and bases. Parrish and Brosilow (1988) used non-linear inferential control in a simple simulated neutralisation process, using static estimation of the concentration of a single monoprotic weak acid. Kulkarni et al. (1991) presented non-linear internal model control for a simulated system of sodium hydroxide (NaOH) and hydrochloric acid (HCl). Li et al. (1990a) and Li and Biegler (1990b) presented non-linear feedback methods for a simulated neutralisation process. In the present work, a non-linear controller design is implemented. It

uses a design procedure presented by Jayadeva et al. (1990b). The controller is implemented practically on a 7-litre reactor.

## THEORY

In the present work, the design of a robust non-linear controller is introduced. It considers a model reference controller developed by Jayadeva et al. (1990a). The method is taken originally from a paper by Yuocef-Toumi and Ito (1987). The control scheme is illustrated in Figure.1.

### Controller Design

Consider a single input and single output (SISO) state variable system of the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \dots, x_n) + g_1(x_1, x_2, \dots, x_n)u + d_1(x, t) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n) + g_n(x_1, x_2, \dots, x_n)u + d_n(x, t) \end{aligned} \quad (1)$$

$$y = c_1x_1 + c_2x_2 \dots + c_nx_n \quad (2)$$

where,  $u$  is a scalar manipulative input,  $x_1, x_2, \dots, x_n$  are the states and  $y$  is a scalar output.  $f_i$  and  $g_i$  are nonlinear functions of state variables.  $d_1, d_2, \dots, d_n$  represent general disturbances. The output variable  $y$  is a linear function of the state variable.  $c_1, c_2, \dots, c_n$  are constant scalars. Yuocef-Toumi and Ito presented a robust nonlinear feedback controller design for a general nonlinear multi-input state variable system, from which a least square solution for the manipulative variable was obtained. The method is applied to the specific form of Equation (1) and (2) to obtain an exact solution for the manipulative variable. Equation (1) and (2) can be written in vector form as

$$\dot{x} = f + gu + d \quad (3)$$

$$y = cx \quad (4)$$

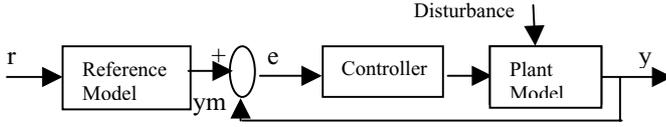


Figure 1: Model Reference Control Scheme

where the vectors  $f$  and  $g$  are functions of  $x$ , and  $c$  is a constant row vector. Let us assume the reference model in the scalar output  $y_m$  is given by

$$\dot{y}_m = \lambda_m y_m + b_m r \quad (5)$$

where,  $\lambda_m$  is the eigen value of reference model. The scalar  $e$  is defined as the difference between reference value and the process output. Therefore,

$$e = y_m - y \quad (6)$$

The control objective is to force the error to vanish with a desired dynamics:

$$\dot{e} = \lambda e \quad (7)$$

Where,  $\lambda$  is the eigen value for the error system. By combining Equations (3) – (7) we obtain the equation that governs the error dynamics. Therefore,

$$\dot{e} = \dot{y}_m - \dot{y} \quad (8)$$

$$= \lambda_m y_m + b_m r - cf - cgu - cd$$

$$\text{as } \lambda_m e = \lambda_m (y_m - y) \text{ therefore,}$$

$$= \lambda_m e + (\lambda_m y + b_m r - cf - cgu - cd) \quad (9)$$

It is possible to determine the manipulative variable  $u$  in Equation (9) such that

$$(\lambda_m y + b_m r - cf - cgu - cd) = ke \quad (10)$$

From which we have the manipulative variable

$$u = (cg)^{-1} [b_m r - cf - cd - ke + \lambda_m y] \quad (11)$$

Therefore Equation (9) becomes

$$\dot{e} = (\lambda_m + k) e$$

$$\dot{e} = \lambda e \quad (12)$$

Where,  $k$  is a scalar error feedback gain. The error system eigen value  $\lambda$  can be assigned arbitrarily through proper choice of the error feedback gain  $k$ . The control law in Equation (11) is used to calculate  $u$  in order to get the desired error dynamics (Jayadeva et al. 1990a).

Now consider the application of the above control design to the model for the pH process described by McAvoy et al. (1972). The process consists of a strong acid flowing into a constant volume tank which is thoroughly mixed with a strong base. The feed flow rate of the base is to be controlled in such a way as to produce a neutral outlet from the tank. The equation describing this process is given by

$$\dot{x} = a_1 x - a_3 u(x + a_2) + a_1 D \quad (13)$$

Where,  $x$  is the deviation from neutrality. Note that,  $x$  and the pH value  $y$  are related by the non-linear equation:

$$x(t) = 10^{-y(t)} - 10^{-y(t)} K_w \quad (14)$$

Where  $K_w$  = water equilibrium constant =  $10^{-14}$ ,  $a_1 = F_1/V$ ,  $F_1$  is the acid flow in litres and  $V$  the volume of the mixing tank;  $a_2 = C_{\text{base}}$  = concentration of base;  $a_3 = 1/V$  are constant parameters;  $u = F_2$ , is the manipulative variable, base flow control in litres;  $D = C_{\text{acid}}$  = concentration of acid = the disturbance variable. It is to be noted that Equation (14) is valid for the strong acid / strong base case only. For the general case, there are two model equations (Wright et al. 1991; Shinsky 1973).

Now, comparing Equation (13) with (3), we have,

$$f(x) = -a_1 x; \quad g(x) = -a_3 (x + a_2); \quad d(t) = a_1 D$$

And the output equation,

$$h(x, y) = x + 10^{y-14} - 10^{-y} = 0 \quad (15)$$

The control objective is to keep  $y(t) = 7 = \text{constant}$  in the presence of disturbances occurring in the process in general, making  $y(t)$  follow a given reference trajectory. In the control design, the output equation is a linear function of the state variables. But, Equation (15) is a non-linear implicit output equation. Hence, for this nonlinear process, the controller design procedure requires to be suitably modified. Therefore we apply the following partial differentiation identity to Equation (15):

$$\frac{\partial h}{\partial y} \dot{y} + \frac{\partial h}{\partial x} \dot{x} = 0 \quad (16)$$

hence

$$\dot{y} = - \left\{ \frac{\frac{\partial h}{\partial x}}{\frac{\partial h}{\partial y}} \right\} \dot{x} \quad (17)$$

Using Equations (3), (6), (8) and (17) we get:

$$\dot{e} = \lambda_m y_m + b_m r + \left[ \frac{\partial h / \partial x}{\partial h / \partial y} \right] [f(x) + g(x)u + d(t)] \quad (18)$$

$$= \lambda_m y_m + b_m r + J [f(x) + g(x)u + d(t)] \quad (19)$$

$$\text{where } J = \frac{\partial h / \partial x}{\partial h / \partial y} \quad (20)$$

If we make,

$$\lambda_m y_m + b_m r + J [f(x) + g(x)u + d(t)] = ke \quad (21)$$

then the control law is calculated as:

$$u = -(Jg)^{-1} (\lambda_m y + b_m r - ke + Jf + Jd) \quad (22)$$

Equation (19) becomes:

$$\dot{e} = (\lambda_m + k)e \quad (23)$$

$$= \lambda e \quad (24)$$

It is to be noted that since the disturbance term  $d(t)$  appears in the control law, it is essentially a combined feedback-feedforward control action (Jayadeva et al. 1990a). The expression for the control law of Equation (22) in terms of the plant variable  $y$  only is given by

$$u(t) = \frac{[(10^{-14} + 10^{-y})(2.303) (\lambda_m r + b_m r - ke) - a_1(-10^{-14} + 10^{-y}) + a_1 D]}{[a_3(-10^{-14} + 10^{-y}) + a_3 a_2]} \quad (25)$$

## EXPERIMENTAL SET UP

Figure 1 shows the experimental set up for the pH neutralisation system. The process stream (influent) consists of a diluted strong acid (HCl) and the titrating stream is a more concentrated strong base (NaOH). Table 1 consists of typical operating conditions. The process stream is fed through two feed tanks, and a 3-way valve is placed in the feed line, which allows switching between two different feed concentrations. A remote control peristaltic pump (RM pump) is used to control the flow rate of the titrating stream. The volume of the reactor vessel is kept constant at 5-litres with an over flow system.

An agitator is used to ensure proper mixing. The pH of the influent, the pH of the mixture in Continuously Stirred Tank Reactor (CSTR) and the influent flow are measured by a data acquisition system (National Instruments E series I/O card and a PC with LabVIEW Instrumentation package). The control objective is to maintain the pH value at the set point = 7. The control output is calculated according to the non-linear model reference control law

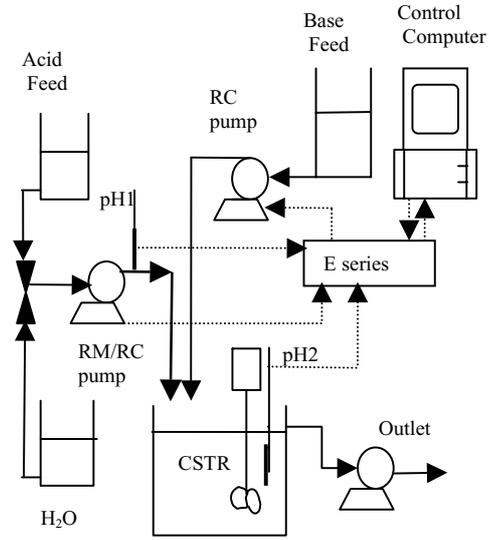


Figure 2: Laboratory set-up of the pH Neutralisation Process

(Equation (25)). The digital output is converted to an analogue output, and the signal is transmitted to a remote control peristaltic pump (RC pump) that controls the base flow rate. The sampling time for the measurements is 0.1 of a second and the control law is executed at approximately the same time (considering the time taken for control computation by the package and the operating system).

## SIMULATIONS

A continuous time simulation of the controller was undertaken to confirm the results obtained by Jayadeva et al. (1990a). Figure 3a shows the open loop response of the plant for a 100% disturbance in the concentration of the influent at 1.4 seconds. Figures 3b and 3c are continuous controlled responses of the plant and the controller

Table 1- Typical operating conditions for the pH neutralisation process

Parameters	Values
Acid Flow ( $F_1$ )	Variable
Base Flow ( $u$ )	Manipulative variable
Conc. Of acid ( $D$ )	0.01M – 0.005M
Conc. Of base ( $a_2$ )	0.2M
Volume ( $V$ )	5 litres

respectively for the same disturbance at the same time. Figure 3d is the simulation response of the plant for a change in the operating point from pH 7 to pH 3 and the disturbance in concentration at 1.4 seconds. The controller responds robustly to both the disturbances. The robustness of the controller was also confirmed with a disturbance in the flow of the influent along with concentration.

Finally, the continuous controller is studied with sampled input and output signals before practical implementation.

Hence zero order holds are applied to model this effect on the continuous process (Figure 4). The Simulink model incorporates the change in disturbance with respect to time as shown in Figure 4. The effect due to the change in influent flow was also studied with slight modification in the model. Analysis was done for the allowable sampling time for real-time implementation with Zero-order hold at both input source and output sampling.

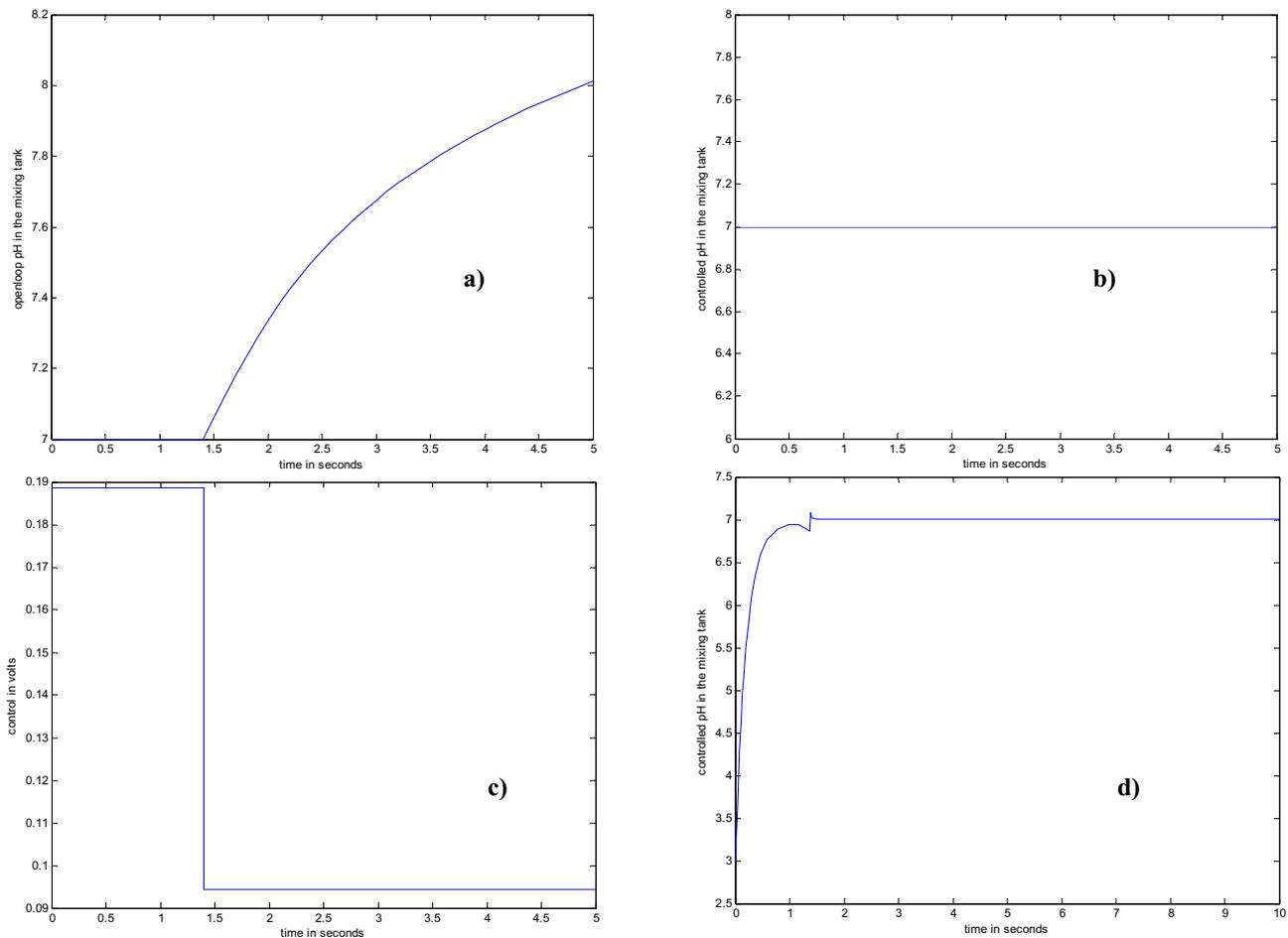
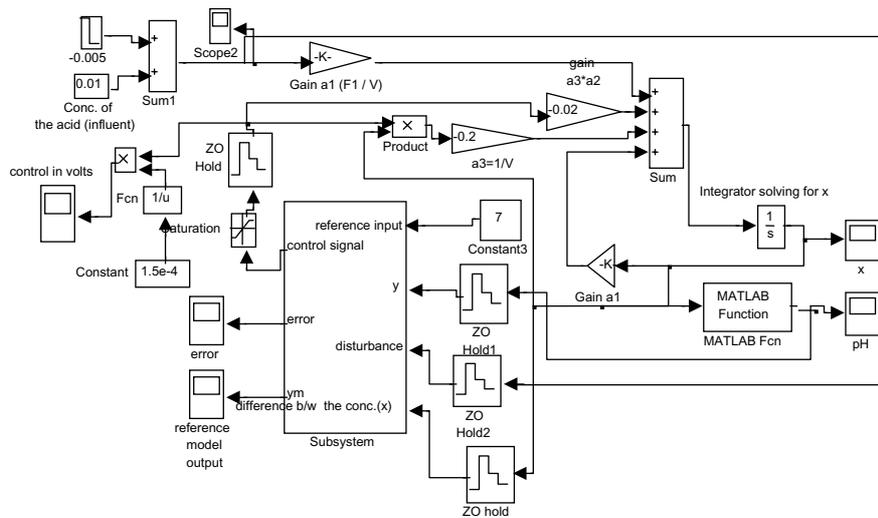


Figure 3: a) Step response simulation of the plant for a disturbance of 100% reduction in influent concentration. b) Controlled pH value for the same disturbance at 1.4 seconds as in a). c) Controller in action for the disturbance of 100% reduction in the influent concentration. d) Controlled pH for a simulation of the controller at different operating point along with the disturbance at 1.4 seconds as in a)

## REAL TIME IMPLEMENTATION OF THE CONTROLLER

The pH sensor is assumed to be linear and the temperature is assumed to be constant (Shinsky 1973).

pH sensors have very high source impedance and it is therefore necessary to use a high input impedance buffer amplifier. A low pass filter is used to reject AC mains 50Hz.



Closed Loop Simulation of the 7-Litre Experiment Rig with Reference Initial pH value at 7 and the plant initial value of  $x = 0$

Figure 4: Simulink Model of the Process.

Differential analogue input mode is preferred to single channel analogue input of the I/O card for sensor signal feed, as Common Mode Rejection Ratio (CMRR) is very high in this mode. The control signal range for the pump so that it responds linearly is 0-10 volts. The concentration of the solutions is accordingly chosen considering the constraint.

### Software Platform

LabVIEW, a real-time virtual instrument package, is used to implement the control strategy. The easy accessibility of Matlab code within the LabVIEW environment is utilised for complete implementation and data retrieval. Figure 5 partly shows the VI diagram of the program for 0.1 of a second sampling delay.

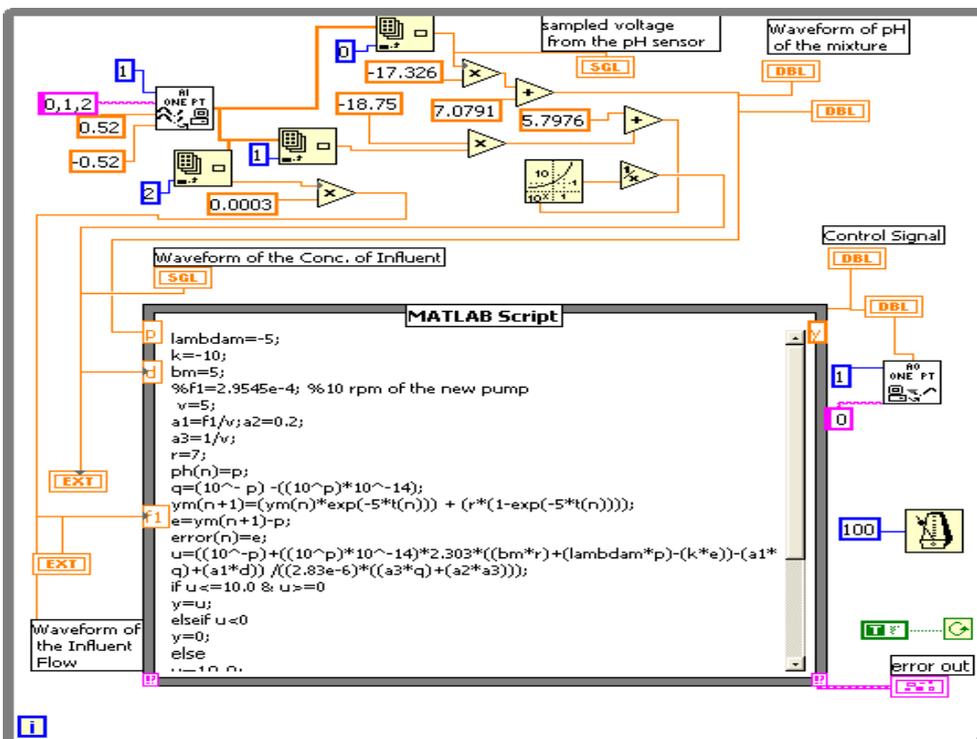


Figure 5: Pictorial VI Block Diagram of the pH Process

## ANALYSIS

The controller is tested for the most common disturbances, which are the change in the flow of the influent and the concentration. The experiment is conducted approximately for 3 minutes with the change

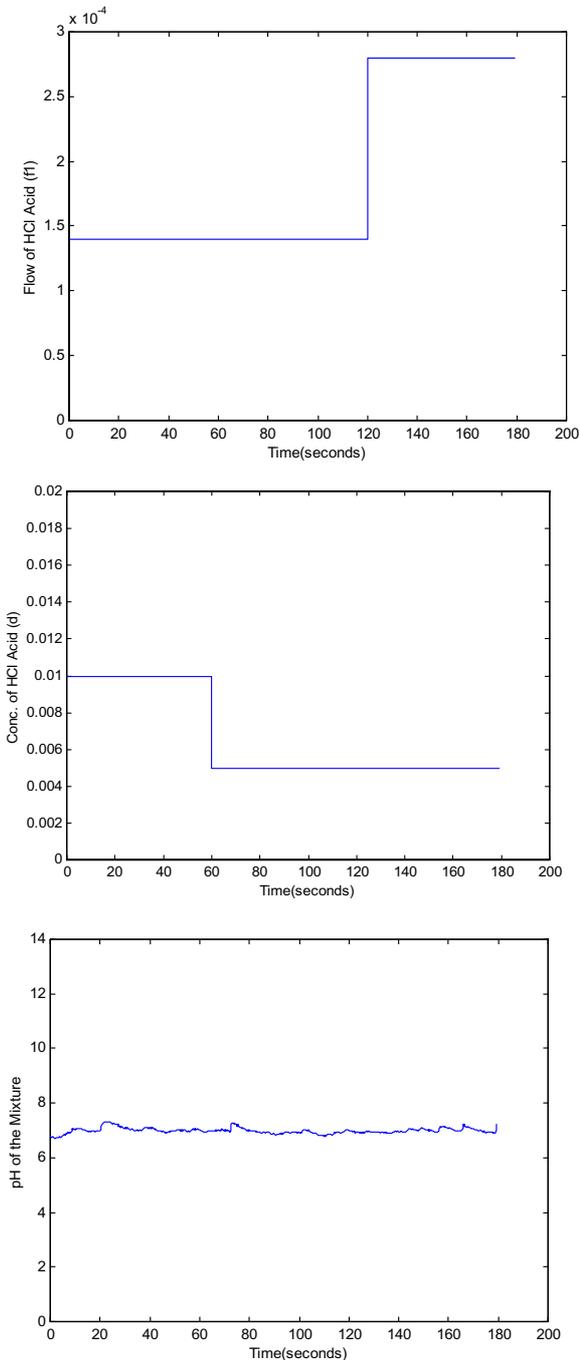


Figure 6: pH response of the mixture in the CSTR for the disturbances in the influent flow (after 2mins and concentrations (after 1min)

in concentration after 1 minute (Figure 6) and then the change in flow after 2 minutes. The controller robustly responds for the disturbances with no apparent change in the pH of the mixture. It was inferred during simulation that, the maximum sampling time can be 0.2 second. But the response of the plant was not as quite the continuous time response (Figure 3). The reason can be studied further. The instantaneous control values were noted to be the same as the continuous time values.

## CONCLUSION

This study is a part of the research to propose a non-linear adaptive control scheme for pH control of wastewater and implement it on an industrial scale for a water company in the United Kingdom. A lot of research is only simulation based understandably due to many factors such as cost etc. Therefore, the importance of real-time implementation has also been emphasised in this study. This study has opened doors for further investigation into simulation and real-time implementation. As this study aimed at exploring requirements to liaise software with hardware, the experimentation has been successful in doing so.

## REFERENCES

- Jayadeva, B. J., Y. S. N. M. Rao, M. Chidambaram and K. P. Madhavan. 1990a. "Nonlinear Controller for A pH Process." *Computers chem. Engng.*, 14(8), 917.
- Jayadeva, B. J., M. Chidambaram and K. P. Madhavan. 1990b. "Robust Control of Batch Reactors." *Chem. Engng. Commun.* 87, 195.
- Kulkarni, B. D., S. S. Tambe, N. V. Shukla and P. B. Deshpande. 1991. "Non-linear pH Control." *Chem. Eng. Sci.*, 46, 995.
- Li, W. C., L. T. Biegler, C. G. Economou, and M. A. Morari. 1990a. "Constrained Pseudo-Newton Control Strategy for Non-linear Systems." *Comput. Chem. Engng.*, 14, 451.
- Li, W. C. and L. T. Biegler. 1990b. "Newton-type Controllers for Constrained Non-linear Processes with Uncertainty." *Ind. Eng. Chem. Res.*, 29, 1647.
- McAvoy, T. J., E. Hsu and S. Lowenthal. 1972. "Dynamics of pH in a Controlled Stirred Tank Reactor." *Ind. Eng. Chem. Process Des. Dev.*, 11(1), 68.
- Parrish, J. R. and C. B. Brosilow. 1988. "Non-linear Inferential Control." *AIChE Journal*, 34, 633.
- Pishvaie, M. R. and M. Shahrokhii. 2000. "pH Control using Non-linear Multiple Models, Switching and Tuning Approach." *Ind. Eng. Chem. Res.*, 39, 1311.
- Shinsky, F.G. 1973. "pH and pION Control in Process and Waste Streams". John Wiley & Sons, New York, USA.
- Wright, R. A., M. Soroush and C. Kravaris. 1991. "Strong acid equivalent control of pH processes: An experimental study." *Ind. Eng. Chem. Res.*, 30, 2437.
- Yuocef-Toumi, K. and O. Ito. 1987. "Controller Design for Systems with Unknown Nonlinear Dynamics." Proc. Am. Control Conf. Minneapolis, 836.

Young, G. E. and S. Rao. 1986. "Robust control of a Non-linear Process with System Uncertainty and Delay using Variable Structure," *Proc. Am. Control Conf.*, 1210.

## **AUTHOR BIOGRAPHIES**

**GEOFF WATERWORTH** started his career with British Aerospace, Stevenage, U.K., in 1964 and went on to work with Ferranti, SGS Fairchild, etc.,. He is currently serving as a Senior Lecturer at the Leeds Metropolitan University, U.K. since 1978. Here, he leads a research group with interests ranging from modern adaptive control techniques, optimisation of pump scheduling in water supply systems and sensor fault detection in water quality monitoring. He has also written a number of books and has published his work in the form of technical papers. His e-mail address is [g.waterworth@lmu.ac.uk](mailto:g.waterworth@lmu.ac.uk)

**NAYEEM N. KARNACHI** was born in Dharwad, India and went to the Karnatak University where he studied Electrical and Electronics Engineering and obtained his degree in 1996. He completed his masters' degree in Control Systems Engineering from Sheffield Hallam University, U.K. in 2001. Currently, he is a part of the research group at the Leeds Metropolitan University, U.K. His interests include adaptive control of nonlinear processes using soft computing techniques, real-time simulation and implementation. His e-mail address is [n.karnachi@lmu.ac.uk](mailto:n.karnachi@lmu.ac.uk)

# A STRUCTURED APPROACH FOR THE IMPLEMENTATION OF DISTRIBUTED MANUFACTURING SIMULATION

Sameh M. Saad, Terence Perera and Ruwan Wickramarachchi  
School of Engineering  
Sheffield Hallam University  
Sheffield, S1 1WB, United Kingdom

## KEYWORDS

Distributed manufacturing simulation, commercial simulation software, IDEF0, Middleware

## ABSTRACT

Manufacturing has been changing from a mainly in-house effort to a distributed style in order to meet new challenges owing to globalization of markets and worldwide competition. Distributed simulation provides an attractive solution to construct cross enterprise simulations to evaluate the viability of the proposed distributed manufacturing enterprises. However, due to its complexity and high cost distributed simulation failed to gain a wide acceptance from industrial users. The main objective of this paper is to address these issues and present a new structured approach to implement distributed simulation with cost effective and easy to implementable tools. A simplified approach for model partitioning for distributed simulation is also included in the proposed approach. The implementation of distributed manufacturing simulation is illustrated with Arena, Microsoft Message Queue (MSMQ) and Visual Basic for Applications (VBA).

## INTRODUCTION

In today's highly competitive world, manufacturing enterprises are confronted with growing competition, the evolution of new markets, more and more sophisticated consumer demand, and increasingly complex global political and economic scenarios. In order to lower the costs, increase profits, reduce product development times, enhance products, and react to environmental changes more positively manufacturing enterprises are moving towards open architectures for integrating their activities with those of their suppliers, customers and partners. In manufacturing, companies may form strategic partnerships for outsourcing some of their operational activities, share resources or joint development of products and services etc., leading to formation of virtual manufacturing enterprises which operate in distributed manufacturing environment. To facilitate the creation of virtual manufacturing enterprises, potential partners must be quickly able to evaluate whether it will be profitable for them to participate in the proposed enterprise. Simulation provides a capability to conduct experiments rapidly to

predict and evaluate the results of manufacturing decisions (McLean and Leong, 2001).

As Law and McComas (1998) noted manufacturing is one of the largest application areas of simulation, with the first uses dating back to at least early 1960s. However, traditional sequential simulation alone may not be sufficient to simulate highly complex Distributed Manufacturing Enterprises (DME). In such situations, distributed simulation provides a promising alternative to construct cross enterprise simulations. The use of distributed simulation allows each partner to hide any proprietary information in the implementation of the individual simulation, simulate multiple manufacturing systems at different degrees of abstraction levels, link simulation models built using different simulation software, to take advantage of additional computing power, simultaneous access to executing simulation models for users in different locations, reuse of existing simulation modes with little modifications etc. (Venkateswaran et al., 2001; McLean and Riddick, 2000; Gan et al., 2000; Taylor et al., 2001). However, Peng and Chen (1996) noted that as a technique, parallel and distributed simulation is not very successful in manufacturing. Most of the distributed manufacturing simulations developed were implemented with either simulation languages or general purpose programming languages such as C++ and Java. This calls for expertise not only in distributed simulation but also in programming too. Moreover, general business community is not very receptive towards distributed simulation due to its complexity, long development times, involvement of steep learning curves, high costs etc.

Another important issue needs to be addressed when designing a distributed simulation is partitioning of the simulation model into sub-models or logical processes (LPs). Efficiency and effectiveness of a distributed simulation system depends on partitioning of the system. Some of the existing approaches require executing of the whole model sequentially in order to collect data before partitioning and mapping carried out based on data collected. However, a simulation is executed in distributed manner because of its inability to run sequentially due to size, complexity, requirements for more computing resources, or need to run geographically distributed manner etc. This creates a dilemma for users especially in business organizations,

who intend to design parallel and distributed simulations.

The objective of this paper is to present a new simplified approach to implement distributed manufacturing simulation (DMS). It includes a simplified approach to model partitioning and mapping, and simulation model development processes for DMS. Instead of implementing the distributed simulation with programming languages, we are proposing to develop the system using commercial simulation software.

**BACKGROUND**

Distributed simulation combines distributed computing technologies with traditional sequential simulation techniques. In recent years, popularity of distributed computing applications increased due to proliferation of inexpensive and powerful workstations, improvements in networking technologies, low cost equipment and incremental scalability. Hence, the use of network of workstations has been evolving into a popular and effective platform for distributed simulation. However, low communication speeds, shortage of network bandwidth and the ever increasing demand for network resources may result slowing down the execution speed of the distributed simulation model. Although the networked workstations are slower than dedicated machines, they may be fast enough and may require much less specialist expertise to put them to use with a fraction of a cost of the price needed for a dedicated parallel processing computer (Cassel and Pidd, 2001).

Throughout the century, the world of manufacturing has changed from a mainly in-house effort to a distributed style of manufacturing. As the term distributed manufacturing implies, DMEs which also known as virtual manufacturing enterprises are ephemeral organizations in which several companies collaborate to produce a single product or product line (Venkateswaran et al., 2001). Participating in this type of collaboration allow partner organizations to use their knowledge, resources and in particular manufacturing expertise to take advantage of new business opportunities and/or gain a competitive advantage that are on a larger scale than an individual partner could handle alone.

**PROPOSED APPROACH**

**Modeling and Model Partitioning for Distributed Manufacturing Simulation**

The degree to which the simulation results are able to characterize the system under study is directly related to the degree the simulation model characterizes the system (Luna, 1993). In order to understand the

problems, requirements and perhaps alternative solutions for many systems especially complex and large ones, it is desirable to build a conceptual model before transforming it into a computer simulation model. Conceptual model is a simulation developer’s way of translating modeling requirements (ie. What to be represented by simulation) into detailed design framework (ie. How it is to be done), from which the software that will make up the simulation can be built (Pace, 1999). Furthermore conceptual model is the ultimate expression of the system functionality and should be the basis for testing and verification and validation procedures (Haddix, 2001).

A conceptual model developed with an appropriate modeling approach and modeling tool facilitates partitioning of the DME model into LPs. Modeling approaches specify the way models are to be developed while Modeling tools provide a standard means of describing and analyzing systems. Hierarchical modeling approach was selected since it provides a way of managing large scale complex systems by considering them as a collection of sub-systems (Kiran, 1998). In a distributed simulation system these sub systems are represented by simulation models that are independently created, modified and saved. Pidd and Castro (1998) also noted that many large systems are inherently hierarchical.

IDEF0 was chosen as the modeling technique for the proposed approach, and it has been widely used in industry due to its user-friendliness, computer support, rigor and conciseness, and well documented rules and procedures (Pandya, 1995; Kateel et al., 1996). Number of authors including Cheng-Leong et al. (1999), Cheng-Leong (1999), Whiteman et al. (1997), Rensburg and Zwemstra (1995) have highlighted the usefulness of IDEF0 as a model representation technique in simulation. Another benefit of using IDEF0 with commercial simulation software is that IDEF0 structure of the model can easily be transformed into simulation model. Figure 1 shows a part of simulation model developed by Arena for an IDEF0 model.

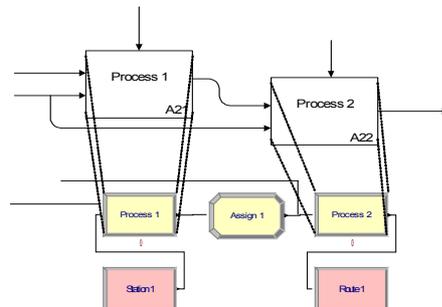


Figure 1: IDEF0 diagram and Arena simulation model

With hierarchical modeling approach and IDEF0 technique, LPs that can function independently could be identified based on interactions between different sections. In the IDEF0 model these interactions are represented by number of lines between boxes that represent different sections of the enterprise.

Once the LPs are identified, they could be validated to make sure that LPs represent individual entities of the enterprise, and the entire enterprise when considered together. The validated LPs could be assigned (mapped) into workstations in a computer network before converting them into computer simulation models and execute as a distributed simulation. Figure 2 shows the proposed approach for modeling, model partitioning and mapping for the distributed manufacturing simulation.

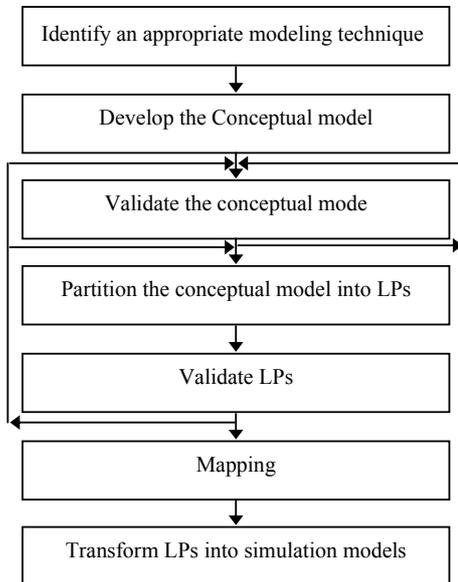


Figure 2: Proposed approach for modeling, model partitioning and mapping

The main difference between existing approaches and the proposed approach is the stage of partitioning carried out in the simulation methodology. According to most of the current approaches partitioning is done only after the system is converted into a computer program using algorithms in order to minimize the communication overheads and optimize the load balance. To simplify the distributed simulation development process it is proposed to partition the conceptual model into LPs and assign them into workstations before transforming LPs into computer simulation models. It is also assumed that networked workstations are freely available to assign LPs and only one LP is mapped into a workstation.

## Development of the Distributed Manufacturing Simulation

Distributed Simulations pose unique synchronization constraints due to their underlying sense of time. When the simulation state can be simultaneously changed by different processes, actions by one process can affect actions of another (Nicol, 1993). In order to make sure that each LP processes arriving messages in their timestamped order and not in their real time arriving order, individual simulation models needed to be synchronized. This requirement is referred to as local causality constraint (Fujimoto, 1990). Optimistic protocols are implemented by saving simulation state at different points of time and rolling back to a previous time point if local causality constraint is violated. If a programming language is used to develop the simulation, then state saving mechanism can be integrated into the distributed simulation engine itself. Since commercial simulation packages generally do not allow saving simulation state at different time points and rolling back to previous time points, it was decided to employ conservative protocol to synchronize the distributed manufacturing simulation. Conservative approaches strictly impose the local causality constraint and guarantee that each model will only process events in non-decreasing timestamp order. Determining a value for lookahead is one of the most important and difficult aspect of conservative protocol. However, it was assumed that minimum-processing times (which can be used as lookahead values) for LPs can be calculated. An approximate synchronization mechanism, especially suitable for distributed manufacturing applications has been proposed by Saad et al. (2003).

In order to synchronize and pass parameters, simulation models need to communicate with each other. Communication methods provided by operating systems often require complex programming. In a distributed simulation, middleware provides simple and reliable solution for this problem. Middleware is a class of software designed to help manage the complexity and heterogeneity inherent in distributed system. It contains a set of enabling services which allow multiple processes running on one or more computers to interact across a network. Analysis of past literature reveals number of attempts to simulate distributed manufacturing systems and supply chains using tools such as HLA, CORBA and GRIDS (see Venkateswaran et al., 2001, Taylor et al., 2001; Gan et al., 2000; McLean and Riddick, 2000). For the proposed approach, Microsoft Message Queue (MSMQ), a Message Oriented Middleware was selected to link simulation models.

As MSMQ is integrated into newer versions of Windows operating systems and available as an additional component for Windows NT, 98 and 98, it

provides a cost effective solution for message passing. MSMQ interacts with simulation model through an application program interface (API). Arena simulation software was used in this study as commercial simulation software to demonstrate the implementation. However, other commercial simulation software such as Automod, Promodel, Witness etc. can also be used for this purpose. Both Arena and MSMQ support Visual Basic for Applications (VBA) and C++. Since, programming of Arena with VBA is more straightforward than with C++, it was decided to use VBA to develop the API. VBA also offers a programming environment similar to popular Visual Basic programming language.

API developed for MSMQ could send messages containing parameters obtained from simulation model to a queue in the same computer or directly to another remote computer. API that resides in the remote computer extracts these messages from the queue and passes the parameters to the simulation model (Figure 3).

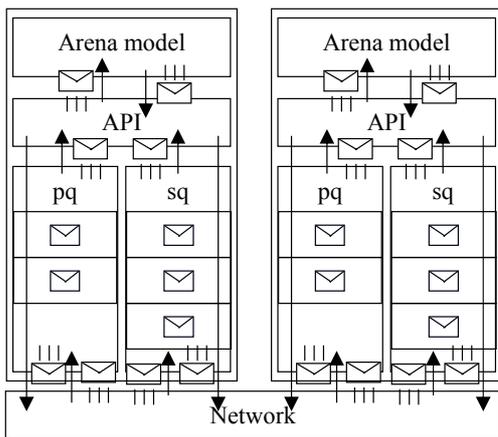


Figure 3: MSMQ, API and Arena

**ILLUSTRATION**

In order to illustrate the development of distributed simulation with Arena, MSMQ and VBA, following brief case study is presented.

A, B and C firms are proposing to form a distributed manufacturing enterprise (Figure 4) to produce a new product called XYZ. Firm A is to produce and process parts X and Z. Part Z is sent to Firm C and part X is sent to Firm B which possesses an expensive processing unit for further processing. Firm B is also to produce part Y and assemble Parts X and Y together to form component XY which is then sent to Firm C for further processing and final assembly. At Firm C, component XY and part Z are to be further

processed and assembled together to produce product XYZ. In addition to processing of parts X, Y, Z, component XY and product XYZ, three firms also produce their own products independently. Parts are to be passed in batches of 1000s and transfer time from one firm to another firm was assumed as 10 hours. Before committing on the DME, firms want to evaluate the feasibility in terms of capacity utilization and how the proposed venture affects their existing operations. As firms are reluctant to pass information of their processes to other firms, it was agreed to develop 3 models separately and run them in a distributed simulation environment.

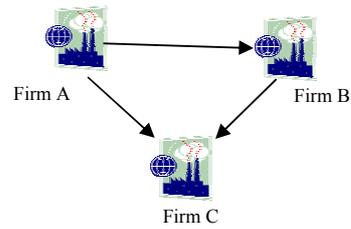


Figure 4: A model for distributed manufacturing

Three Arena simulation models were independently developed, verified and validated for firms A, B and C. Entities created within models were used instead of inputs from other models for B and C. Then B and C were modified by replacing the 'Create module' with a 'Create block' and adding a 'VBA block' just before the 'Dispose module' (Figures 5 and 6). 'Create block' can be used to release entities into the model which created by API of the model to represent output from A and/or B. Two MSMQ queues were created in each workstation, one to accept inputs (pq) and the other to synchronize (sq) the distributed simulation. Once batch of 1000 units were processed at A or B, the batch goes through a VBA block and API written in VBA sends a message to destination model (Figure 6). When a message is reached its destination queue, it is processed automatically with built-in 'qevent' event. Once a message comes to 'pq', 'qevent' creates an entity and schedules to release it after 'transfer time' at 'Create block'. The 'Separate module' adds additional 999 units to make a batch of 1000 which was passed as output from previous model (Figure 5). At model C, output from A and B can be identified by message label. Figures 7 and 8 show sample code written for VBA block and 'qevent' respectively.

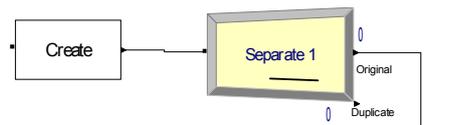


Figure 5: Adding output received from other models as input

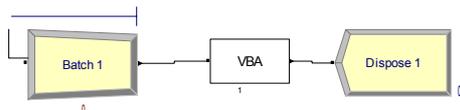


Figure 6: Passing output to destination model VBA block

```
Private Sub VBA_Block_1_Fire()
  Dim qinfo As MSMQQueueInfo
  Set qinfo = New MSMQQueueInfo
  qinfo.FormatName = "DIRECT = OS:ENG-4130-10
  \private$\mbq"
  Dim qQueue As MSMQQueue
  Set qQueue = qinfo.Open(MQ_SEND_ACCESS,
    MQ_DENY_NONE)
  Dim qMsg As MSMQMessage
  Set qMsg = New MSMQMessage
  qMsg.Label = "A"
  qMsg.Body = "1000"
  qMsg.Send qQueue
  qQueue.Close
End Sub
```

Figure 7: Sample code of VBA block

```
Sub qEvent_Arrived(ByVal Queue As Object, ByVal
  cursor As Long)
  Dim vEntityIndex As Long
  Dim vPictureIndex As Long
  Dim aQueue As MSMQQueue
  Set aQueue = Queue
  Dim qMsg As MSMQMessage
  Set qMsg = aQueue.Receive(, , 0)

  vPictureIndex =
  ThisDocument.Model.SIMAN.SymbolNumber
  ("Picture.Package")
  vEntityIndex = ThisDocument.Model.SIMAN
  .EntityCreate
  Call ThisDocument.Model.SIMAN.EntitySetPicture
  (vEntityIndex, vPictureIndex)

  If qMsg.Label = "A" Then
    Call ThisDocument.Model.SIMAN
    .EntitySendToBlockLabel(vEntityIndex
    , 10, "CreateBlockA")
  Else
    Call ThisDocument.Model.SIMAN
    .EntitySendToBlockLabel(vEntityIndex,
    10, "CreateBlockB")
  End If

  aQueue.EnableNotification qEvent
End Sub
```

Figure 8: Sample code to create and schedule an entity to represent input (of model C)

## DISCUSSION

This paper presented a simplified approach to implement a distributed manufacturing simulation. Although a distributed manufacturing application was used to illustrate the implementation, the proposed approach may be able to use in other application areas too. The main benefit of this research is the simplified

approach employed when developing the distributed simulation models using commercial simulation software and, connecting and running them in distributed environment with MSMQ and VBA. Not only Arena, but also other simulation software packages such as Promodel, Automod and Witness can be used to implement the simulation. Furthermore, simulation models developed with different simulation software can be connected together and run as a distributed simulation as long as they support either VBA or C++. Cost involved can be kept low as no additional costs involved with middleware and application program interface (API), provided workstations are running on a windows operating system. The proposed approach also encourages reusability of existing simulation models. Existing simulation models developed for traditional sequential simulation require only minor modifications to adopt for a distributed simulation.

It is expected that this simplified approach may address criticisms made against distributed simulation because of its complexity to develop and implement, higher costs involved, need for more expertise etc. Working of the simulation model can be animated easily as commercial simulation software is used for implementation. Animation may play a very effective role in convincing the benefits of simulation to non simulation users such as managers, workers etc.

Main shortcoming of the proposed approach is the sacrifice made in performance of the distributed simulation mainly in terms of speedup. It may also not be feasible to employ the proposed approach to implement highly complex systems such as telecommunications systems, computer networks, logic circuits etc. However, applications which are not executed in distributed manner only to gain speedups, and applications that specifically require executing in distributed simulation environment are ideally fit for the approach we presented in this paper.

## CONCLUSIONS

The proposed approach addressed some of the criticisms leveled against distributed simulation with cost effective and simplified implementation approach for distributed manufacturing simulation.

Performance comparisons between distributed simulation implemented using proposed approach and conventional approaches may provide an opportunity to fully understand the benefits and the shortcoming of our work. Unlike sequential simulation, output from a distributed simulation can be obtained for individual models as well as for the whole system under investigation. Since outcome of the simulation effort depends on the output of the simulation, it may worth

investigating strategies to identify and generate output locally at individual models and for the entire distributed simulation system.

## REFERENCES

- Cassel, R.A. and M. Pidd. 2001. "Distributed discrete event simulation using the three-phase approach and Java." *Simulation: Practice and Theory* 8, No.8, 491-507.
- Cheng-Leong, A., K.L. Phengm and G.R.K. Leng. 1999. "IDEF\*: a comprehensive modeling methodology for the development of manufacturing enterprise systems." *International Journal of Production Research* 37, No.17, 3839-3859.
- Cheng-Leong, A. 1999. "Enactment of IDEF models." *International Journal of Production Research* 37, No.15, 3383-3397.
- Fujimoto, R.M. 1990. "Parallel discrete event simulation." *Communications of the ACM* 33, No.10, 30-53.
- Gan, B.P., L. Liu, S. Jain, S.J. Turner, W. Cai, and W. Hsu. 2000. "Distributed supply chain simulation across enterprise boundaries." *Proceedings of the 2000 Winter Simulation Conference* 1245-1251.
- Haddix, F. 2001. "Conceptual modeling revisited: A developmental model approach for modeling & simulation." *Proceedings of the 2001 Simulation Interoperability Workshops*
- Kateel, G., M. Kamath and D. Pratt. 1996. "An overview of CIM enterprise modeling methodologies." *Proceedings of the 1996 Winter Simulation Conference* 1000-1007.
- Kiran, A.S. 1998. "Hierarchical modeling: A simulation based application." *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics* 3079-3089.
- Law, A.M. and M.G. McComas. 1998. "Simulation of manufacturing systems." *Proceedings of the 1998 Winter Simulation Conference* 49-52.
- Luna, J.J. 1993. "Hierarchical relations in simulation models." *Proceedings of the 1993 Winter Simulation Conference* 132-137.
- McLean, C. and F. Riddick. 2000. "The IMS mission architecture for distributed manufacturing simulation." *Proceedings of the 2000 Winter Simulation Conference*, 1539-1548.
- McLean, C. and S. Leong. 2001. "The role of simulation in strategic manufacturing." *International Working Conference on Strategic Manufacturing* 239-250.
- Nicol, D.M. 1993. "The cost of conservative synchronization in parallel discrete event simulations." *Journal of the Association for Computing Machinery* 40, No.2, 304-333.
- Pace, D.K. 1999. "Development and documentation of a simulation conceptual model." *Proceedings of the 1999 Simulation Interoperability Workshops*.
- Pandya, K.V. 1995. "Review of modeling techniques and tools for decision making in manufacturing management." *IEE Proceedings of Science, Measures and Technology*.
- Peng, C. and F.F. Chen. 1996. "Parallel discrete event simulation of manufacturing systems: A technology survey." *Computers and Industrial Engineering* 31, No.1/2, 327-330.
- Pidd, M. and B.R. Castro. 1998. "Hierarchical modular modeling in discrete simulation." *Proceedings of the 1998 Winter Simulation Conference*, 383-390.
- Resenburg, A.V. and N. Zwemstra. 2002. "Implementing IDEF techniques as simulation modeling specifications." *Computers and Industrial Engineering* 29, No.1-4, 467-471.
- Saad, S.M., T. Perera, and R. Wickramarachchi. 2003. "Simulation of distributed manufacturing enterprises: A new approach." *To be published in Proceedings of the 2003 Winter Simulation Conference*.
- Taylor, S.J.E., R. Sudra, T. Janahan, G. Tan and J Ladbrook. 2001. "Towards COTS distributed simulation using GRIDS." *Proceedings of the 2001 Winter Simulation Conference* 1372-1379.
- Venkateswaran, J., M.Y.K. Jafferli and Y Son. 2001. "Distributed simulation: An enabling technology for the evaluation of virtual enterprises." *Proceedings of the 2001 Winter Simulation Conference* 856-862.
- Whitman, L., B. Huff and A Presley. 1997. "Structured models and dynamic systems analysis: The integration of the IDEF0/IDEF3 modeling methods and discrete event simulation." *Proceedings of the 1997 Winter Simulation Conference*.

## AUTHOR BIOGRAPHIES

**SAMEH M. SAAD** (BSc, MSc, PhD, CEng, MIEE, ILTM) is a Reader in Advanced Manufacturing Systems and Enterprise modeling and management and Postgraduate Course Leader at the Systems and Enterprise Engineering Division, one of the three divisions in the School of Engineering, Sheffield Hallam University, United Kingdom. Dr Saad's research interests revolve around aspects of design and analysis of manufacturing systems, production planning and control, systems integration, reconfigurable manufacturing systems, manufacturing responsiveness, enterprise modeling and management and next generation of manufacturing systems. He has published over 55 articles in various national and international academic journals and conferences. His contact email address is <s.saad@shu.ac.uk>

**TERRENCE PERERA** (BSc, PhD) is Professor and Head of Enterprise and Systems Engineering at School of Engineering, Sheffield Hallam University. He also leads the Systems Modeling and Integration Research Group. His current research interests include the implementation, integration and practice of virtual modeling tools within all industrial sectors. His email address is <t.d.perera@shu.ac.uk>

**RUWAN WICKRAMARACHCHI** is a PhD student at Sheffield Hallam University, United Kingdom. He received his MPhil and BSc degrees from University of Cambridge, United Kingdom and University of Kelaniya, Sri Lanka respectively. His main research interest focused on distributed enterprise simulation with emphasis on distributed manufacturing applications. He can be contacted by <w.ruwan@shu.ac.uk>

# INTELLIGENT DYNAMIC SIMULATION OF A SOLAR COLLECTOR FIELD

Esko K. Juuso

Control Engineering Laboratory

Department of Process and Environmental Engineering

FIN-90014 University of Oulu, Finland

Mail: esko.juuso@oulu.fi

## KEYWORDS

Solar power plant, dynamic modelling, intelligent simulation environments, non-linear models, linguistic equations, fuzzy set systems

## ABSTRACT

Linguistic Equation (LE) modelling approach has various applications in non-linear multivariable systems. Insight to the process dynamic operation is maintained, and automatic generation of systems, model-based techniques and adaptation techniques can be applied in developing and tuning systems for process modelling and control. The multimodel LE approach provides a compact modelling of more or less smooth input-output dependencies. The overlapping operating areas are obtained by fuzzy clustering. The Fuzzy-ROSA method (FRM) serves for a data-based rule generation to model a given input-output dependency and is efficient for modelling complicated local non-linear structures. These properties are combined in a hybrid data-based modelling concept applied to dynamic simulation of a solar collector field. The hybrid fuzzy LE simulator was tested in data-based modelling of dynamic behaviour of a solar collector field. The new adaptive controller tuned with this technique has reduced considerably temperature differences between collector loops. Efficient energy collection is achieved even in variable operating condition.

## INTRODUCTION

In intelligent control design, hybrid techniques combining different modelling methods in a smooth and consistent way are essential for successful comparison of alternative control methods. Switching between different submodels in multiple model approaches should be as smooth as possible. For slow processes, predictive model-based techniques are necessary at least on the tuning phase. Adaptation to various non-linear multivariable phenomena requires a highly robust technique for the modelling and simulation.

Dynamic simulators based on Linguistic Equations are continuously used in development of multilayer linguistic equation controllers, in which the basic PI type LE

controller is extended with a working point controller and a module for asymmetry handling and braking. This new type of controller was first implemented on a solar collectors field in a solar power station at *Plataforma Solar de Almeria* [JBL97, JBV98]. Adaptive set point procedure and feed forward features have later been included for avoiding overheating. The present controller takes also care of the actual set points of the temperature [JV03].

The multilevel linguistic equation controller has been applied in the control of the burning end of the lime kiln [JJA01]. The multilevel LE controller has been in on-line use in an industrial lime kiln for more than four years, and the experiences are very similar to the simulation results [Juu98]. Smooth production rate changes are found to be preferable also in the real process. The robust dynamic simulator based on Linguistic Equations is an essential tool in fine-tuning of all these controllers.

## SOLAR POWER PLANT

The aim of solar thermal power plants is to provide thermal energy for use in an industrial process such as seawater desalination or electricity generation. If such plants are to provide a viable, cost effective alternative to more polluting forms of power production, they must achieve this task despite fluctuations in their primary energy source, the sunlight. In addition to seasonal and daily cyclic variations, the intensity depends also on atmospheric conditions such as cloud cover, humidity, and air transparency. The purpose is not to maintain a constant supply of solar produced thermal energy in spite of the disturbances. Rather the aim of the control scheme should be to regulate the outlet temperature of the collector field in order to supply steam to the turbine in a range as constant as possible despite disturbances, changes of the solar radiation, ambient temperature, inlet oil temperature etc.

This is beneficial in a number of ways. Firstly, it collects any available thermal energy in a usable form, i.e. at the desired temperature, which improves the overall system efficiency and reduces the demands placed on auxiliary equipment as the storage tank. Secondly, the solar field is maintained in a state of readiness for the resumption of full-scale operation when the intensity of the

sunlight rises once again; the alternative is unnecessary shutdowns and start-ups of the collector field, which are both wasteful and time consuming. Finally if the control is fast and well damped, the plant can be operated close to the design limits thereby improving the productivity of the plant.

All the experiments were carried out in the *Acurex Solar Collectors Field of the Plataforma Solar de Almeria* located in the desert of Tabernas (Almeria), in the south of Spain. The *Acurex field* supply thermal energy (1 MW) in form of hot oil to an electricity generation system or a Multi-Effect Desalination Plant. The solar field consists of parabolic-trough collectors [JBL97, JBV98]. Control is achieved by means of varying the flow pumped through the pipes during the plant operation. In addition to this, the collector field status must be monitored to prevent potentially hazardous situations, e.g. oil temperatures greater than 300 °C. When a dangerous condition is detected software automatically intervenes, warning the operator and defocusing the collector field.

Trial and error type controller tuning does not work since the operating conditions cannot be reproduced. The dynamic of the process depends on the general field operating conditions and characterised by the following aspects:

- Time varying transport delay depends on the manipulated variable (oil flow rate).
- The dynamics, in particular high frequency peaks in the frequency response of the plant, is difficult to model.
- The plant has a non-linear behaviour, and therefore linearised models depend on operating point.
- The solar radiation acts as a fast disturbance with respect to the dominant time constant of the process.

Test campaigns cannot be planned in detail because of changing weather conditions. Usually, test campaigns include step changes and load disturbances. Weather conditions take care of irradiation disturbances. As the process must be controlled all the time, modelling is based on process data from controlled process.

Operating conditions cannot be reproduced and weather conditions have seasonal differences. Therefore, dynamic simulators are needed in controller design and tuning. Conventional mechanistic models do not work: there are problems with oscillations and irradiation disturbances. For non-linear multivariable modelling on the basis of data with understanding of the process there are two alternatives: fuzzy set systems and linguistic equations.

## DATA-BASED MODELLING

For the modelling of technical complex processes one is often restricted to only with data-based methods since a complete mathematical process description is not practicable with justifiable expenditure. Various modelling approaches try to combine the advantages of the physical and data-driven modelling techniques, e.g. parameters for mechanistic models are approximated by

black-box techniques. Since the identification is on a practical level only for linear systems, a lot of work with local linear models is needed.

Intelligent methods have extended the toolbox to hybrid, semi-mechanistic or grey-box modelling. Fuzzy clustering is an extension of fuzzy knowledge based systems to data-driven techniques. Neuro-fuzzy modelling and identification techniques include fuzzy-logic-based methods to neural computing. Linguistic equations have close links to both fuzzy set systems and neural networks.

## Data Preprocessing

Direct measurement value is not always best one to be used in modelling. Sometimes moving variances, standard deviations or value ranges are more informative for the phenomena. Also moving skewness and kurtosis can be obtained. Selecting appropriate window for this moving statistics is also an important decision. Trend removal on the basis of the user defined window (moving average or median) can be included to the preprocessing if the variation around the trend is important for the modelling.

The FuzzEqu Toolbox developed in Matlab-Simulink environment provides tools for experimenting with different methods and windows [Juu00]. The data set is updated only after accepting the operation. Several statistical operations can be applied also sequentially to the data, e.g. after trend removal the resulting data can be analysed other statistical methods. For small systems, delays can be taken into account by moving the values of input variables correspondingly.

## Linguistic Equation Approach

Linguistic equation models consist of two parts: *interactions* are handled with linear equations, and nonlinearities are taken into account by *membership definitions* [Juu99]. The basic element is a compact equation

$$\sum_{j=1}^m A_{ij} X_j + B_i = 0, \quad (1)$$

where  $X_j$  is a linguistic level for the variable  $j$ ,  $j = 1 \dots m$ . Linguistic values very low, low, normal, high, and very high correspond to integer numbers -2, -1, 0, 1 and 2. The direction of the interaction is represented by interaction coefficients  $A_{ij}$ . The bias term  $B_i$  was introduced for fault diagnosis systems. Linguistic equations can be used to any direction. The directions of interaction are usually quite clear in this kind of small systems: only the absolute values of the coefficients need to be defined.

The membership definition is a non-linear mapping of the variable values inside its range to a certain linguistic range, usually  $[-2, 2]$ . The mapping is represented with two monotonous, increasing functions, which must overlap in the center at the linguistic value 0. In the present system, these functions are second order polynomials. Coefficients are extracted from data or defined on the basis of expert knowledge.

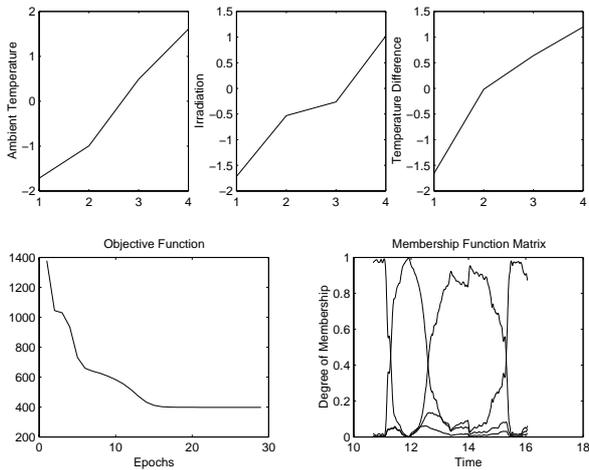


Figure 1: Four operating areas obtained by Fuzzy C-Means Clustering.

Modelling with linguistic equations has following stages:

- Membership definitions are generated by using pre-processed data.
- Linguistic relations are obtained by non-linear scaling.
- Linguistic equations are generated from the scaled data denoted as linguistic relations.
- Selecting equations from alternatives is based either on the overall fit or on the prediction performance.
- Tuning modifies membership definitions, linguistic equations or both to improve fitting to the training data.

Real-valued approach is now the main method in applications because of efficient tuning techniques. A neural network based tuning can be done for selected variables. A recently generated genetic tuning method can handle several variables at a time by varying parameters of membership definitions.

The modelling technique can be extended to several equations as well, e.g. by using Takagi-Sugeno (TS) type fuzzy models together with ANFIS method for development of local linear models for different operating areas. As *LE* models are non-linear, also these local models are non-linear.

For model development, the training data consist of several data sets. Some overlap of the working point areas is automatically introduced when process data is used. Fuzzy C-Means Clustering is used for finding these overlapping operating areas (Figure 1). Alternatively the operating areas can be obtained by Self-organizing Maps as well (Figure 2). The delays are taken into account in tuning. The interaction matrix is normally the same for all working areas, which is quite reasonable since the directions of interactions do not change considerably between different working points. The differences between the models are handled with membership definitions.

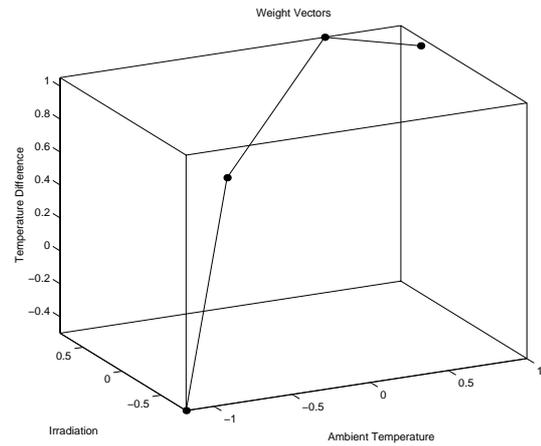


Figure 2: Four operating areas obtained by a Self-organizing Map.

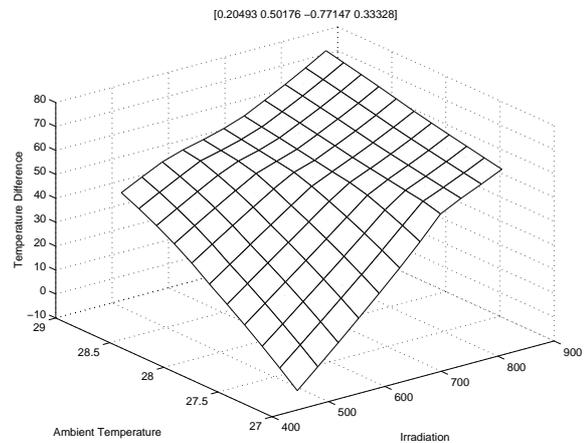


Figure 3: LE model for working point variables.

The working point variables already define the overall normal behaviour of the solar collector field. The model shown in Figure 3 has a quite high correlation to the real process data (Figure 4). The differences have a clear relation to operating conditions, e.g. oscillatory behaviour is a problem when the temperature difference is higher than the normal. Separate dynamic models (Figure 5) are needed to capture the dynamic behaviour in different operating conditions (Figure 1).

The *FuzzEqu* toolbox contains tools for all the development and tuning stages described above [Juu00]. It also contains routines for modifying membership definitions interactively to adapt the models to changing operating conditions and routines for building *LE* systems from large fuzzy systems including various ruleblocks implemented in FuzzyCon or *Matlab(r)* FuzzyLogic Toolbox. Other fuzzy modelling approaches can be used as channels for combining different sources of information. Fuzzy systems as *Dora for Windows* blocks can be included in *Simulink* environment.

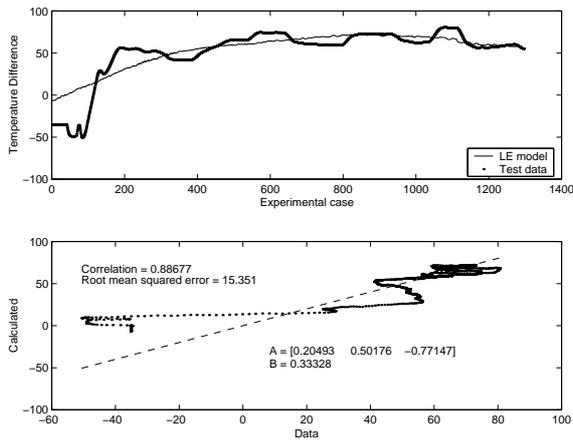


Figure 4: LE model for working point variables.

## Dynamic LE modelling

Dynamic fuzzy models can be constructed on the basis of state-space models, input-output models or semi-mechanistic models [Juu99]. In the state-space models, fuzzy antecedent propositions are combined with a deterministic mathematical presentation of the consequent. The most common structure for the input-output models is the NARX /Non-linear AutoRegressive with eXogenous input) model which establishes a relation between the collection of past input-output data and the predicted output:

$$y(k+1) = F(y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)), \quad (2)$$

where  $k$  denotes discrete time samples,  $n$  and  $m$  are integers related to the systems' order. Multiple input, multiple output (MIMO) systems can be built as a set of coupled multiple input, single output MISO models.

Effective delays depend on the working conditions (process case); e.g. the delays are closely related to the production rate in many industrial processes. Initial estimates of the delays can be developed by correlation analysis, but similarities detected by the correlation analysis can be accidental in some cases. The delays should be assessed against process knowledge, especially if normal on-line process data is used [Juu99]. An appropriate handling of delays extends the operating area of the model considerably.

The basic form of the *LE* model is a static mapping, and therefore dynamic *LE* models could include several inputs and outputs originating from a single variable [Juu99]. However, rather simple input-output models, e.g. the old value of the simulated variable and the current value of the control variable as inputs and the new value of the simulated variable as an output, can be used since nonlinearities are taken into account by membership definitions. Comparisons with different parametric models, e.g. autoregressive moving average (*ARMAX*), autoregressive with exogeneous inputs (*ARX*), *Box-Jenkins* and Output-Error (*OE*), show that the performance improvement with additional values is negligible.

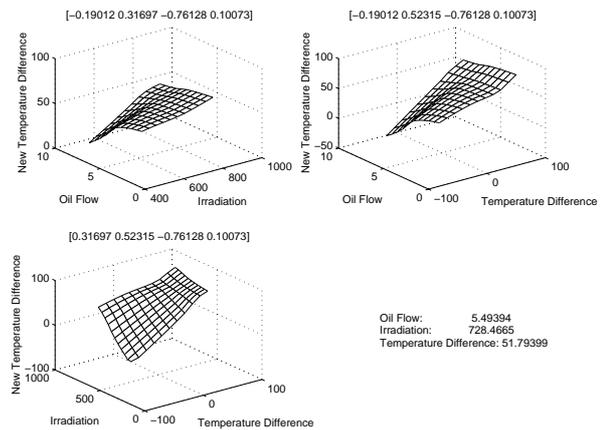


Figure 5: A dynamic LE model for temperature difference.

In the single model approach, also variables affecting to the working point of the model are included to the model. In small models, all the interactions are in a single equation. For larger models, the equation system is a set of equations where each equation describes an interaction between two to four variables. The development work starts with an automatic generation of membership definitions, which are then used in generation of interaction alternatives. Any equation can be rejected or modified on the basis of expert knowledge before or during the tuning phase.

The dynamic model of the solar collector field is based on a compact LE model for the temperature difference is shown in Figure 5. The new temperature difference between the inlet and outlet depends on the irradiation, oil flow and previous temperature difference. This model provides the driving force for the simulator, and the speed of the change depends on the operating conditions.

A multimodel approach based on fuzzy LE models has been developed for combining specialised submodels. The approach is aimed for systems that cannot be sufficiently described with a single set of membership definitions because of very strong non-linearities. Additional properties can be achieved since also equations and delays can be different in different submodels. In the multimodel approach, the working area defined by a separate working point model. The submodels are developed by the case-based modelling approach.

Various modelling methodologies have been compared for both dynamic and working point models in the FuzzEqu Toolbox. Feedforward neural networks, radial basis networks and ANFIS method provide better fitting to the training data but generalisation is worse in these systems as they include parts which are not consistent with process operation. Each LE submodel could include several alternative equations combined with fuzzy logic but these models have same overfitting problems. According to the tests with real process data, the fuzzy LE system with four operating areas is clearly the best overall model.

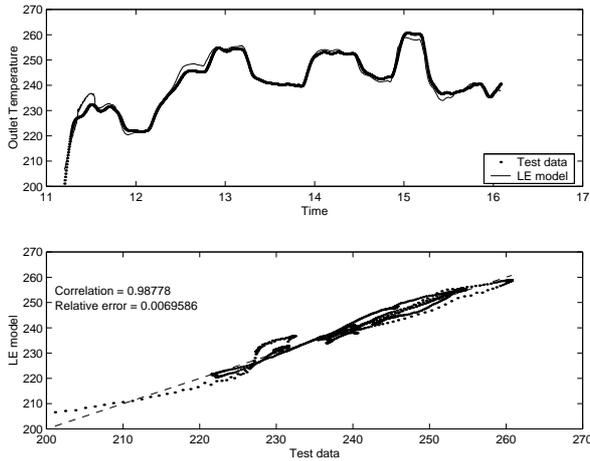


Figure 6: Simulation results of the LE model.

## Fuzzy–ROSA Method (FRM)

The Fuzzy–ROSA<sup>1</sup> method (FRM) serves for a data-based generation of fuzzy rules which model a given input–output dependency. The basic idea of the FRM is to apply a relevance test to single fuzzy rules to assess their ability to describe a relevant aspect of the system under consideration. This reduces the problem of finding a good rule base to the problem of finding single relevant rules. On the other hand, since each rule with high relevance is supposed to express an important aspect of the system, such rules are meaningful by themselves, which leads to more transparent and comprehensible rule bases.

The FRM uses generalising (incomplete) rules, which consist of a varying number of linguistic statements (combination depth) in the premise. If there are fewer statements than input variables, one rule covers several linguistic input situations. The rule generation process is divided into four main steps [JSSK00]. There are alternative strategies available for each step, so that FRM can be adapted to different application requirements (e.g., for modelling, classification, approximation or prediction) and problem sizes (e.g., numbers of variables, linguistic values and data sets).

## COMBINED APPROACH

Linguistic equation (LE) models provide a good overall behaviour in different operating conditions (Figure 6). Oscillations are well represented, and the temperature is on an appropriate range in the case of irradiation disturbances. However, some problems have been detected in extensive comparisons with process data: there is a shift in temperature level for some operating conditions. In some conditions the shift is positive and in some conditions negative. The present model needs

<sup>1</sup>RuleOrientated Statistical Analysis

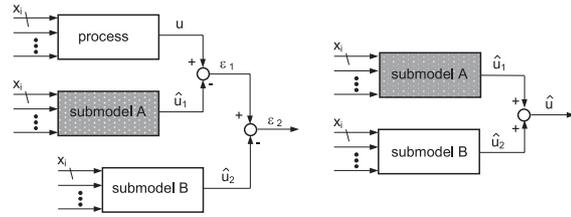


Figure 7: Cascaded modelling (left) and resulting model (right)

improvements also for load disturbances.

Flexible fuzzy models generated with the Fuzzy–ROSA method provided additional tools for these situations [JSSK00]. These fuzzy models are useful in handling special situations in limited operating range. However, functional relationship between the output variable and the input variable are partly smooth and partly complicated non-linear [JSSK00]. A *straightforward* application of the FRM may result in a high number of rules or an undesired competition between locally and globally acting rules.

To overcome these problems, a cascaded rule generation (Figure 7 left) has been proposed in [Kie99]: A first pass generates a submodel A for the more or less smooth global structure, a second pass generates a submodel B for the remaining usually locally complicated error  $\varepsilon_1$  between submodel A and the real process. The final model is the superposition of the submodels A and B (Figure 7 right).

Since smooth dependencies can be described easily by simple equations, we take the Linguistic Equations (LE) as a promising approach for the generation of a compact submodel A. Since complicated local structures are efficiently detected by the FRM, we apply the FRM for the generation of submodel B. Thus the cascaded modelling with the LE and FRM combines the advantages of both methods, which can result in a considerable improvement of the quality of the resulting final model. Feasibility of the combined LE–FRM approach was demonstrated by applying it to a solar power plant [JSSK00].

## Dynamic LE Simulator

The dynamic model for temperature difference between inlet and outlet temperatures of the collector field has been developed for the solar collector field. The simulator includes models for different operating conditions. Smooth transitions between the models are based on fuzzy logic. Working point model is defined by the irradiation and the difference between the inlet and outlet temperatures.

According to the test results at the *Plataforma Solar de Almeria*, the dynamic simulator of the solar collector field represents very accurately the field operation

(Figure 6). In steady weather conditions, the present simulator operates within 2 degrees centigrade. Oscillatory conditions are also handled correctly. The simulator is based on the multimodel LE approach with four specialised LE models developed for different operating conditions. The simulator moves smoothly from start-up mode via low mode to normal mode. Later the field visits shortly in high mode and low mode before returning to low mode in the afternoon.

Correlation between the calculated and measured temperatures is very high for all time period: 0.992 for the whole day, 0.988 for the normal operating area and 0.961 for the start-up period. The relative errors are 2.9 percent for the whole day, 0.7 percent for the normal operating area and 16.8 percent for the start-up period [JSSK00].

For start-up the dynamic LE simulator requires improvement since the process changes considerably during the first hour [JSSK00]. The simulator underestimates the temperature growth because of unevenness of the oil flow. For radiation disturbances, the LE simulator operates quite well: the temperature is on the appropriate range all the time and the timing of the changes is very good. The simulator can also handle correctly oscillations although the dynamics depends on the operating point. A considerable temperature shift can be seen some periods. The LE model should be improved in these areas. Another alternative is to combine LE modelling and fuzzy modelling.

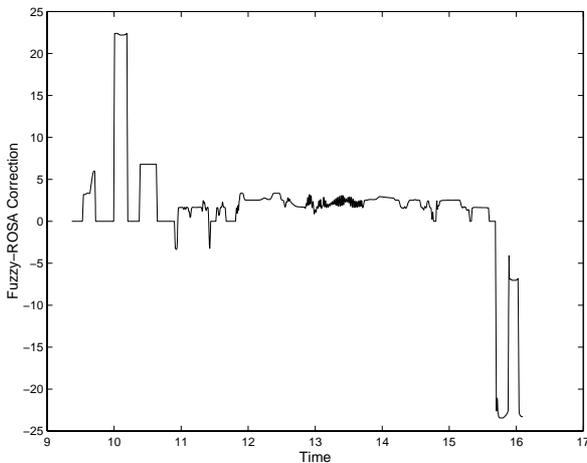


Figure 8: Model obtained with the FRM.

### Fuzzy-ROSA Modelling

As described in Section we apply the FRM to model the remaining error of the LE-model. The learning data consist of simulation results of four selected days. In a preliminary feature selection process, we found the following seven input variables to be strongly correlated to the output variable: *daytime*, *oil flow*, *corrected radiation (moving average)*, *ambient temperature*, *delayed inlet temperature*, *delay* and *working point*.

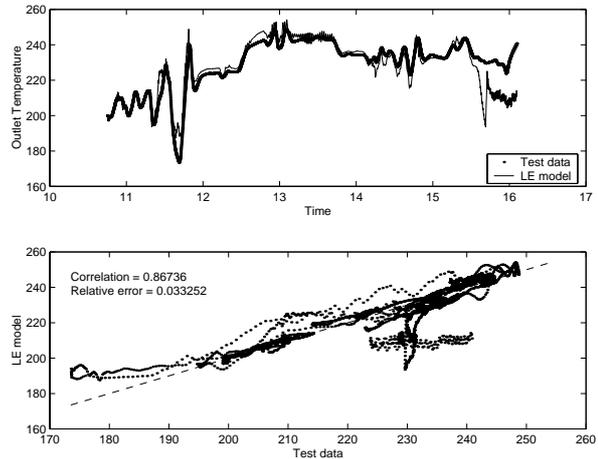


Figure 9: Simulation results of the the combined LE-FRM model.

In order to reduce the computational effort we use only these input variables for the fuzzy-modelling. The membership functions of the input and output variables are extracted knowledge based by considering their distributions. This leads to seven linguistic expressions for the input variables and nine for the output variable. For rule generation we apply a complete search considering all rules which refer to not more than four input variables (maximum combination depth of four). As the data are disturbed strongly by stochastic influences, we choose the Mean Value Based Index as test- and rating strategy.

This approach leads to a fuzzy rule base of 173 relevant rules, which model the remaining error of the LE-model. In a second step we apply the optimising conflict reduction. The final rule base consists of 77 rules and the modelling error on learning data is reduced to 2.7 degrees centigrade in the combined approach.

### DYNAMIC SOLAR PLANT SIMULATOR

The fuzzy model was combined with the LE-model and used in a close-loop operation in the dynamic simulation [JSSK00]. This serves for validation as the dynamic simulation generates situations (data sets) which differ from the learning data sets.

Fuzzy error model is included to the estimation of the new temperature difference goal. The fuzzy system developed with Fuzzy-ROSA method<sup>2</sup> was included as a *Dora for Windows 6.2* block to the *Simulink* simulator. The fuzzy system produces additional temperature difference (Figure 8) in the dynamic model. For the clear day, there is hardly any correction, which means that the model is not much improved. Important is that the Fuzzy-ROSA method does not develop any rules for the conditions where it cannot improve performance. Correlation between the calculated and

<sup>2</sup>Obtained with the WINROSA 2.0 software tool: <http://esr.e-technik.uni-dortmund.de/winrosa/winrosa.htm>.

measured temperatures was about the same as for the LE model: 0.991 for the whole day, 0.981 for the normal operating area and 0.960 for the start-up period. The relative errors are 3.0 percent for the whole day, 0.8 percent for the normal operating area and 17.0 percent for the start-up period.

For the period after radiation disturbances (Figure 9), the combined model improves the result considerably from the results of the LE model. Correlation between the calculated and measured temperatures depends now on the operating conditions: 0.964 for the whole day, 0.967 for the normal operating area, 0.969 for the start-up period and 0.176 for the load disturbance in the end of the day. The relative errors are 6.6 percent for the whole day, 1.8 percent for the normal operating area, 18.9 percent for the start-up period and 8.9 percent for the load disturbance.

The dynamic LE simulator is a practical tool in the controller design. The LE controller tuned with this simulator combines smoothly various control strategies into a compact single controller. Control strategies ranging from smooth to fast are chosen by setting the working point of the controller. The controller takes care of the actual set points of the temperature. The operation is very robust in difficult conditions: startup and set point tracking are fast and accurate in variable radiation conditions; the controller can handle efficiently even multiple disturbances. Adaptive set point procedure and feed forward features are essential for avoiding overheating. The new adaptive technique has reduced considerably temperature differences between collector loops. Efficient energy collection was achieved even in variable operating condition [JV03].

## CONCLUSIONS

The combined modelling approach improves performance of the dynamic simulator. The smooth and fairly accurate overall behaviour is achieved with Linguistic Equations. The result is further improved by fuzzy systems generated for special situations with Fuzzy-ROSA method. The combined dynamic model is feasible for controller tuning but more special cases need to be analysed to expand the operating area of the dynamic simulator. Fuzzy clustering methods provide feasible techniques for selecting new cases for modelling from the extensive experimental data. The new adaptive control technique has reduced considerably temperature differences between collector loops. Efficient energy collection was achieved even in variable operating condition.

## References

- [JBL97] E. K. Juuso, P. Balsa, and K. Leiviska. Linguistic Equation Controller Applied to a Solar Collectors Field. In *Proceedings of the European Control Conference -ECC'97, Brussels, July 1 - 4, 1997*, volume Volume 5, TH-E I4, paper 267 (CD-ROM), 6 pp., 1997.
- [JBV98] E. K. Juuso, P. Balsa, and L. Valenzuela. Multilevel linguistic equation controller applied to a 1 mw solar power plant. In *Proceedings of the ACC'98, Philadelphia, PA, June 24-26, 1998*, volume 6, pp. 3891-3895. ACC, 1998.
- [JJA01] M. Järvensivu, E. Juuso, and O. Ahava. Intelligent control of a rotary kiln fired with producer gas generated from biomass. *Engineering Applications of Artificial Intelligence*, 14(5):629-653, 2001.
- [JSSK00] E. K. Juuso, D. Schauten, T. Slawinski, and H. Kiendl. Combination of linguistic equations and the fuzzy-rosa method in dynamic simulation of a solar collector field. In *Proceedings of TOOLMET 2000 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 13-14, 2000*, pp. 63-77, Oulu, 2000. Oulun yliopistopaino.
- [Juu98] E. K. Juuso. Intelligent dynamic simulation of a lime kiln with linguistic equations. In *ESM'99: Modelling and Simulation: A tool for the Next Millenium, 13th European Simulation Multiconference, Warsaw, Poland, June 1-4, 1999*, pp. 395-400, Delft, The Netherlands, 1998. SCS.
- [Juu99] E. K. Juuso. Fuzzy control in process industry: The linguistic equation approach. In H. B. Verbruggen, H.-J. Zimmermann, and R. Babuska, editors, *Fuzzy Algorithms for Control, International Series in Intelligent Technologies*, pp. 243-300. Kluwer, Boston, 1999.
- [Juu00] E. K. Juuso. Linguistic equations for data analysis: FuzzEqu toolbox. In *Proceedings of TOOLMET 2000 Symposium*, pp. 212-226, Oulu, 2000. Oulun yliopistopaino.
- [JV03] E. K. Juuso and L. Valenzuela. Adaptive intelligent control of a solar collector field. In *Proceedings of Eunite 2003 - European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, July 10-11, 2003, Oulu, Finland*, pp. 26-35. Wissenschaftsverlag Mainz, Aachen, 2003.
- [Kie99] H. Kiendl. Design of advanced fuzzy systems. In *Proceedings of TOOLMET'99 Symposium-Tool Environments for Intelligent Systems*, pp. 57-76, Oulu, Finland, 1999.

## ACKNOWLEDGEMENTS

All the experiments described in this paper were carried out within the projects "Innovative Training Horizons in Applied Solar Thermal and Chemical Technologies"(C.N: ERBFMGECT950023) and "Improving Human Potential Programme-Access to Research Infrastructures Activity" (C.N: HPRI-CT-1999-00013) supported by the DG XII. The combined modelling research was sponsored by the Deutsche Forschungsgemeinschaft (DFG), as part of the Collaborative Research Center Computational Intelligence (531) of the University of Dortmund.

# AN HLA FEDERATION FOR EVALUATING MULTI-DROP STRATEGIES IN LOGISTICS

*Roberto Revetria*

MISS Genoa – University of Genoa  
Via Opera Pia, 15  
16145 Genoa Ge, Italy  
e-mail: revetria@itim.unige.it

*P.E.J.N. Blomjous, S.P.A. van Houten*

Faculty of Technology, Policy and Management – Delft University of Technology  
Jaffalaan 5, 2528 BX  
Delft, The Netherlands  
e-mail: s.a.vanhouten@student.tbm.tudelft.nl

## Abstract

Distributed Simulation has proven to be very effective in modeling complex system. Since High Level Architecture has turned to a strict DoD requirement several civil domain application have been experienced. Despite the high potential of the methodology the Industrial Community is facing some resistance in implementing HLA due to the lack of directly applicable tool. In this paper authors proposes a federation for supporting Supply Chain Management by integrating a set of Arena™ based simulator able to support the optimization of a Multi Drop Delivery problem using the Capacitated Routing Vehicle Problem (CRVP). The paper outline the main issues of the Federation Design and Implementation as well as a Real Life Application of the proposed methodology.

## Introduction

The recent developments in Manufacturing have boost up the practice of outsourcing in which suppliers are continuously specializing and improving in order to meet the market changes. This point is not only a way to reduce production costs but is also a common practice to increase the flexibility to the market: new ideas and proof of concept can be obtained from the “external world” and transformed into real product. The “externalize & specialize” approach has turned in to several Spin Off Projects in which Companies can gain business performance from previously critical division. A production line that is underused can be turned into an interesting business unit by transforming it into a separate company and open to the open market (i.e. PUMA Experience for Gas Turbine Power Plants Maintenance). This issue requires a efficient level of control that is becoming extremely complex for highly distributed Manufacturing Systems, in which Simulation is largely used due to the extremely non linear nature of the problems. Simulation, here, is often

used to improve process. Until now all the members of a supply chain had to simulate separately their processes and information are not shared among the Supply Chain Partners But for taking advantage from the proposed methodology a full scale simulator model may be used in order to to build a model that resembles reality more effectively. The High Level Infrastructure (HLA) is a standard framework that supports simulations composed of different simulation models. From now on the different models, parts of the total model are called federates. In order to design a simulation composed of different federates on different computers it is necessary to connect them together and establish a communication protocol, this is done by HLA that clearly separate Simulation from Communication Process. As the HLA supports interoperability, the different federates must communicate with each other via the Run Time Infrastructure (RTI) according to the 10 HLA Rules.. The application of the HLA has many advantages since it offers interoperability, encourages reusability and makes it possible to use confident information in models without the necessity of being visible to other partners in the supply chain. In this way other partners don't have access to confident information and can't use it in a strategic way. The other partners will only see the results of simulation runs/steps and not the data behind the results. Because every partner now is more willing to use confidential information the total quality of a model of a supply chain increases and hence the benefits for the partners. Furthermore, because each partner builds it's own module, it is much easier to keep modules up-to-date both in term of data (i.e. directly from ERP). The University of Genoa was particularly involved in the Web Integrated Logistic Designer Project (WILD I & WILD II). These projects involve the development of a federation composed by simulators and dynamic programming systems (i.e. Nash Equilibrium Negotiation). The WILD Federation reproduces the supply chain and supports on-line the distribution among Suppliers, Main Contractors,

Outsourcer and was successfully tested for an Supply Chain in the aerospace industry.

To make the HLA accessible also for Commercial Off of the Shelves Simulators (COTS) the HLA Operative Relay Using Sockets (HORUS™) has been developed. The HORUS™ manages the communication between the different simulation models and the HLA. To take advantage of interoperability and reusability existing federates, implemented in various languages, can be integrated.

### High Level Architecture for Logistic

The High Level Architecture (HLA) is a standard framework that supports simulations composed of different simulation modules. Many complex simulations involve a combination of simulations of several different types of systems with different aspects of the total environment to be simulated. Unfortunately it is often necessary to make extensive modifications to adapt an existing simulation model so that it can be integrated into a new combined simulation (federation). In some cases it may prove easier to implement a completely new simulation of a system than to modify an existing one. In other words, traditional simulation models often lack two desirable properties: reusability and interoperability. Reusability means that modular simulation federates can be reused in different simulation scenarios and applications. Interoperability means that the reusable simulator can be combined with other federates into newly create exercises. without the need for re-coding. The HLA is an architecture that makes it possible for different modules (federates) to communicate with each other. A group of federates forms a federation.

### HORUS™ for Integrating Arena in HLA

To make the HLA architecture more accessible to COTS, the MISS University of Genoa developed a Middleware called HORUS that acts as a Delegate Simulator. A schematic overview of the HLA and HORUS is presented in figure 1.

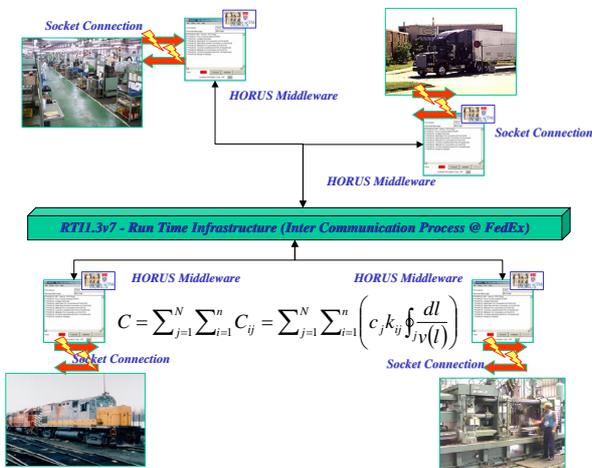


Figure 1 – HORUS Architecture

Each federate has its own HORUS, in which a federate registers the attributes it would like to publish (to other federates) and subscribe to (from other federates). The HORUS is responsible for receiving information from HLA and sending information to federates via RTI. After it receives information, it passes it on to the HLA architecture. The HLA sends it back and the HORUS delivers it to a federate. The HORUS is used to make it easier to make a connection between the HLA and modules. One of the benefits is the ease to let the HLA know what kind of information a federate is interested (Declaration Management) in and what sort of information it wants to publish. In this way the RTI solves all the issues about the information routing. In this way it is possible to construct several parts of the model in different federates and to it in a simplified way. Possibilities for reusability are improved because the Building Block (Federates) have general processes, which can be used in other simulations without having to heavily re-code the model. When a federate contains too much specific processes, is not generally useful to be used in another simulation exercise. designed for general supply chains. Not only does this not improve reusability due to the greater complexity of a module, also more communication is needed to tailor the specific processes.

### A Case Study

In order to practically demonstrate the proposed approach a Federation was designed in order to model a Supply Chain devoted to improve a Multi Drop Delivery Process. For this purpose Supply Chain was divided in three different modules. These are a Supplier, a Carrier and a Main Contractor Federate. In this project, cargo forms a bundle of loads. The Main Contractor sends orders to the Supplier. Besides ordering goods from the Supplier, the Main Contractor takes care of receiving cargo and has a terminal process to unload the transporters. When the Supplier has produced goods it sends a transportation request to the Carrier to transport the cargo. The requests have an attribute item, which indicates the number of loads (cargo) for a destination. The Supplier module has a production process (here the ordered goods are produced) and a terminal process (to load cargo on a transporter). The Carrier module provides the transporters and drivers. This module takes care of the transporters, drivers and of course the transport itself. The modular design makes it able to add extra modules (e.g. a maritime Carrier module) later on and to attach our modules to other modules. The HLA provides the possibility to connect Federates, but does, of course, not guarantee the usability of the information exchanged. In this project information is exchanged by means of formatted strings. In order to improve the data exchange String can be re-formatted in XML™ and furthermore parsed by the Simulator itself. To represent the objects described in this chapter, entities are used. In this way it is possible to design the supply chain in Arena™. The entities are rebuilt in the

receiving modules based on strings composed of attributes. The entities chosen are orders, requests, cargo and transporters. The following general attributes, which are necessary for a supply chain, are chosen:

- Origin: is unique identifier of the pick up location.
- Destination: is the unique identifier of the delivery location.
- Transporter: is the unique code that identify the Carrier into the System
- Items: is the dimension of the batch that is shipped in this delivery.
- Kind: is the identification of the typology of the cargo (i.e. bulk, parcels, gas, etc.), it is used to identify the best suitable vehicle.
- Id: is the unique id for every order, it is used for the cross reference on the Federation.
- Time Delivered: is the time when cargo is delivered to the Main Contractor

Transporters have some general attributes and some extra attributes that have to do with the architecture of the modules. For the transporters the following extra attributes are chosen: Last\_Destination, Status and Costs. Last\_Destination indicates the destination a transporter last visited. Status indicates whether a transporter is in use or not and costs are the transportation costs for the cargo. These are the attributes specific enough for a supply chain and general enough to couple modules to other modules as other modules designed for a supply chain will use these kind of attributes.

In the supply chain there are three terminals and several destinations taken into account. The terminals serve as home stations for the transporters. The terminals also have the cargo the Supplier produces in stock. Unlike an hub-and-spoke network is there no transport of cargo to the terminals (hubs). There is only one way transport of cargo from the terminals to the destinations. So the terminals act as distribution centers. Main Contractor serve as a system that orders goods for multiple locations (see Figure 2). The Supplier serves as a system that resembles the production process for all orders the Main Contractor sends to the Supplier.

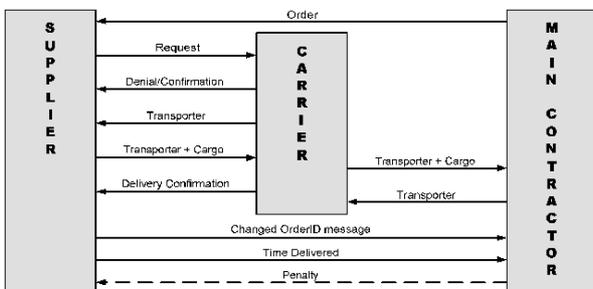


Figure 2: Federation Architecture

The Carrier serves as a system that picks up cargo from the terminals and transports it to the different eleven destinations (the locations). From the terminals the cargo is distributed to the different destinations. The Main Contractor sends orders to the Supplier. These

orders have one of the terminals as origin. The destination can be every location, except its own origin.

In the Supplier four objects arrive, namely orders from the Main Contractor and empty transporters, a delivery confirmation and an answer to the transportation request from the Carrier. The Supplier also sends some objects: loaded transporters, transportation requests and time delivered.

### Suppliers' Logic

When the Supplier receives orders from the Main Contractor, it starts to produce the ordered goods.

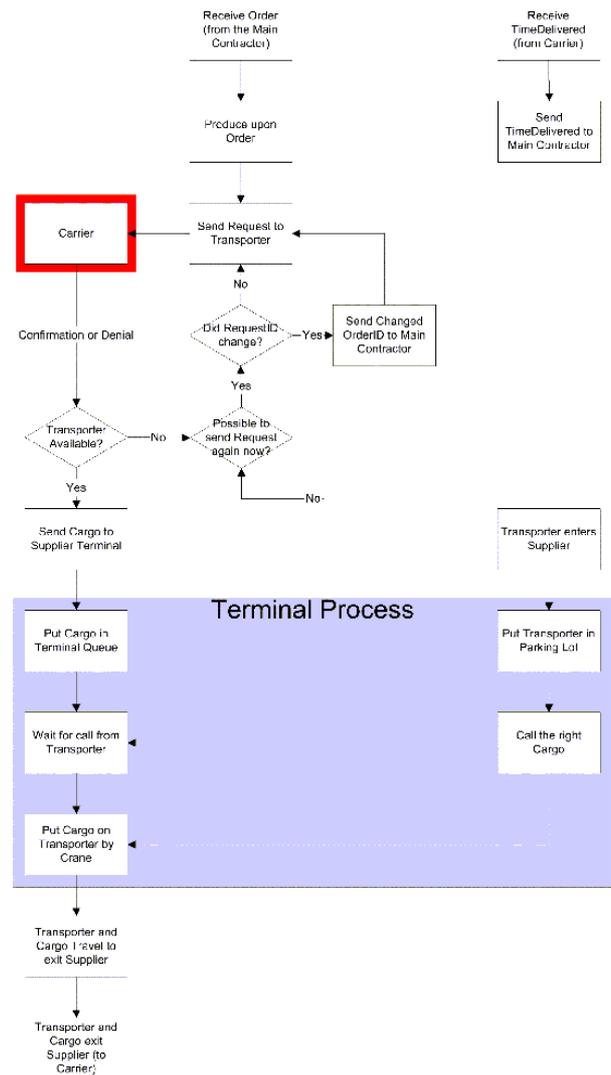


Figure 3 - The Suppliers' Logic Diagram

When the production is finished, requests are created. A request represents a number of loads (cargo) for a destination. These are sent to the Carrier. After a while the Carrier sends a message whether it is possible or not to transport this cargo. So the carrier decides whether it is possible to transport cargo to a destination (see Figure 3). Depending on the answer of the Carrier the Supplier sends the cargo to the Supplier terminal or, when the Carrier gives a negative answer, the Supplier sends the request for the same cargo again in the next time step. When a request can not be sent

immediately (due to the unavailability of a transporter), the Supplier keeps on trying to send denied requests to the Carrier. Whether denied requests are sent immediately depends on the number of other requests waiting for transportation. The Supplier decides whether it is possible to immediately send denied requests again or not. When a transporter arrives at the Supplier it goes to a central parking lot. After a while it calls its cargo, which it has to transport, from the terminal queue. This is done when the crane in the terminal is free, because there are no other transporters to serve. After the transporter has called its cargo, it travels to the crane. The called cargo also goes to the crane. The crane puts the cargo on the transporter. When the transporter is fully loaded, it leaves the terminal and travels, with its cargo, to the exit of the Supplier and exits. Its next event is entering the Carrier. In the Supplier, beside empty transporters, also delivery confirmations arrive from the Carrier. When a transporter has delivered its cargo and returns to the Carrier it sends a delivery confirmation to the Supplier.

### Carriers' Logic

In the Carrier three objects arrive, namely requests and transporters with cargo from the Supplier and empty transporters from the Main Contractor. The Carrier also sends messages, confirmations or denials and delivery confirmations to the Supplier (See Figure 4).

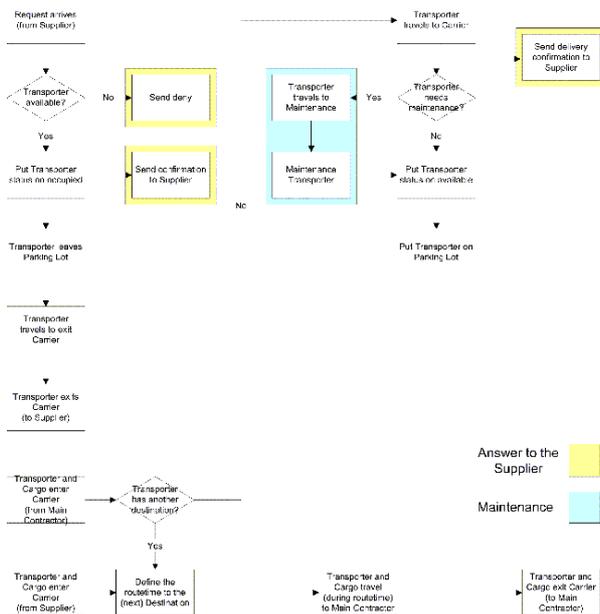


Figure 4: Carriers' Logic Diagram

When a request arrives from the Supplier, the Carrier checks whether a transporter is available or not for the requested destination. Here the Carrier decides whether requests for transportation are denied or approved. If a transporter is available the status is set on occupied. If no transporter is available the Carrier sends a negative answer (denial) to the Supplier. Otherwise a positive answer (confirmation) is send to the Supplier. When in the Carrier transporters arrive with their cargo from the Supplier, they travel (during their RouteTime)

in the Carrier to the Carrier exit and enter the Main Contractor.

When transporters arrive at the Carrier from the Main Contractor first of all a delivery confirmation is sent to the Supplier. Next the decision is made if they have another destination besides returning to the nearest terminal. If another destination is found, the transporter travels to its this destination. If there is no destination it has to visit, it travels to its nearest terminal. At the same time the decision is made whether or not the transporter needs some maintenance. If it needs some maintenance, the transporter travels to the maintenance. Otherwise the transporter travels to the parking lot corresponding to its nearest terminal. When a transporter arrives at the maintenance, it undergoes repair for a certain amount of time. After this period the transporter travels to the parking lot.

### Main Contractors' Logic

The Main contractor sends two objects. These are orders and transporters (with or without cargo). The orders are created and sent to the Supplier. When the orders are sent, their time and some other characteristics are written in a sheet and an expected time of arrival of the goods is estimated (see Figure 5). In the Main Contractor three objects arrive, namely transporters with their cargo from the Carrier, a delivery confirmation and a Changed OrderID message from the Supplier.

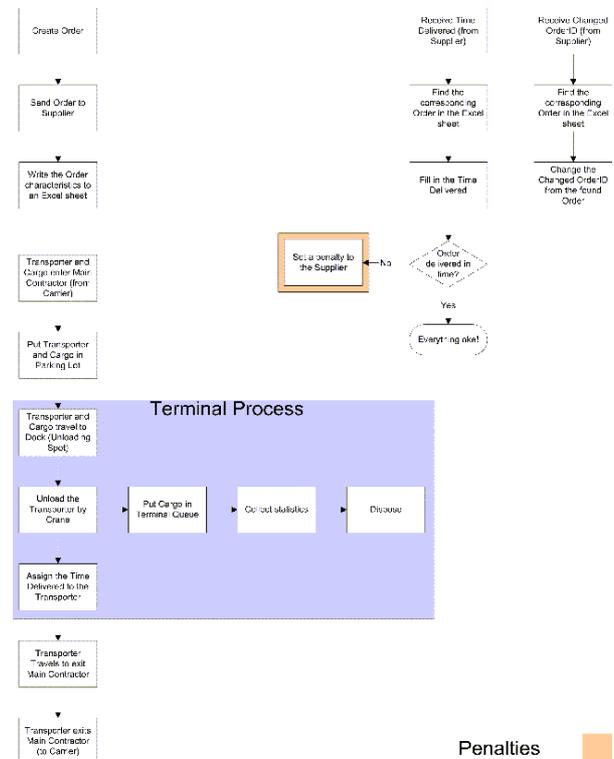


Figure 5: Main Contractors' Logic Diagram

In the Main Contractor's transporters arrive from the Carrier carrying its cargo. When they arrive at the Main Contractor, they go to a central Parking Lot, where they will wait. When the crane is available, a transporter travels to the terminal. After a transporter

arrives at the terminal, a crane unloads it. The cargo is put in the Main Contractor's terminal, where some statistics are collected and after that the cargo is disposed. As soon as a transporter is unloaded, its *Time\_Delivered* attribute is set and it travels to the Main Contractor's exit, exits the Main Contractor and enters the Carrier. When the cargo is delivered, the transporter sends a delivery confirmation, based on the *Time\_Delivered*, to the Supplier after it entered the Carrier. After a while the Main Contractor receives a delivery confirmation from the Supplier including the OrderID, so the Main Contractor can match this with its original send orders. Together with this time and the expected delivery time, a calculation is made whether a penalty should be given to the Supplier. When a Changed OrderID message is received in the Main Contractor, this is matched with the original orders and the OrderID of the corresponding order(s) is changed in the changed OrderID. This is necessary regarding the delivery confirmation in a later stadium. When two orders are combined into one order, one of the OrderID's is changed. To make it possible to know in a later stadium the order (with the changed OrderID) is delivered it is necessary to change its OrderID into the new one.

#### *Terminals' Logic*

As the terminal process is depending on two objects, namely cargo and transporters, both must be available for the terminal process. For this matter it is not enough to just model the terminal process as a stochastic delay. As there are different modules, the terminal processes are assigned to modules. As cargo can not transport itself in reality, transporters are moved to other modules. For this reason the terminal processes are assigned to the Supplier and the Main Contractor.

The Carrier receives requests from the Supplier and sends transporters to the Supplier. The Supplier receives them on its own internal terminal queue. After loading, the Supplier returns the transporters and their cargo to the Carrier. The same goes for the relation between the Carrier and the Main Contractor. In this situation it concerns an unloading process.

#### **VV&A: Learned Lessons**

Verification is the process of determining that a model implementation accurately represents the developer's conceptual description and specifications. Validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. Due to the exploring character of this project, many problems were already found and solved during the design phase. Verification of distributed models is more difficult than verification of non-distributed models (ignoring differences in logic and size of the models). This because when something goes not as expected, the source of the problem is hard to find. It is possible that a problem in one module is caused by an error in another module. In this way it is not possible to isolate

problems easily and in many cases multiple (long lasting) runs are necessary to find a problem and to solve it.

#### **Conclusions**

Experimental Campaign is under development and preliminary results show great potential of the outlined methodology.

Due to the extreme long runtime that experiments take an replicated half-fractional factorial  $2^{2-1} \times 2$  design was used, the obtained Response Surface proven that a such federation could be used in order to provide a meta model able to improve the performance of a Production Planner.

Commercial stand alone modeling software used to build HLA Supply Chain Federates have shown long simulation run time that can reduce significantly the use of the simulator on the other hand the use of regression meta models must be considered as a performing alternative.

#### **References**

1. Bruzzone A., Mosca R., Revetria R. (2001) "Supply Chain Management over the web in Aerospace Industry by using Simulation: WILD", Proceedings of Virtuality2001, Turin
2. Bruzzone A., Roberto Mosca, Flavio Tonelli, R.Revetria, Viganò G., Diglio G. (2001) "Advanced Issues in Distributed Verification and Validation Process for Supply Chain
3. Bruzzone A.G, Mosca R., Revetria R., Tonelli F. (2002) "Sistemi Integrati di Gestione Avanzata per la Quick Response della Catena di Fornitura", Proceedings of XXIX Convegno Nazionale ANIMP, Sorrento, Italy October
4. Bruzzone A.G., Giribone P., Revetria R. (2002) "Genetic Algorithms and Simulation for Aftersales Supply Chain Re-Engineering Process", Proceedings of MIC2002, Innsbruck, February 18-21
5. Bruzzone A.G., Giribone P., Revetria R., Simeoni S. (2001) "Potential of Artificial Intelligence Techniques and Simulation in Improving Container Terminal Performances", Proceedings of POMS2001, Orlando, March 31 April 2
6. Bruzzone A.G., Mosca R., Revetria R. (2001). "Gestione Integrata di Sistemi Produttivi Interagenti: Metodi Quantitativi Avanzati per la Quick Response", DIP Genova, Italy, ISBN: 88-900732-0-9
7. Bruzzone A.G., Mosca R., Revetria R. (2002) "Cooperation in Maritime Training Process using Virtual Reality Based and HLA Compliant Simulation", Proceedings of XVIII International Port Conference, Alexandria Egypt, January 27-29
8. Bruzzone A.G., Mosca R., Revetria R. (2002) "Gestione della Supply Chain Mediante Federazione di Simulatori Interagenti: Compendium", DIP University of Genoa, Genoa, December ISBN 88-900732-1-7

9. Bruzzone A.G., Mosca R., Revetria R. (2002) "Supply Chain Management Dynamic Negotiation using Web Integrated Logistics Designer (WILD II)", Proceedings of MAS2002,
10. Bruzzone A.G., Mosca R., Revetria R. (2002) "Web Integrated Logistics Designer and Intelligent Control for Supply Chain Management", Proceedings of Summer Computer Simulation Conference 2002, San Diego, July
11. Bruzzone Agostino, Mosca R., Revetria R. (2001) "Web Integrated Logistics Designer: A HLA Federation Devoted to Supply Chain Management", Proceedings of Summer Computer Simulation Conference, Orlando, July 15-19
12. Giribone P., Revetria R., Mantero E., Diglio G., Viganò G., (2000) "A General Purpose System for Supporting Transportation System Re-Engineering", Proceedings of HMS2000, Portofino, October 5-7
13. Revetria R., Blomjous P.E.J.N., Van Houten S.P.A (2001) "Simulating Supply Chain Logistics with Arena™ and The High Level Architecture" DIP Tech. Report, Savona January
14. Revetria R., Tucci M. (2001) "Different Approaches in Making Simulation Languages Compliant with HLA Specification" Proceedings of the Summer Computer Simulation Conference, Orlando FL, July 14-19



# **SIMULATION IN ELECTRONICS, COMPUTERS AND TELECOM**



# A New Approach to the Simulation of HIPERLAN Wireless Networks

G. I. Papadimitriou\*, T. D. Lagkas\*, M. S. Obaidat\*\*, and A. S. Pomportsis\*

\*Department of Informatics, Aristotle University  
Box 888, 54124, Thessaloniki, Greece

\*\* Corresponding Author:

M. S. Obaidat, Dept. of Computer Science  
Monmouth University  
West Long Branch, NJ 07764, USA  
E-mail: obaidat@monmouth.edu

## KEYWORDS

HIPERSIM, HIPERLAN, modeling and simulation, WLAN, wireless, sense range, performance evaluation.

## ABSTRACT

This paper presents a new simulation mechanism that is used by the simulator "HIPERSIM," which was developed to examine the behavior of the HIPERLAN wireless networks. The ETSI HIPERLAN and the IEEE 802.11 are the most popular WLAN standards. The lack of HIPERLAN specialized simulators and the need for an accurate simulation engine have led to the development of HIPERSIM. The latter is a simulation environment for the HIPERLAN Type 1 networks which uses an exhaustive simulation engine in order to simulate accurately most of the important features of a HIPERLAN wireless network. Specifically, it fully simulates the complicated EY-NPMA MAC protocol of HIPERLAN, which is based on active signaling, hidden nodes, packet forwarding mechanism, power saving process, among others. A rather original characteristic of HIPERSIM is the fact that it distinguishes between the communication range and the sense range of a node. The main focus of this work is to provide a simulation mechanism appropriate for wireless networks, and especially for the HIPERLAN WLANs. The HIPERSIM results show that the HIPERLAN protocol is effective and suitable for the wireless environment. Probably there could be some improvement in order to avert the collisions close to the receiver.

## 1. INTRODUCTION

The increasing interest in wireless networks has led to the development of some standards that try to find their way in the market. This technology seems to be mature, but it still tries to meet the increasing demands and needs of new applications. WLAN standards demand continuous improvement in order to support QoS successfully. The recent applications for wireless networks are quite demanding, because they involve synchronized transmission (e.g. voice and video transmission) and reliability. Beside the need for improvement of the WLAN standards, it is necessary

for the manufacturers to evaluate the different WLAN solutions that are offered today. Because of the above mentioned reasons, the need for suitable WLAN simulation tools has now arisen.

The two most popular WLAN standards are the 802.11 (IEEE) and the HIPERLAN (ETSI). Most of the researchers use general-purpose network simulators with the appropriate modules for the wireless topology. Some of them are forced to create their own simulation tools in order to analyze some limited features of a wireless network. Especially for the HIPERLAN networks, there are very few suitable simulation environments. This paper examines the existing WLAN simulation methods in general, and presents the HIPERSIM simulation environment. HIPERSIM is a simulator specialized in HIPERLAN networks simulation, it is fully parameterized, and it supports most of the features of HIPERLAN protocol and wireless topology. One of the differences between the simulation method that HIPERSIM uses and the classical simulation methods is the fact that HIPERSIM uses a time-based simulator in order to simulate accurately and in an exhaustive way the complicated Elimination Yield Non-pre-emptive Priority Multiple Access (EY-NPMA) MAC protocol of HIPERLAN. Also, HIPERSIM distinguishes between the communication range and the sense range of a node, which is rather original.

## 2. WLAN SIMULATION

The basic principles of a WLAN simulation do not differ significantly from the principles of a wired LAN simulation. So, the primary simulation entities (e.g. node, medium, buffer, and packet) remain the same, and the results mainly concern the network throughput and the average packet delay under various conditions. However, the wireless nature of a WLAN has some special characteristics that need extra analysis and emphasis.

In a WLAN, it is usually assumed that the communication medium, that is the air, is common for all the nodes, like a bus-network. The most simulation environments adopt this assumption. However, this is not completely accurate. In a wired bus-network, all the nodes that are connected to the cable have the same view

of the medium status. This is not happening in a wireless network. In a WLAN, every node has its own view of the medium status, which depends on the range of its antenna. When two nodes of a WLAN are not able to communicate directly with each other, then they are called hidden nodes. The “hidden nodes” issue is very important for the WLAN operation that is why most of the simulation environments take into account the hidden nodes. HIPERSIM goes a step further by distinguishing between the communication range and the sense range of a node.

Specifically, a pair of hidden nodes can be either in sense range or not in the HIPERSIM simulation environment. This issue is analyzed later on. Also, when simulating a WLAN, the packet forwarding mechanism of the specific protocol affects significantly the overall network performance, so it deserves further analysis. Another feature that is simulated by HIPERSIM is the power saving mechanism. The nodes of a WLAN, like HIPERLAN, are usually mobile devices with a limited battery life. Since the antenna of a node consumes a great amount of energy when transmitting or receiving, the simulation of the power saving mechanism provided by the protocol is quite interesting.

Most of the network simulators, like OPNET, are event-based. This means that the network operation is divided into a number of discrete events. There is an event list where all the generated events are stored. When an event is processed, new events are generated and they are added to the list. Then, the event that takes place earlier than the others is selected from the list, and the simulation clock is updated. The simulation stops when the simulation time is over or another termination condition is true. In Figure 1, the flowchart of a typical event-based simulator is shown (Obaidat et al. 2003; Nicopolitidis et al. 2003; Sadiku and Ilyas 1995).

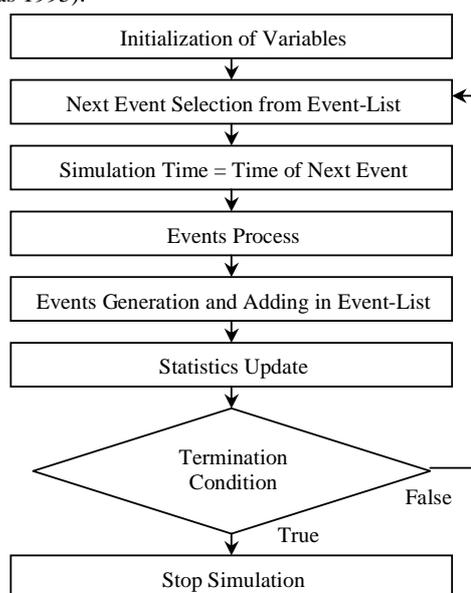


Figure 1. Typical Event-based Simulator Flowchart

The MAC protocols of the IEEE 802.11b and the HIPERLAN Type 1 standards are based on the classic carrier sense multiple access (CSMA) protocol, since a node senses the carrier before attempting to transmit. However, these WLAN MAC protocols have a lot of enhancements in order to be suitable for the wireless environment, thus they differ significantly from the classic CSMA that is used in the wired LANs. The event-based simulation models are the most popular schemes, but their great disadvantage is the fact that they demand simplifications of the system operation in order to distinguish discrete events. HIPERSIM, on the other hand, implements an exhaustive time-based simulation model in order to simulate accurately the complicated EY-NPMA MAC protocol of the HIPERLAN standard. Also, this simulation method avoids simplifications that concern the wireless topology, like the assumption that all nodes have the same view of the carrier status. In Figure 2, the flowchart of a typical time-based simulation model is presented (Obaidat et al. 2003; Nicopolitidis et al. 2003; Sadiku and Ilyas 1995).

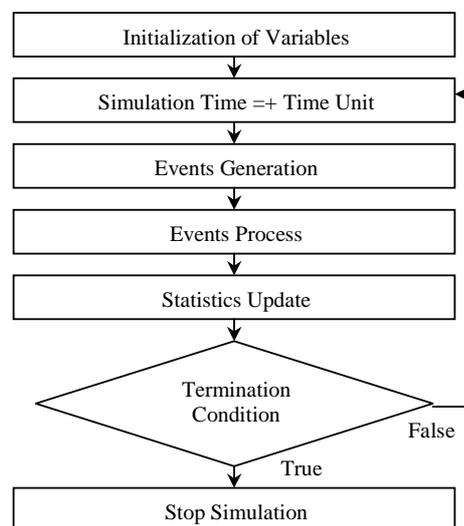


Figure 2. Typical Time-Based Simulator Flowchart

### 3. THE HIPERLAN NETWORK

HIPERLAN (High Performance Radio Local Area Network) is a standard for wireless LANs that was defined by the European Telecommunications Standards Institute [ETSI]. HIPERLAN is a rather short range WLAN (~50m), it supports slow moving stations (1.4 m/s), and can be of infrastructure or ad hoc type. Its high transmission rate is 23.529Mbps, and it operates at the 5GHz band.

An important characteristic of HIPERLAN is that it can support a variety of services. It combines asynchronous communication, such as file transfer, with time bounded communication, such as voice and video transmission. This is due to the fact that the Medium Access Control protocol supports Quality of Service (QoS), aided by the Channel Access Mechanism (CAM) that assigns the packet priorities.

### 3.1. The Priority Mechanism

The QoS support is based on two mechanisms: the user priority assigned to a packet and the lifetime of the latter. The user application sets the user priority of a packet at the value low or high. The value of the user priority and the packet lifetime give the CAM priority. In Figure 3, the CAM priority values are shown according to the user priority and the residual lifetime. The highest CAM priority is the one with the smallest value (ETSI 1998; Jacquet et al. 1996b; FU et al. 1996).

Notation:

ML: MPDU Lifetime

RML: Residual MPDU Lifetime

UP: User Priority (low = 1, high = 0)

UP ↓	RML (msec)	< 10	10 - 20	20 - 40	40 - 80	> 80
	→					
0		0	1	2	3	4
1		1	2	3	4	4

Figure 3. The CAM Priority Assigning

The Earliest Deadline First (EDF) is the queuing discipline of the packet buffer. This method selects the packet that needs to be sent earlier than the others, according to its CAM priority, residual lifetime and user priority. An example of this packet selection method is shown in Figure 4.

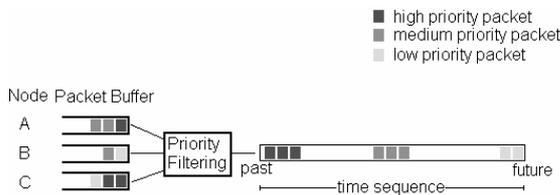


Figure 4. The Priority Scheme

HIPERSIM simulates the priority mechanism of HIPERLAN. Every packet carries a residual lifetime, a user priority and a CAM priority. These attributes get updated when the packet buffer is managed. The HIPERSIM results show the efficiency of the EDF method in contrast to the classic FIFO (ETSI 1998; Jacquet et al. 1996b).

### 3.2. The EY-NPMA Medium Access Protocol

The medium access protocol (MAC) that is used in HIPERLAN is the Elimination Yield Non-pre-emptive Priority Multiple Access (EY-NPMA) protocol. EY-NPMA is based on active signaling and it is suitable for wireless local area networks, like HIPERLAN. The medium access mechanism is based partially on the well known CSMA, since the node eventually “listens” to the

medium to find out if it can transmit or not. However, the main mechanism differs from CSMA, since every node contests for the medium access in an active way by transmitting some special signals (Jacquet et al. 1996a). Below, the operation of the EY-NPMA protocol is presented briefly, according to the “ETSI functional specification EN 300 652 v1.2.1” (ETSI 1998).

A node that wants to transmit initially senses the channel. If it is idle, it enters the “channel free condition” and eventually transmits the packet. If the channel is not idle, then the node waits to synchronize with the others nodes at the end of the current transmission, so it enters the “synchronized channel condition.” The synchronized channel access cycle consists of three phases, which define the access pattern for the competitive nodes (ETSI 1998; Jacquet et al. 1996a; Chevrel et al. 1996; LaMaire et al. 1996).

The first phase is the “Prioritization Phase” where the nodes carrying the highest CAM priority win and make the rest defer. In Figure 5, it is shown how node B “wins” during the Prioritization Phase and makes node A defer. CAM priority of node A packet is 2, while CAM priority of node B packet is 1 (higher priority).

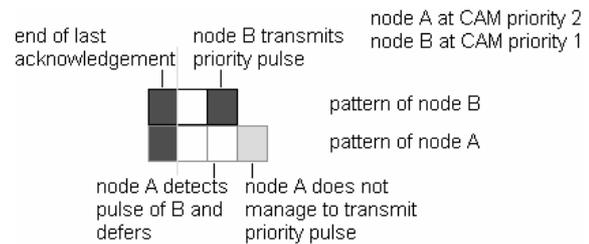


Figure 5. The Prioritization Phase

The nodes that “survive” the Prioritization Phase proceed to the Elimination Phase. During the Elimination Phase, a great percentage of the contending nodes is eliminated, but at least one of them survives. This takes place in a random manner and according to a geometric distribution of probability  $p = 1/2$ . In Figure 6, it is shown how node B “wins” during the Elimination Phase and makes node A defer. Node A transmits an elimination pulse 1 slot long, while node B transmits an elimination pulse 2 slots long.

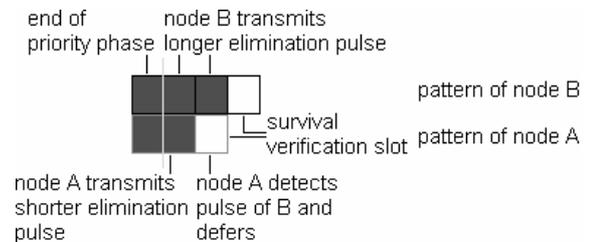


Figure 6. The Elimination Phase

The Yield Phase is the last phase before the transmission of a data packet and it is the last effort to reduce the number of the contending nodes. The nodes “win” randomly and according to a uniform distribution. If more than one node survive this phase, they will start transmitting simultaneously and a collision will take place. In Figure 7, it is shown how node B “wins” during the Yield Phase and makes node A defer. Node A allows 3 idle slots, while node B allows 2 idle slots.

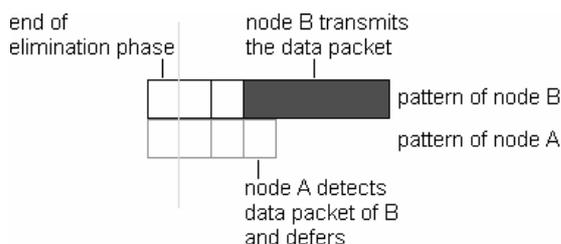


Figure 7. The Yield Phase

In Figure 8, an overview of the EY-NPMA protocol is shown.

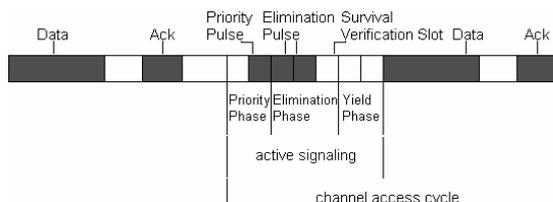


Figure 8. Overview of the EY-NPMA Protocol

## 4. THE HIPERSIM SIMULATION ENVIRONMENT

### 4.1. The HIPERSIM Features

#### 4.1.1. Communication Range Issues

In a WLAN, not all of the nodes are expected to be able to communicate directly with each other. This is due to the fact that the antenna of a node has a limited range, and the obstacles that might intercept the communication. Specifically, the range of HIPERLAN is approximately 50 meters when the nodes are moving slower than 1.4m/s.

A special characteristic of the wireless environment is the fact that when two nodes are not able to communicate directly, they might be able to sense the signal transmitted by each other. More specifically, in a wireless network, like HIPERLAN, two nodes are able to exchange data packets when they are able to detect the transmitted signal and identify the bits sent. This happens when the signal attenuation due to distance and the signal fading are not intense enough to make the communication between the two nodes impossible. In this case, the two nodes are assumed to be in communication range. When the quality of the received signal is not good enough to allow data transmission, but it can still be detected, then the two

nodes are assumed to be in sense range (signal detection range). Obviously, the sense range is greater than the communication range, since the former includes the latter, so it is likely that in a WLAN two nodes can detect each other even when they are not able to communicate directly. A special feature of HIPERSIM is that it distinguishes between these two kinds of ranges, when it simulates the hidden nodes (Wilkinson b).

The hidden nodes are pairs of nodes where one of them cannot receive data from the other, because of the distance or the obstacles between them. They cause overall reduction of the system performance. The range issues mentioned before are related to the “hidden nodes.” We assume that two nodes are hidden when they are out of communication range. Two hidden nodes might be either in sense range or not. The simulation results show that a great reduction of the network performance is caused mainly by the hidden nodes that are out of sense range. That is the reason why it is important to make the distinction between the communication range and the sense range. In Figure 9, we can see three nodes of a HIPERLAN network. Every  $C_i$  area represents the communication range (data reception range) for node  $i$  and every  $S_i$  area represents the signal detection range (sense range) for node  $i$ . As we can see, node B is able to send and receive data from nodes A and C, while nodes A and C are hidden from each other, that is to say they are not in communication range. All three nodes are able to detect the signal of the others. In general, EY-NPMA takes advantage of the fact that two hidden nodes in sense range are able to detect the transmitted signal (FU et al. 1996; Weinmiller et al.; Moh et al. 1998).

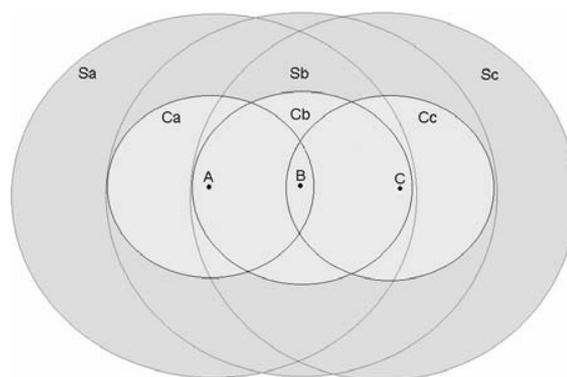


Figure 9. Representation of the Communication Range ( $C_i$ ) and the Sense Range ( $S_i$ )

The simulation engine of HIPERSIM initially defines the pairs of the hidden nodes of the network. Then, it defines the hidden pairs that are out of sense range and those that are in sense range. The object that represents the carrier carries the information about all the transmitting nodes. The nodes that check the carrier can find out which signal they are able to detect, and whether they can identify the

bits sent. If the transmitter is not hidden, then the node can detect the signal and identify the bits sent (communication range). If the transmitter is hidden but in sense range, then the node can just detect the transmitted signal, but it cannot establish a direct communication. Lastly, if the transmitter is hidden and out of sense range, then the node that checks the carrier cannot even detect the signal.

#### 4.1.2. Packet Forwarding

There are two types of nodes in HIPERLAN that are simulated in HIPERSIM: the forwarders and the non-forwarders. The non-forwarders know only their direct neighbors; nodes which are in communication range. On the other hand, the forwarders are aware of the network topology. In case a non-forwarder wants to transmit a packet to a node that is not in communication range (hidden node), it sends it to a forwarder by setting the latter as the intermediate node of transmission. Packet forwarding in HIPERLAN is based on a table-driven routing protocol. This forwarding mechanism increases the system complexity since it requires the continuous watch of the network topology, which changes dynamically. In Figure 10, three nodes (A, B, C) of a HIPERLAN network are represented, where nodes A and B are in communication range, nodes B and C are in communication range, while nodes A and C cannot communicate directly with each other. Nodes A and C constitute a hidden pair, while node B is a forwarder. This means that it can forward packets from node A to B and vice-versa. The packet forwarding mechanism is simulated in HIPERSIM. Initially, the forwarders are defined. Both the forwarders and the non-forwarders are represented by the "Node" class, where there is a property

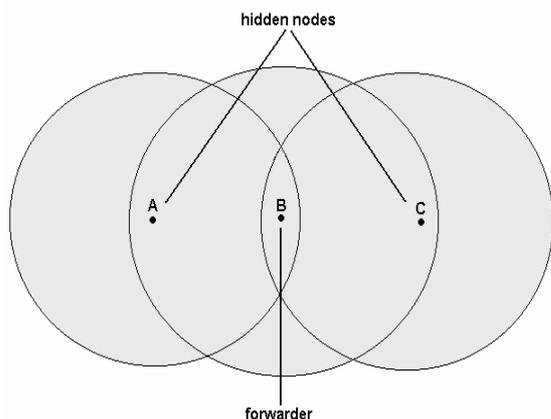


Figure 10. Representation of the Packet Forwarding

that defines whether the specific node is a forwarder or not. So, in HIPERSIM, the forwarders and the non-forwarders have the same structure, except from the fact that a forwarder can communicate directly with any other

#### 4.1.3 Power Saving

The mobile devices that constitute a WLAN have limited energy autonomy. When an antenna transmits or receives a signal, it consumes a great amount of energy. Therefore, a possible solution to the power problem is to turn the antenna off when there is no data transmission or node of the network and it works as an intermediate node that forwards packets between the hidden nodes. All the nodes are aware of the forwarders (ETSI 1998; Weinmiller et al.; Zeng 2000).

reception. In HIPERLAN, some nodes are P\_Savers, while some others are P\_Supporters. P\_Savers are set at status "OFF" for specific time intervals. During these time intervals, they are not able to receive data packets. P\_Supporters have the responsibility to collect data packets that have as destination a P\_Saver which is "OFF." When the P\_Saver returns to normal operation status, the P\_Supporters forward the packets to it. Power saving is optional and it is not fully defined by the HIPERLAN protocol. The HIPERSIM simulates the power saving mechanism and shows that it has satisfactory results. Initially, the P\_Savers and the P\_Supporters are defined. These nodes have the same structure with any other node of the network. The difference is that during the simulation the P\_Savers are turned off from time to time, and the P\_Supporters collect the packets and send them to the P\_Savers when they are back on (ETSI 1998; Zeng 2000)

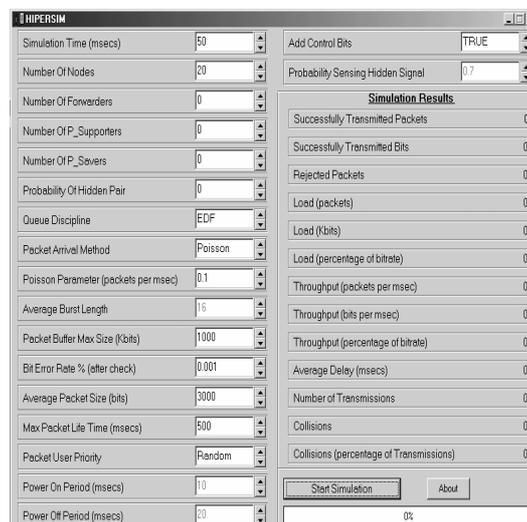


Figure 11. The Graphical User Interface of HIPERSIM

## 4.2. The HIPERSIM Environment

### 4.2.1. Environment Description

Figure 11 shows the environment of HIPERSIM. The user is able to set the simulation parameters, start the simulation, watch the progress bar and finally get the

results that are shown on the right side of the HIPERSIM window. The results of every simulation experiment and the current values of the parameters are automatically saved in a database file by using the ADO mechanism. Every record stores the values of the parameters and the results of the corresponding simulation. HIPERSIM was developed in C++, and it is a W32 application which uses the respective Graphical User Interface (GUI).

#### 4.2.2. Code Structure

The implemented classes in the simulation mechanism are: “Packet”, “Carrier” and “Node.” The class “Packet” represents the common packet in the network. Whatever transmitted in HIPERLAN is a “Packet” object. Among the properties of the “Packet” are size, generation time, sender’s ID, receiver’s ID, lifetime, priority etc. When there is an intermediate node, this might be either a forwarder or a P\_Supporter. The basic methods are the “GetRML” that returns the residual packet lifetime (Residual ML), and the “GetCAM\_Priority,” which returns the CAM priority of the packet. In Figure 12, the structure of the class “Packet” is presented.

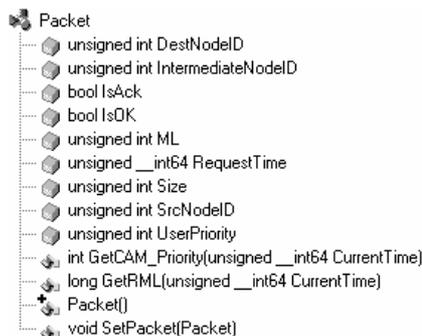


Figure 12. The Class “Packet”

The class “Carrier” represents the communication medium of the network. In general, the class “Carrier,” which produces a single object, “stores” current transmitted packet, status of the medium (idle or data transmitting or in collision et al), nodes transmitting the current moment, and time that the current transmission will be completed. Every node checks the carrier to find out if there is a transmission for it. More than one node might transmit simultaneously during the simulation. These nodes might be in the communication range or the sense range or out of the sense range of the “listener” node. The basic methods are the “PutPacketOnCarrier” which “puts” a packet on the medium, and the “AddNodeInTransmittingInfo,” which adds the node that starts transmitting in the list of the nodes that are currently transmitting. A view of the class “Carrier” is shown in Figure 13.

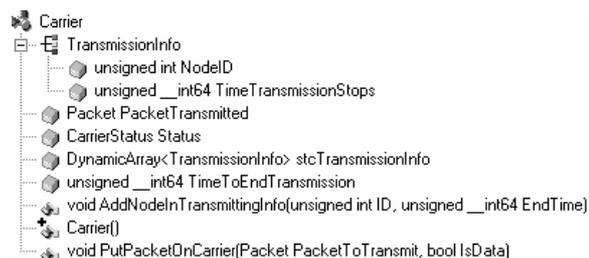


Figure 13. The Class “Carrier”

The class “Node” is definitely the most significant class and it implements most of the operations of the simulator. Every node is represented by an object of the class “Node” and carries a unique ID. There are a large number of properties and methods in the class “Node.” The most important properties are the ID of the node, type of node (Simple or Forwarder or P\_Supporter or P\_Saver), list of nodes that are hidden from it, list of the hidden nodes that it can sense, Packet Buffer, packet to be sent, and a large number of variables that are time indicators and which help to implement the medium access protocol of HIPERLAN. The methods of the class “Node” constitute the “heart” of the simulation mechanism. Below is a list and brief description of each method.

- AddPacketInBuffer: It adds a packet in the buffer.
- ChangeP\_Status: It concerns only the P\_Savers. It checks the conditions and changes the status of the node from On to Off and vice-versa.
- ChannelAccess: This is the basic function that implements the MAC protocol of HIPERLAN, the “EY-NPMA.” When a node has a packet to transmit, it uses this method to gain access to the channel.
- CheckExpiredPackets: It checks the packets in the buffer to discover packets whose lifetimes have expired. After that, it rejects them.
- CheckNode: This is the method that is called to check the node state and decide which actions must be executed by calling other methods.
- GetCurrentBufferSize: It returns the current size of the packet buffer in bits. It is used to find out if there is enough space in the buffer to add a packet.
- InternalPacketArrival: It implements the generation of a new packet inside the node. It decides the size of the packet, destination, lifetime etc. .
- IsNodeHidden: The Boolean returned result shows if the node, which is the argument of the method, is hidden from the node that calls the method.
- IsTransmissionListened: If the node that is calling this method can detect the current transmission, then the method returns the time that this transmission is completed.
- NextArrival\_Burst: It computes the time that the next packet generation takes place, when the selected packet arrival method is the “Bursting.”
- NextArrival\_Poisson: It computes the time that the next packet generation takes place according to the Poisson distribution.

-ReceiveData: This method receives the transmitted data packet which is destined to the calling node. If the calling node is an intermediate (Forwarder or P\_Supporter) for this packet, then the packet is added to the buffer so that it is sent later to its final destination.

-RetransmitData: After the transmission of a data packet and the non-reception of the corresponding acknowledgement, this method is called in order to reschedule the transmission of the packet that was not acknowledged.

-SelectPacketToSend: The packet to be sent is selected from the buffer. The selection is made using the “EDF” or the “FIFO” method, according to the user’s choice.

-SendAck: It implements the acknowledgement sending after the successful reception of a data packet.

-SendData: This method is responsible for the transmission of a data packet. If it is necessary, an intermediate node for this transmission is set.

In Figure 14, the whole structure of the class “Node” with its properties and methods is presented.

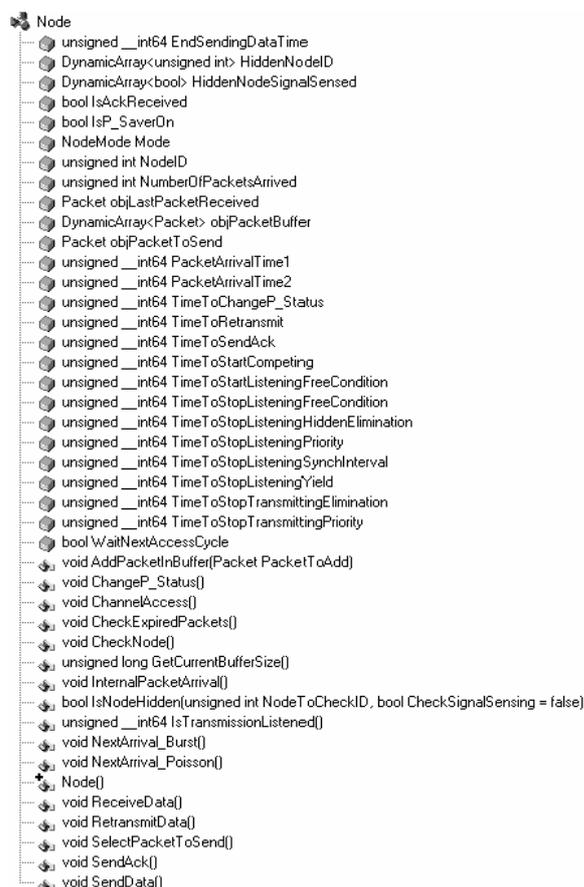


Figure 14. The Class “Node”

### 4.2.3. Operation Sequence

Here we describe briefly the sequence of the actions that take place during the simulation. First of all, we must make it clear that every simulation experiment is

independent from others since the simulation mechanism is initiated every time.

According to the parameters set, packets are generated inside the nodes, and if there is enough space they are stored in the buffer. The destination of every packet is another node of the LAN. The source node tries to gain access to the channel, according to EY-NPMA, in order to send the packet selected from the buffer. In an exhaustive way, the nodes are checked one by one every time unit. The simulation clock step or time unit is equal to the high rate bit-period, which is the time needed to transmit a bit in high transmission rate (23.529 Mbps). The simulation is terminated when the simulation time is over and the results are calculated and stored.

In the beginning of every simulation, all the nodes are initiated as elements of a dynamic list of “Node” objects. The carrier, which is represented by the single object of the class “Carrier” (objCarrier), is also initiated. Next, the “forwarders” and the “P\_Supporters” are selected, provided that the user has made the corresponding choices. After that, the P\_Savers, the pairs of hidden nodes and the hidden nodes that are in the sense range are selected. As it was mentioned earlier, a special feature of HIPERSIM is that it distinguishes hidden nodes from those that are in the sense range and those that are out of sense range. In the next section, the HIPERSIM results show that hidden nodes that are out of sense range affect significantly the HIPERLAN performance. The time the first packet generation takes place is calculated for every node. Afterwards, the application enters the main loop of the simulation, which checks every node at every time moment by using the method “CheckNode.”

The function “CheckNode” decides whether some other methods of the class “Node” will be called. Initially, it checks if there is a packet arrival from another node, so as to call the function “ReceiveData.” Then, it checks if a data packet retransmission is needed, and if it does, it calls the function “RetransmitData.” After that, the function “ChannelAccess” is called. If there is no packet to be sent at the current moment, the “ChannelAccess” function performs no action. Then, the function “CheckNode” checks if an acknowledgement must be sent, so as to call the function “SendAck.” Afterwards, if it is time to generate a new data packet, the function “InternalPacketArrival” is called. Lastly, the function “CheckNode” checks if the node status must change from On to Off and vice-versa by using the function “ChangeP\_Status.” It is obvious that the latter function can be called only when the calling node is a P\_Saver.

When the simulation time is over and the simulation is completed, the results are calculated and presented. A new record is created in the table of the “results.mdb,” and the values of all the parameters and results are recorded there. In order to get these results, some special variables are inserted in different places in the code and their values are updated during the simulation (statistics update). There are also some global, assistant functions, which are called when needed. For example, one of these

functions is the msec conversion to high rate bit-periods (1 msec = 23529 high rate bit-periods). The flowchart of the simulation mechanism is shown in Figure 15.

### 4.3. HIPERSIM Simulation Results

This section analyses the HIPERSIM results of the HIPERLAN under various conditions.

In every simulation experiment, the packet generation inside the nodes is Poissonian. The packet size is variable and exponential. Its default average value is 3000 bits, without counting in the added control bits. The packet lifetime is randomly selected between 10 and 800 msec

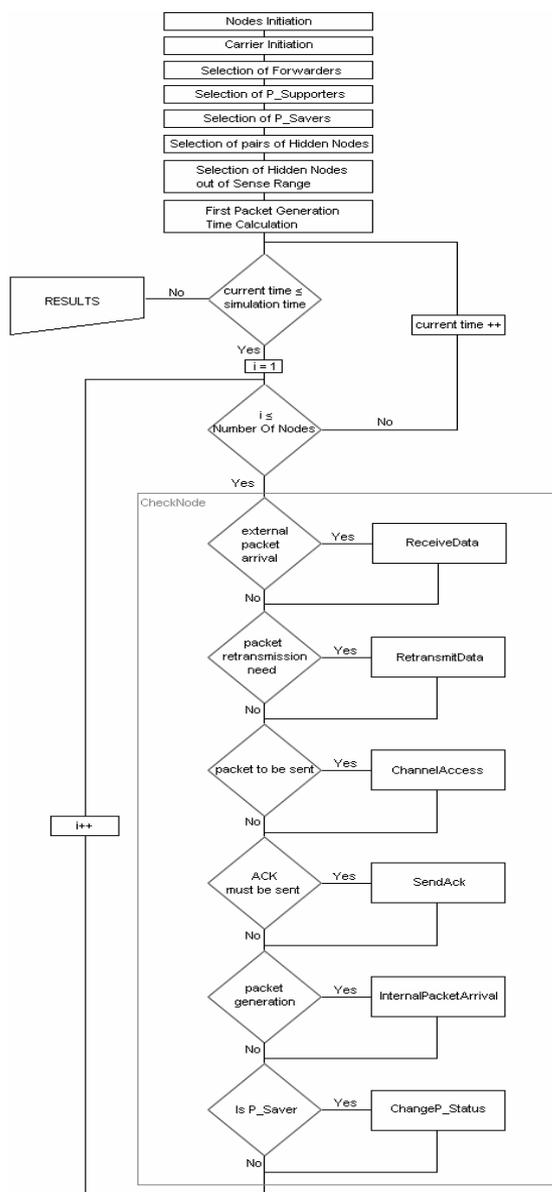


Figure 15. Flowchart of the HIPERSIM Mechanism

Initially, we examine the system throughput depending on the number of nodes of the WLAN. In Figure 16, we notice that the throughput value remains almost stable when the number of nodes increases. This satisfactory performance is due to the fact that EY-NPMA deters the great increase of the collisions even when the number of nodes increases significantly.

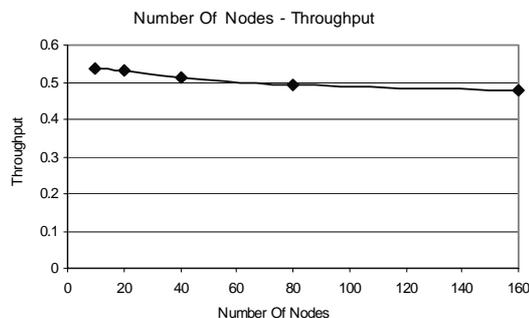


Figure 16. System Throughput versus Number of Nodes

Likewise, the average packet delay, that is the average time the packet remains in the system from its generation time till its successful transmission and reception, is almost stable while the number of nodes increases; see Figure 17.

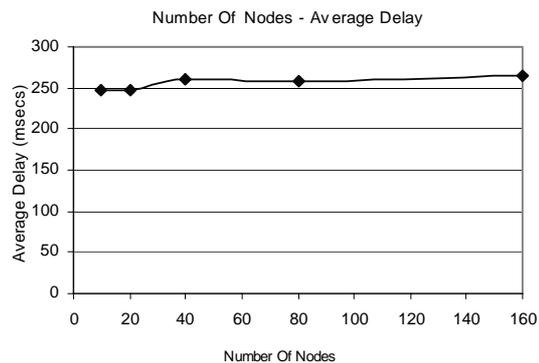


Figure 17. Average Delay versus Number of Nodes

Figure 18 presents the average packet delay versus the Poisson parameter which concerns the packet generation in every node every msec. Actually, the increase of the value of the Poisson parameter causes the increase of the system load, since the packet generation rate increases. In this graph, we examine the performance of the Earliest Deadline First (EDF) buffer discipline in comparison to the traditional First In First Out (FIFO). As it was expected, the average delay is greater when FIFO is used, especially when the load increases. HIPERSIM can simulate both the EDF and the FIFO queue disciplines.

We examine the way that the average packet size affects the throughput, assuming two different values for the Bit Error Rate (after check). The “bit error rate” is the rate a bit error occurs, after the application of all the predefined error detection and error correction checks (that is why

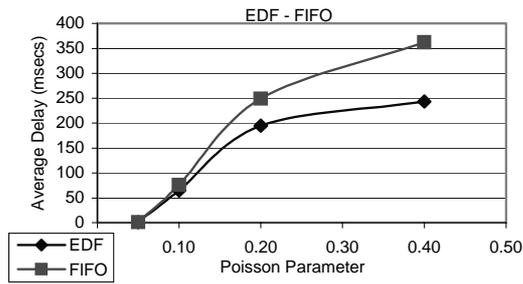


Figure 18. Comparison between the EDF( and the FIFO Queuing Discipline

we assume so low bit error rates and we add the “after check” characterization). According to the value of the bit error rate and current packet length, HIPERSIM decides whether a bit error occurs during a packet transmission. Specifically, this decision is made when a packet is “put” on the carrier using the method “PutPacketOnCarrier” of the object “objCarrier.” In Figure 19, we see that the increase of the average packet size initially causes some increase in the throughput (because of the smaller overhead), which eventually stabilizes. But when we increase the bit error rate, the increase of the average packet size eventually causes a decrement in the throughput.

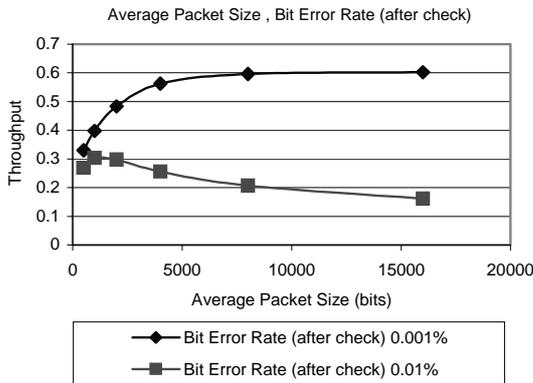


Figure 19. System Throughput as a function of the Average Packet Size and the Bit Error Rate

Another issue is the WLAN performance when there are nodes that operate as P\_Savers. It is found that when the number of the P\_Savers increases, the system throughput decreases. HIPERSIM simulates the “ON” and “OFF” P\_Saver periods. In Figure 20, the simulation results are presented for different numbers of P\_Savers, when the total number of nodes is 30 and the number of P\_Supporters is 2. Every P\_Saver can randomly use any P\_Supporter.

HIPERSIM simulates active signaling, a feature of EY-NPMA, according to which, a node that can just sense the signal of another node (sense range) is able to participate

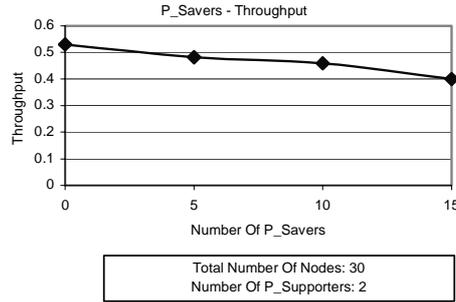


Figure 20. System Throughput versus Number of P\_Savers

successfully with the latter in the synchronized channel access cycle. In case there are two hidden nodes that cannot even detect the signal of one another (they are not in sense range), there is a high possibility that collisions may occur. This would lead to a significant performance degradation. The reason for this performance reduction is the fact that these two nodes would not be able to synchronize directly since each one would have a different view of the channel status. In HIPERSIM, every node has its own view of the channel status, depending on the network topology.

In Figures 21 and 22, we use the term “Probability of Sensing Hidden Signal.” This is the probability that two hidden nodes are in sense range. The “Probability of Hidden Pair” is the probability that two nodes are hidden from each other. As we can see in Figures 21, when the Probability of Hidden Pair increases, the network throughput decreases. If the Probability of Sensing Hidden Signal is low (less than 0.5), the system performance can be characterized as unacceptable. But when the value of the Probability of Sensing Hidden Signal is high and close to one, then the throughput is sufficiently high. The same behavior can be seen in Figures 22, by studying collisions rate. As it was expected, the Percentage of Collisions is high when the Probability of Hidden Pair is high. When the Probability of Sensing Hidden Signal increases, the collisions rate decreases.

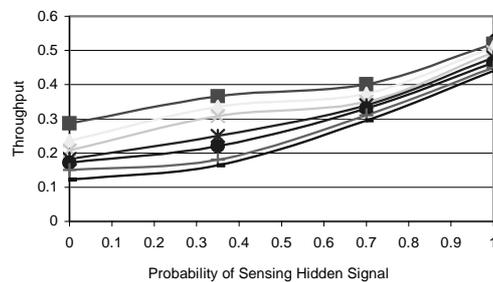
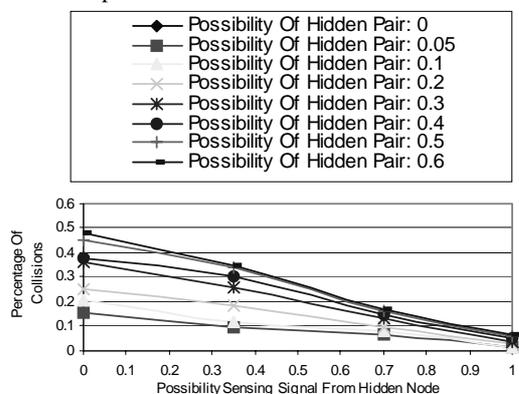


Figure 21. Throughput as a function of the “Probability Of Sensing Hidden Signal” and the “Probability Of Hidden Pair”

## 5. CONCLUSIONS

Simulation of a wireless local area networks (WLANs) has some special features that are different from that for



Fi

Figure 22. Collisions as a function of the "Probability Of Sensing Hidden Signal" and the "Probability Of Hidden Pair"

wired networks. The simulation environments that are used for the traditional wired LANs might be inappropriate for the WLANs. Specifically, a WLAN protocol, like HIPERLAN, has a complicated MAC protocol, which differs significantly from the classical CSMA that is used in most wired LANs. The "hidden nodes" problem is another challenge of the wireless networks which needs extra analysis. HIPERSIM is a simulation environment for the HIPERLAN wireless networks, and it simulates most of the special features of the wireless environment. It is fully parameterized and able to test the behavior of HIPERLAN networks under various conditions and operating environments. The simulation mechanism of HIPERSIM is rather original in the fact that it distinguishes between the communication range and the sense range of a node. Specifically, this work assumes that the communication range is the area where the signal can be detected and the transmitted bits can be identified, while the sense range is the area where the transmitted signal can just be detected. The simulator works in an exhaustive way in order to be accurate. The code structure is object oriented and it is developed on the W32 platform.

The HIPERSIM results have shown that HIPERLAN is an efficient WLAN standard. Probably, some improvement of the protocol is necessary, so that the frequent collisions between nodes that are out of the sense range are avoided. Basically, the "hidden nodes" problem concerns the collisions that take place close to the receiver and not in the sender's region. So it would be efficient if there were a mechanism that informed the neighbors of the receiver about the oncoming transmission. In that case, the receiver's neighbors would not collide, so the overall system performance would improve.

## REFERENCES

- Anastasi, G.; L. Lenzi; and E. Mingozzi, "Stability and Performance Analysis of HIPERLAN," Dept. of Information Engineering, University of Piza, Italy.
- Chevrel, S.; A.H. Aghvami; H.-Y. Lach; and L. Taylor. 1996. "Analysis and Optimization of the HIPERLAN Channel Access Contention Scheme," *Wireless Personal Communications*, Vol. 4, pp. 27-39.
- Coutras, C.; Peng-Jun Wan; O. Frieder. 2000. "Analytical modeling and performance evaluation of the HIPERLAN CAC layer protocol for real-time traffic", *25th Annual IEEE Conference on Local Computer Networks (LCN'00)* November 08 - 10, Tampa, Florida.
- ETSI. 1998. EN 300 652 V1.2.1 (1998-07), ETSI, Broadband Radio Access Network (BRAN); High Performance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification.
- FU, K.; Y.J. GUO; and S.K. Barton. 1996. "Performance of the EY-NPMA Protocol," *Wireless Personal Communications* Vol. 4, pp. 41-50.
- Halls G.A. 1994. "HIPERLAN: the high performance radio local area network standard", *Electronics and Communication Engineering Journal* 6(6) (December 1994) 289-296.
- Jacquet, P.; P. Minet; P. Muhlethaler; and N. Rivierre. 1996. "Priority and Collision Detection with Active Signaling – The Channel Access Mechanism of HIPERLAN," *Wireless Personal Communications*, Vol. 4, pp. 11-26.
- Jacquet, P.; P. Minet; P. Muhlethaler; and N. Rivierre. 1996. "Data Transfer for HIPERLAN," *Wireless Personal Communications* Vol. 4, pp. 65-80.
- LaMaire, R. O.; A. Krishna; P. Bhagwat; and J. Panian. 1996. "Wireless LANs and Mobile Networking: Standards and Future Directions", *IEEE Communication*, 34(8):86-94.
- Moh, W. M.; D. Yao; and K. Makki. 1998. "Wireless LAN: Study of hidden terminal effect and multimedia support," *Proc. Computer Communications and Networks*, pp. 422-431, October 12-15.
- Nicopolitidis, P.; M.S. Obaidat; G.I. Papadimitriou; and A.S. Pomportsis. 2003. "Wireless Networks," Wiley.
- Obaidat, M. S.; and D. Green. 2003. "Simulation of Wireless Networks," in *Applied Systems Simulation: Methodologies and Applications* (M. S. Obaidat and G. I. Papadimitriou, Eds.), Kluwer (in press).
- Papadimitriou, G. I.; and A. S. Pomportsis. 2000. "Learning-Automata-Based TDMA Protocols for Broadcast Communication Systems with Bursty Traffic," *IEEE Communication Letters*, Vol. 4, No.3.
- Sadiku, M.; and M. Ilyas. 1995. "Simulation of Local Area Networks," *CRC Press*.
- Tanenbaum Andrew S. 1996. *Computer Networks*, 3d Edition, Prentice-Hall, Inc.
- Weinmiller, J.; M. Schlager; A. Festag; and A. Wolisz, "Performance Study of Access Control in Wireless LANs – IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN," Technical University Berlin, Germany.
- Wilkinson, T.; T. G. C. Phipps; and S. K. Barton. 1995. "A report on HIPERLAN standardization", *International Journal of Wireless Information Networks*, Vol. 2, No. 2, pp.99-120.
- Wilkinson Tim. "HIPERLAN", HP Labs Europe.
- Zeng J. 2000. "Wireless LAN Standards: HIPERLAN and IEEE802.11," *ECPE 6504 Wireless Networks and Mobile Computing*, Individual Project Report, 04-24-2000.

# From the electronic circuit to the simulation component : an automatic component building process

Vincent Fischer, Laurent Gerbaud

Laboratoire d'Electrotechnique de Grenoble, CNRS UMR 5529 INPG/UJF

ENSIEG BP 46, 38402 Saint Martin d'Hères, France

*vincent.fischer@leg.ensieg.inpg.fr, laurent.gerbaud@leg.ensieg.inpg.fr*

## KEYWORDS

ODE solving, matrix exponentials, constraint optimisation

## ABSTRACT

This paper proposes a process to obtain automatically a simulation component dedicated to optimisation purposes from the schema of an electrical, electronic or power electronics circuit. This component, which computation is based on a specific matrix exponential calculation, has the ability to give the gradients (for SQP optimisation process), and offers short computation time and low memory occupation. After the introduction of the optimisation component problematic, the specific use of the matrix exponential for ODE solving is presented. The computation of the matrix exponential is discussed, then the complete ODE solving and the gradients computation are detailed. After, the component building process is explained. The circuit is analysed in order to obtain its equations as a state system, the calculation code is then generated, and finally the packaging of the component is done. In the last part, some results are analysed in terms of accuracy and computation time.

## 0. INTRODUCTION

This paper deals with the automatic building of the model of an electronic or power electronics circuit, and the integration of this model with its solving algorithm into a Java component designed in an optimisation aim. This component has to provide a computation time as short as possible, with good result accuracy, and the ability to give gradients (for SQP optimisation process). In the paper, only linear state equations are treated, as they are most of the time encountered in the power electronics area. The simulation of hybrid state systems is not taken into account, but will be easily possible in the future, as it is in fact composed of a series of linear state systems, sequenced with continuity conditions on the states.

## 1. THE AIM OF THE PAPER

The constrained sizing of an electrical device can be achieved through an optimisation process, which is based on a sizing model. Such a process is characterised as shown in Fig. 1 :

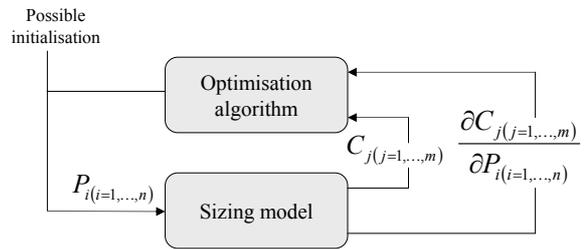


Figure 1: the optimisation process

The outputs  $C_j$  of this model are the values of the sizing criteria. They may concern values of state variables (mainly current in inductors and voltages in capacitors) at a specific date, extrema, r.m.s. or average values, etc... These criteria are calculated from the inputs  $P_i$  of the sizing model, mainly the parameters of the device. These parameters are inductances, capacitances, resistances, etc... The sizing criteria may depend on the state variables  $X$  of the application to size. In the context of the sizing of a power electronic structure, a linearity hypothesis is often possible for the state equation which is defined by (1).

$$\dot{X}(t) = A \cdot X(t) + B \cdot u(t) \quad (1)$$

Here,  $t$  represents the time, and  $u(t)$  is the expression of the state inputs (e.g.: the sources of the electronic circuit).  $A$ ,  $B$ , and  $u(t)$  depend directly on the  $P_i$ , the parameters of the circuit.

Furthermore, some optimisation algorithms are based on gradient methods (e.g. VF13 [7]), which need the partial derivatives of the sizing criteria according to the model inputs, i.e.  $\frac{\partial C_j}{\partial P_i}$ . These derivatives can be expressed with the partial derivatives of the state variables,  $\frac{\partial X(t)}{\partial P_i}$ .

Different approaches exist to obtain the states values from the inputs  $P_i$  [1][2]. They are often

based on numerical simulation using ODE solving algorithms like Trapeze, Runge-Kutta, etc. Obtaining the partial derivatives of the states can be achieved through finite differences. These numerical approaches lead to important computation times, especially if the values of state variables have to be estimated only at a specific date. In the same way, if the sizing model has many outputs depending on several inputs, the computation of the gradient of the state variables according to the model inputs may be very time consuming and numerically sensitive.

Another approach can be considered. As electronic circuit state systems are often linear ones, the ODE solving based on matrix exponentials is a good way to reduce the computation times, as shown below.

The aim of the paper is to propose a tool that generates dedicated calculation components of linear state equations, specifically in the case of electrical circuits. Such a component allows calculating the state variables at a given date, without having to run a complete simulation. It also calculates the partial derivatives of the state variables according to the physical parameters of the state equations (fig. 2), without the need of a second evaluation, as the finite differences method needs.

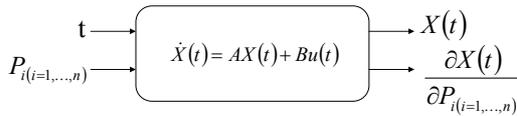


Figure 2: the calculation component

Computation time and memory occupation are as low as possible during the use of this component.

## 2. THE MATRIX EXPONENTIAL FOR ODE SOLVING

As in the paper, the state equation is supposed to be linear, its complete solution can be expressed by the following expression [3] :

$$X(t) = e^{At} \cdot X(0) + \int_0^t e^{A(t-\tau)} \cdot B \cdot u(\tau) \cdot d\tau \quad (2)$$

In order to value such an expression, the exponential of the matrix A has to be defined. Several algorithms can be used to estimate it [4] [5]. Some of these use the eigenvalues of A, but they are limited for our purposes. In this paper, the selected algorithms do not use such methods. They are fast and require few memory space.

Two methods have been selected : the Taylor series development and the Padé approximation. Both may be used in our tool. There are slight differences between the results obtained with each method, but these differences are negligible. The use cases of each method are determined by the state system expression. For very large state systems, the Padé approximation will be privileged, when for smaller systems, the Taylor series will be used.

### 2.1. The Taylor Series Development

The first selected algorithm is based on the Taylor Series Development of the exponential operator :

$$e^A = \sum_{n=0}^{+\infty} \frac{A^n}{n!} \quad (3)$$

Computing this infinite sum is obviously impossible. An upper bound  $N$  must be chosen, depending on the desired calculation accuracy  $\mathcal{E}$ .

To obtain the accuracy  $\mathcal{E}$ , the upper bound  $N$  must satisfy to the following criteria [4]:

$$0 < \left( \frac{\|A\|^{N+1}}{(N+1)!} \right) \left( \frac{1}{1 - \frac{\|A\|}{(N+2)}} \right) \leq \mathcal{E} \quad (4)$$

As long as the Frobenius norm ( $\sqrt{\text{Trace}(A^T \cdot A)}$ ) of the matrix A is smaller than 1, this algorithm gives accurate results, but when the norm is greater than 1, the accuracy is lost.

To compensate for that issue, the scaling and squaring method is used. This method is based on the following expression :

$$e^A = \left( e^{\frac{A}{2^S}} \right)^{2^S} \quad (5)$$

S is chosen so that  $\left\| \frac{A}{2^S} \right\| < 1$ . Then the exponential

of  $\frac{A}{2^S}$  is computed, and finally the result is squared S times. This algorithm gives very accurate results for all kinds of matrixes, as long as they are well conditioned.

The condition number of a matrix measures the sensitivity of the solution of a system of linear equations to errors in the data. It also indicates the numerical disparity between the elements of the matrix. The condition number is calculated as shown in equation (6) :

$$\text{Cond}(A) = \frac{V_{S_{MAX}}}{V_{S_{MIN}}} \quad (6)$$

where  $V_{S_{MAX}}$  is the highest singular value of the matrix, and  $V_{S_{MIN}}$  the smallest one.

When the condition number of a state system matrix is too high, it generally comes from the fact that two different modelling levels have been mixed. For example, mixing the first order modelling with the HF EMC modelling gives components with very disparate numerical values (the first order model will lead to components with big numerical values, as the HF EMC model will lead to very small numerical values). The models of

circuit must be homogeneous on the modelling level. For example, to study the global functioning of a system, the first order model will be used, while the EMC will be studied with a specific HF EMC model.

## 2.2. The Padé Approximation

The  $(p, q)$  Padé approximation is based on the following expressions (7):

$$\begin{aligned} N_{p,q}(A) &= \sum_{i=0}^p \frac{(p+q-i)!p!}{(p+q)!i!(p-i)!} A^i \\ D_{p,q}(A) &= \sum_{i=0}^q \frac{(p+q-i)q!}{(p+q)!i!(q-i)!} (-A)^i \\ R_{p,q}(A) &= [D_{p,q}A]^{-1} [N_{p,q}(A)] \end{aligned} \quad (7)$$

It gives accurate results with reduced computation time [6] as long as the norm of the matrix is small. So, as for the Taylor series development, the scaling and squaring technique has to be used. The results are quite equivalent with the Taylor series methods. There are slight differences in terms of computation time or memory occupation, but it depends on the matrix of which the exponential is computed, and the differences are negligible.

## 2.3. Obtaining the complete solution of the state equation

As the considered systems are power electronics ones, the electrical sources are either constant or sinusoidal.

As it will be detailed in the following part, equation (1), may be derivated according to the input parameters (e.g.: inductance, resistance, etc), giving a new state equation where expression of  $u(t)$  may be constant, sinusoidal, or the product of a polynomial expression with a sinusoidal expression. If  $u(t) = U$  (e.g. a constant source), then :

$$\int_0^t e^{A(t-\tau)} \cdot B \cdot u(\tau) d\tau = [A^{-1} \cdot e^{At} - A^{-1}] B \cdot U \quad (8)$$

This gives:

$$X(t) = e^{At} \cdot X(0) + A^{-1} [e^{At} - I] B \cdot U \quad (9)$$

For a sinusoidal source (e.g.  $u(t) = \sin(\omega t)$ ), the integral term of the solution becomes :

$$[A^2 - \omega^2 I]^{-1} \left[ e^{At} [\omega \cos(\varphi) + \sin(\varphi)] - [\omega \cos(\omega t + \varphi) + A \sin(\omega t + \varphi)] \right] B \quad (10)$$

For a polynomial source, (e.g.  $u(t) = \sum_{k=0}^{d_u} c_k t^k$

where  $d_u$  is the degree), the integral term is:

$$\sum_{k=0}^{d_u} \left[ -c_k A^{-(k-1)} \sum_{n=0}^k \left[ k! e^{At} + \frac{(At)^n k!}{n!} \right] \right] B \quad (11)$$

Finally, all the different solutions for every single source have to be added to obtain the complete solution of the state system:

$$X(t) = e^{At} \cdot X(0) + \sum_{i=1}^{N_s} B_i \int_0^t e^{A(t-\tau)} u_i(\tau) d\tau \quad (12)$$

where  $N_s$  is the number of sources.

## 2.4. Obtaining the states partial derivatives

Obtaining the states partial derivatives can be achieved by numerical methods such as finite differences, which shall be avoided as these methods use more computation time and memory. Moreover, these methods are numerically very sensitive to their adjustment parameters.

Another method giving the states partial derivatives is based on a system symbolic recombination. In this way, the system equation is:

$$\dot{X}(t) = A \cdot X(t) + B \cdot u(t) \quad (13)$$

The derivative of this equation according to any input  $P_i$  of the sizing model gives:

$$\frac{\partial \dot{X}(t)}{\partial P_i} = \frac{\partial A}{\partial P_i} \cdot X(t) + A \cdot \frac{\partial X(t)}{\partial P_i} + \frac{\partial B}{\partial P_i} \cdot u(t) + B \cdot \frac{\partial u(t)}{\partial P_i} \quad (14)$$

A new system can be created by these two equations:

$$\dot{\tilde{X}} = \tilde{A} \cdot \tilde{X} + \tilde{B} \cdot \tilde{u}(t) \quad (15)$$

where :

$$\tilde{X}(t) = \begin{bmatrix} X(t) \\ \frac{\partial X(t)}{\partial P_i} \end{bmatrix}, \tilde{A} = \begin{bmatrix} A & 0 \\ \frac{\partial A}{\partial P_i} & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} B & 0 \\ \frac{\partial B}{\partial P_i} & B \end{bmatrix}$$

$$\text{and } \tilde{u}(t) = \begin{bmatrix} u(t) \\ \frac{\partial u(t)}{\partial P_i} \end{bmatrix}.$$

This new equation system is linear, and can be solved with the methods presented in this paper. In order to obtain the gradients of the states, the state equation has to be derivated according to every input of the sizing model, e.g. the  $P_i$  where  $i = \{1, \dots, N_D\}$ . To obtain the lowest computing time, a new system will be created and solved for each derivate calculated. A system containing all the derivatives could be created with the following matrices:

$$\tilde{X}(t) = \begin{bmatrix} X(t) \\ \frac{\partial X(t)}{\partial P_1} \\ \vdots \\ \frac{\partial X(t)}{\partial P_{N_D}} \end{bmatrix}, \tilde{A} = \begin{bmatrix} A & & & \\ \frac{\partial A}{\partial P_1} & A & & (0) \\ \vdots & & \ddots & \\ \frac{\partial A}{\partial P_{N_D}} & & & (0) & A \end{bmatrix},$$

$$\tilde{B} = \begin{bmatrix} B & & & \\ \frac{\partial B}{\partial P_1} & B & & (0) \\ \vdots & & \ddots & \\ \frac{\partial B}{\partial P_{N_D}} & & & (0) \quad B \end{bmatrix} \text{ and } \tilde{u}(t) = \begin{bmatrix} u(t) \\ \frac{\partial u(t)}{\partial P_1} \\ \vdots \\ \frac{\partial u(t)}{\partial P_{N_D}} \end{bmatrix}.$$

With this system, only one solving is needed to obtain all the derivatives of the states. But the solving of this system needs the exponential of the matrix  $\tilde{A}$ , of which the computation time proportional to  $((N_D + 1) * n)^3$  where  $n$  is the size of  $A$  and  $N_D$  the number of derivatives computed. When the derivatives are calculated separately from each other, the total computation time is proportional to  $(n * 2)^3 * N_D$ . Solving several small systems requires less computation time than solving one large system.

### 3. THE COMPONENT BUILDING

The process leading from the circuit schema to the simulation component is divided in three distinct parts, as shown in Fig. 3. First of all, the current equations are obtained from the circuit scheme, and the state system is extracted from the circuit equations. The computation code is then generated. Finally, the component is packaged.

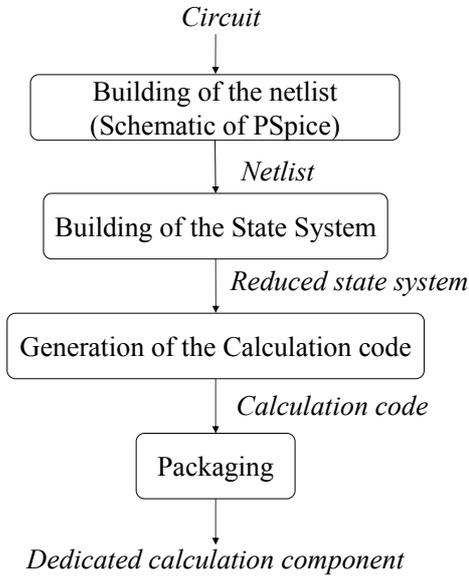


Figure 3: the component building process

#### 3.1. Expressing the state system

The circuit is described using Pspice Schematic. From this tool, a netlist describing the circuit is obtained and can be used by a builder which gives the reduced state equation of the circuit. This builder parses the PSpice netlist to extract the node equations of the circuit. From that, an

independent set of mesh equations is built and the equations of each components are given [8]. The modelling level used in a sizing process allows having simple linear models for the components (resistors, inductors, capacitors).

Having all the equations of the circuit, the reduced state equations are obtained using symbolic treatment implemented in Macsyma [9].

The time differentiated variables of a circuit are the currents in its inductors and the voltages on its capacitors. Thanks to the algebraic relations between the voltages of the circuit (from the mesh equations) and the currents in the circuit (from the node equations), the state variable vector is only made of a part of these variables.

In this way, the state equation is defined by equation (1). The state variable vector  $X$  is made with currents in inductors and the voltages on capacitors. The input vector  $u(t)$  is made with the circuit sources. Finally, the coefficients of the state variables and the sources in each equation are symbolically computed, defining the matrixes  $A$  and  $B$ . So, the state system is symbolically expressed.

#### 3.2. Automatic building of the calculation code

With the expression of the state system, the model calculation code can be generated automatically. In this process, such a component calculates the state variables at a given date, without having to run a complete simulation. It can also calculate the partial derivatives of the state variables according to the physical parameters of the circuit (e.g. resistances, inductances, capacitances, etc.) (see fig. 4).

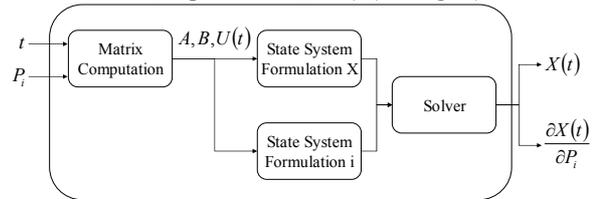


Figure 4: Structure of a generated calculation component

The created calculation component is Java based, but it uses some algorithms written in C (the matrix exponential algorithm for example). The generated Java code calls the C code during the calculation of the model. First of all, the calculation code of the state system matrixes ( $A$ ,  $B$ , and  $u(t)$ ) can be generated without any particular difficulty, as their expressions are ANSI-C compliant. The generation of the calculation code of the partial derivatives of the state system matrixes requires a symbolic derivation treatment. In the paper, a Java based lightweight derivation tool developed in our laboratory (RAMA [11]), has been used.

Then the code using the calculation of the matrix exponential and the calculation of the state system

matrixes is generated. This code can calculate the value of the state vector  $X$  at any date, without simulating any transient state. Finally, the specific code of the calculation component is generated. This last code includes the definition of the inputs and outputs of the component. It also includes the default implementation of the simulation component interface. This interface contains the methods necessary to access to its inputs and outputs (reading and writing their values, and connecting an input to an output of another component). It also contains two computation methods. The first one launches the computation of the outputs with the actual values of the inputs. The second one launches the computation of the partial derivatives of the outputs according to the specified inputs. All of these methods are implemented automatically. The code is then compiled.

### 3.3. Packaging the component

All the generated Java classes and the dynamic library containing the C functions (matrix exponential computation, matrix inversion, matrix determinant, ...) used by the Java code are included in a Jar (Java ARchive) file containing a manifest pointing to the main class of the component, which is the class implementing the component interface. During the loading of this class, all the other classes (which are used by the component main class) are loaded, as well as the dynamic libraries, which contain the C code. The component is ready to compute the model, or to be connected with others components...

## 4. RESULTS

The results obtained with a static converter plugged on an electronic circuit representing the RSIL filter of a power electronics converter (fig. 5) is presented to illustrate the approach. The results given by the generated calculation component are compared with numerical simulations, in terms of accuracy, computation time and memory occupation. Simulations have been made with Simplorer [10].

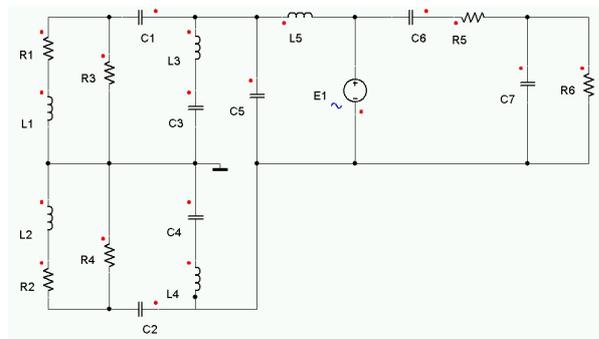


Figure 5: the electronic circuit

The generation of the component is done in less than one minute, including the analysis of the

circuit, the generation of the state equation, the generation of the computation code and the packaging into a component.

The comparison is done for several frequencies of the voltage source, to get a good representation of the harmonic spectrum of the operating mode.

Although the whole simulation takes a little more computation time with the matrix exponential approach than with a numerical approach, the time gain lies in the fact that each point of the simulation can be obtained without the knowledge of the others, whereas the numerical simulation needs the knowledge of the past to obtain the next point. It means that obtaining the states values at  $t = 1s$  will require at least thousands of calculated points with numerical integration methods, whereas only one resolution is needed with the matrix exponential approach. As a consequence, the computation is far faster and the memory occupation far lower with the matrix exponential method than with the numerical simulation. The accuracy is equivalent with both methods, as shown in Fig. 6.

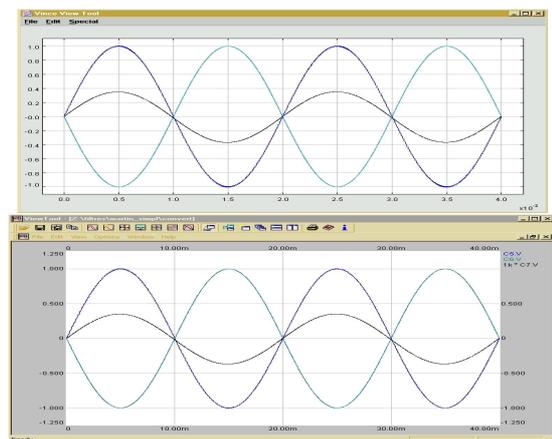


Figure 6: the results of the matrix exponential approach (up), and the numerical simulation (down)

Moreover, the calculation component gives also the gradients for optimisation processes.

## 5. CONCLUSION

In this paper we have presented a process which builds automatically a Java based component containing a model of an electronic circuit, and the algorithms required for the computation of this model. The component is automatically generated from the schema of the circuit.

The computation of the model is based on a methodology using symbolic treatments of the models and matrix exponential to solve the ordinary differential equations involved in the models of electronic circuits. This method gives directly the values of the states at a specific date and their partial derivatives according to the circuit parameters, for optimisation purposes. To the contrary, numerical methods must start from the

origin and integrate the ODE step by step until they reach the desired date. To obtain the partial derivatives of the states with the finite differences, another simulation must be computed for each input of the model.

The presented methodology requires few computing time and few memory occupation. The states are obtained with a very good accuracy, as well as the partial derivatives, to the contrary of the derivatives obtained with finite differences, which are numerically sensitive to the differentiation step, and which may be unstable.

The purpose of this methodology based on matrix exponentials is not the simulation of electrical devices, as other tools can do it very well, but to reduce the computing time while obtaining the values of the states at a certain date and their partial derivatives. This methodology shall reduce the overall computing time of the optimisation of an electrical device.

## 6. REFERENCES

- [1] Kragh H., Blaabjerg F., Pedersen J.K., "An advanced tool for optimised design of power electronic circuits", proceeding of IEEE-IAS'98, Saint Louis, Missouri, USA, October 12-15, 1998, pp 991 – 998
- [2] Viarouge P., Tourkhani F., Kamwa I., Le-Huy H., "Nonlinear optimization techniques for the design of static converters", proceedings of the IMACS-TC1'93, 4<sup>th</sup> international conference, Association for Mathematics and Computers in Simulation, Montreal, Canada, July 7-9, 1993, pp 543 – 547
- [3] J. D'Azzo, C. Houpis, "Linear Control System Analysis and Design", 4th Ed., McGraw Hill Book Co., 1995
- [4] C. Moler, C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix", SIAM Review 1978, Vol. 20 No.4, pp 801 – 836
- [5] W. Harris Jr, J. Fillmore, D. Smith, "Matrix Exponentials – Another Approach", SIAM Review 2001, Vol. 43 No 4, pp 694 – 706
- [6] V. Fischer, L. Gerbaud, J. Bigeon, "Solving ODE for optimisation: Specific use of the matrix exponential approach", OIPE 2002, Lodtz, Poland
- [7] VF13, <http://www.cse.clrc.ac.uk/nag/hsl/>
- [8] C. Lechevalier, L. Gerbaud, J. Bigeon, "Automatic design of discrete time-models of static converter", Simulation in Industry, 8th SCS-ESS'96 (European Simulation Symposium), Genoa, Italy, October 24-26, 1996, pp 475-479.
- [9] L. Gerbaud, J. Bigeon, G. Champenois, "Modular approach to describe electromechanical systems. Using Macsyma to generate global approach simulation software", Conference record of the IEEE PESC'92 (Power Electronics Society Conference), June 29 - July 3, 1992, Toledo, Spain, pp 1189-1196
- [10] Simplorer, <http://www.ansoft.com/products/em/simplorer/>
- [11] V. Fischer, L. Allain, "RAMA : a lightweight rule-based tool for expressions analysis and code generation", ESS 2003, 26 – 29 October 2003, Delft, The Netherlands

# SIMULATION IN CRYPTOGRAPHIC PROTOCOL DESIGN AND ANALYSIS

Ning Su  
Richard N. Zobel  
Frantz O. Iwu  
Department of Computer Science  
University of Manchester  
Oxford Road, Manchester, M13 9PL  
United Kingdom

[ningsu@lachine.co.uk](mailto:ningsu@lachine.co.uk), [rzobel@cs.man.ac.uk](mailto:rzobel@cs.man.ac.uk), [iwuo@cs.man.ac.uk](mailto:iwuo@cs.man.ac.uk)

## KEYWORDS

Agent-based simulation, cryptographic protocols, encryption, decryption, Java security features

## ABSTRACT

Security and safety were and still are a major concern for distributed computing systems and similar networks of computer environments. A number of cryptographic protocols have been proposed to achieve security and safety of network communication. There is growing interest in using computer simulation to help with understanding, analysing and designing of dynamic complex real systems. This paper studies the Agent-based simulation system on modelling the cryptographic protocols. A general formula of Agent for the simulation of cryptographic protocols has been proposed, and the dynamical environment of the simulation, which includes encryption, decryption of messages, and communication between the Agents, has been achieved by using the Java technology. The simulation system provides an approach for the designer to analyse and verify the cryptographic protocol during the design process.

## INTRODUCTION

There are two important aspects concerned in the security and safety of the distributed computing systems and similar networks of computers environment:

- (a) Authentication of different computers and users in the network.
- (b) Protection of messages passing among them and preventing illegal access of resources.

Data passing over networks is particularly vulnerable to attack. Encryption is necessary when transferring confidential information, such as bank details, ID and personal data, etc. In general, there are two encryption schemes: symmetric (secret-key) encryption and asymmetric (also called public-key) encryption [1]. In the symmetric encryption, the same key is used for

encryption and for decryption, and therefore it must be distributed through a secure channel in the first place. In public encryption, there is a key pair consisting of a public key and a matching private key, in which one key is used for encryption and another for decryption. Asymmetric encryption (such as RSA and ECC) has advantages over symmetric encryption in the aspects of security and key management, but usually is much slower, requiring much more computation.

Simulation techniques have proven to be useful for designing real world systems. Modelling and simulation can provide a way of analysing, understanding and optimising the dynamic complexity of real systems. Simulation can also be used to verify reliability and correctness of system designs. The cryptographic protocol can be simulated to help for design, analysis and test before the real system is constructed and deployed.

Java is an object-oriented programming language with a comprehensive set of security and safety measures built in. The multithreading is also supported directly in Java. The simulation environment of cryptographic protocols presented in this paper has been achieved by using the Java technology.

## CRPTOGRAPHIC PROTOCOLS

A cryptography system is built on many levels. Building upon the encryption algorithms are protocols. A number of cryptographic protocols have been proposed to achieve security and safety for network communications [2]. It is not sufficient to study the security of the underlying algorithms alone, as a weakness on a higher-level protocol can render the application insecure regardless of how good the underlying cryptographic algorithms are. Protocols involving shared-key cryptography use an authentication server which shares a key with each entity and typically generates new session keys for communication between the entities. Public-key protocols use a certification authority which is trusted to pass on the public keys of the entities. Entities in

cryptographic protocols are referred as principals and assumed to have unique identities.

### The Otway-Rees Protocol

A share-key authentication protocol was proposed by Otway and Rees in 1987 [3]. The details of the Otway-Rees Protocol are as below:

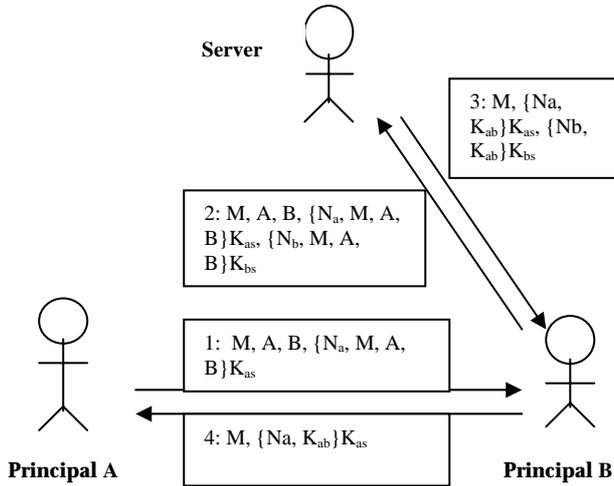


Figure 1 The Otway-Rees protocol

- State 1: Principal A sends “ $M, A, B, \{N_a, M, A, B\}K_{as}$ ” to principal B.
- State 2: Principal B sends “ $M, A, B, \{N_a, M, A, B\}K_{as}, \{N_b, M, A, B\}K_{bs}$ ” to Server.
- State 3: Server sends “ $M, \{N_a, K_{ab}\}K_{as}, \{N_b, K_{ab}\}K_{bs}$ ” to principal B.
- State 4: Principal B forwards “ $M, \{N_a, K_{ab}\}K_{as}$ ” to principal A.

A and B are unique identities of principals A and B.  $M, N_a$  and  $N_b$  are specific statements generated by principals A and B.  $K_{as}$  and  $K_{bs}$  are shared keys between server and principals A and B.  $K_{ab}$  is a session key generated by server for communication between principals A and B.

In the protocol, principal A sends principal B some encrypted information encrypted with  $K_{as}$ , together with enough information for principal B to generate a similar encrypted message with  $K_{bs}$ . Principal B forwards both information to the server. The server decrypts the encrypted information and checks whether they match  $M, A$  and  $B$ . If they are matched, then the server generates session key  $K_{ab}$  and embeds it in two encrypted messages using shared key  $K_{as}$  and  $K_{bs}$  with appropriate statements. The two encrypted messages are sent back to principal B and then principal B forwards the appropriate part to principal A. Principals A and B decrypt the messages and verify the contents.

If both satisfied, then principals A and B start to use session key  $K_{ab}$  to communicate between them.

### A GENERAL FORMULA OF AGENT FOR SIMULATION OF CRYPTOGRAPHIC PROTOCOL

Intelligent agents are one of the most important developments in computer science to have emerged in the past decade [4]. Agent-based simulation can be used to model cryptographic protocols [5] [6]. A conceptual model of the simulation system needs to be introduced to describe how agents may communicate within a simulation environment and formalised using the Discrete Event Simulation (DEVS). DEVS can be used to describe the autonomous and dynamic behaviour of agents and their reaction for events [7]. A cryptographic protocol can be considered to include a set of agents and communication channels. Each agent is an autonomous and reactionary entity (principal) with the capability of performing a sequence of operations (events) on information. A channel is an abstraction of the communication facility. The agents interact each other according to some predefined rules to send, receive and process information via the communication channels. A general formula of agent characterised by a tuple is introduced as below:

$$\Sigma_{Agent} = ( X, S, Y, \delta_{int}, \delta_{ext}, \lambda, \gamma, M, ta )$$

in which,

- $X = \{x_1, x_2, \dots, x_n\}$  is a non empty set of input events.
- $S = \{s_1, s_2, \dots, s_n\}$  is a non empty set of admissible sequence of states.
- $Y = \{y_1, y_2, \dots, y_n\}$  is a non empty set of output events.
- $\delta_{int}$  : An internal state transition function describing the behaviour of a finite state automaton.
- $\delta_{ext}$  : An external state transition function describing reaction of the agent to external events.
- $\lambda$  : An output function, which maps the internal agent state to the output set.
- $\gamma$  : an input/output coupling relationship
- $M$  :  $\{m_1, m_2, \dots, m_n\}$  is a non empty set of unique component references.
- $ta$  : Represents the time the agent stays in a particular state before transiting to the next sequential state.

### AGENT SIMULATION MODEL OF OTWAY-REES PROTOCOL

Using the agent definition, agent simulation model for the Otway-Rees protocols is presented. In the model, principal A, principal B and the Server are referred as Agent A, Agent B and Agent S. Figure 2 shows the input and output ports of Agent A, Agent B and Agent S in Otway-Rees Protocol respectively:

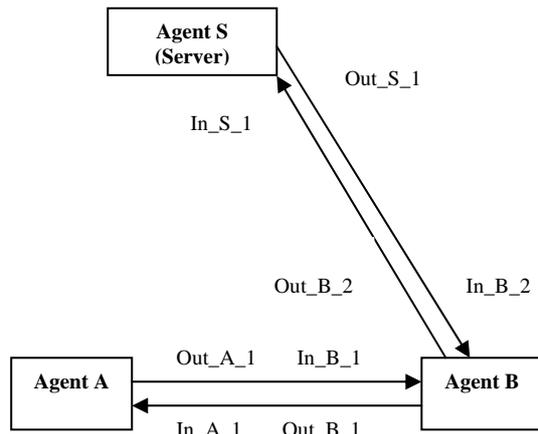


Figure 2: Simulation model of Otway-Rees Protocol

### Agent A:

In this model, Agent A has an output port Out\_A\_1 and an input port In\_A\_1 for communication with Agent B. A formal description of agent A is shown as below:

$$\Sigma_{AgentA} = (X, S, Y, \delta_{int}, \delta_{ext}, \lambda, \gamma, M, ta)$$

$$X = \{In\_A\_1\}$$

$$S = \{idle, send, receive, save, retrieve, process\_message - verify, generate, encrypt/decrypt\}$$

$$Y = \{Out\_A\_1\}$$

$$\delta_{int}(idle) = (make\_INI\_REQUEST).$$

$$\delta_{int}(cond = true) = (process\_message).$$

$$\delta_{int}(cond = true) = (send\_message, send).$$

$$\delta_{ext} = In\_A\_1 = (receive\_message, receive).$$

$$\lambda(cond = true) = Out\_A\_1 = message.$$

$$\gamma = \{(Out\_A\_1, In\_B\_1), (In\_A\_1, Out\_B\_1)\}$$

$$M = \{Agent\ B\}$$

$$ta = Time\ value.$$

The *cond* is a conditional variable, which indicates satisfaction of processing messages. The agent first makes an internal transition from *idle* to *process\_message* state. When this occurs, a message is generated and then the state is transmitted to *send* state. The agent remains in this autonomous state until an external event occurs. Similarly, Agent B and Agent S can be developed trivially [8].

## IMPLEMENTATION

The Java technology has been used to achieve the dynamical environment of the simulation. The virtual networking, which uses special address of **localhost**, has been used to establish the simulation environment [9]. In the DEVS models, each agent comprises various parts as shown in figure 3.

Agent name
Declaration of variables
Communication ports
Dynamic Events: Send and receive information; Save and retrieve information; Processing information – verificate, generate, encrypt, decrypt information, etc.

Figure 3 Basic structure of an agent composition

In the model, some agent needs to interact with more than one agent and others just communicate with one agent. The class **socket** is used to establish a communication channel for the agent without multi-communication capability:

```
Socket s = new Socket ("localhost", port number);
```

The multi-communication capability of agent is achieved by combining classes **Serversocket** and **Thread**:

```
ServerSocket ser = new ServerSocket (port number);
```

And a loop for the thread class:

```
While(!false)
{
Socket s = ser.accept();
Thread t = new ThreadedHandler (s);
t.start();
}
```

The class of Threadedhandler drives from class Thread and contains the communication loop with the other agent in its run method. Thus, for each new connection with a new agent (i.e. each new socket connection), a new thread will be launched to take care the communication and therefore the agent can communicate with more than one agent at the same time. Different types of information (text, binary data and serialized object) can be transferred by using different classes. In the simulation model, all information (plaintext or ciphertext) are converted to binary data and then transmitted [8].

In order to using the Java security features, packages **java.security** and **javax.crypto** should be imported into the Agent (Java program). If using RSA algorithm, a new provider ("Legion of Bouncy Castle") needs to be added in the security description file [10] (*jre/lib/security/java.security*):

```
Security.provider.6=org.bouncycastle.jce.provider.BouncyCastleProvider
```

The following classes have been used for generating secret key:

```
KeyGenerator keygen = KeyGenerator.getInstance
    ("algorithmName");
SecureRandom random = new SecureRandom();
Keygen.init(random);
SecretKey key = keygen.generateKey();
```

In the simulation models, DES cipher algorithm is used, so

```
algorithmName = DES;
```

For generating asymmetric keys, following classes have been used:

```
KeyPairGenerator keysgen =
    KeyPairGenerator.getInstance
    ("algorithmName", "providerName");
SecureRandom random = new SecureRandom();
Keysgen.initialize(keysize, random);
KeyPair keys = keysgen.generateKeyPair();
Key publickey = keys.getPublic();
Key privatekey = keys.getPrivate();
```

In the simulation models, RSA cipher algorithm is used and the provider is BC (Legion of Bouncy Castle), so

```
algorithmName = RSA;
providerName= BC;
```

The class *Cipher* is used for all encryption algorithms:

```
Cipher cipher =
    Cipher.getInstance("algorithmName",
        "providerName");
```

Details of the implementation of cryptosystem are presented in [8].

## SIMULATION RESULTS

The computer used for the simulation is a PC (Pentium 2.0GHz, 256MB memory, Windows XP). Java™ 2 SDK, Standard Edition Version 1.4.0 has been installed in the PC.

As one window can only run one Agent, multiple windows should be opened simultaneously for the simulation of the protocols.

Symmetric key DES and asymmetric key RSA have been used throughout the simulation.

First, a simple simulation model of Otway-Rees Protocol was implemented to test the cryptosystem and communication between Agents. In the test, the plaintext of Agent A was encrypted using session key

*Key<sub>AB</sub>* that was generated by Agent S (Server) and then sent to Agent B. Agent B received the encrypted message and decrypted it using the same key. The screen output of the test is shown in Figure 4 and 5. The results show that the encryption, decryption and transfer of the message have been successfully carries out.

```
Plaintext sent to Agent_B:
    Hello, message from Agent_A

Encrypted ciphertext in text form:
    ?x?? c?t?}},??<roJ}?b4<??
```

```
Encrypted ciphertext sent in binary form:
-121 10 127 58 -121 24 -108 116 -9 -75 93 125 44 -115 13 120
-64 4 99 10 -95 60 114 111 74 125 -36 98 52 60 -15 6
```

Figure 4 Agent A's message (plaintext and ciphertext) sent to Agent B

```
Encrypted ciphertext received in binary form:
-121 10 127 58 -121 24 -108 116 -9 -75 93 125 44 -115 13 120
-64 4 99 10 -95 60 114 111 74 125 -36 98 52 60 -15 6

Encrypted ciphertext received in text form:
    ?x?? c?t?}},??<roJ}?b4<??
```

```
Decrypted plaintext received from Agent_A:
    Hello, message from Agent_A
```

Figure 5 Agent A's message (ciphertext and plaintext) received by Agent B

The simulation models of four Cryptographic protocols (Otway-Rees, Needham-Schroeder, Kerberos and Digital Envelope protocols) have been developed and details of simulation results are presented in [8].

In the simulation of Otway-Rees Protocol, three windows have been opened simultaneously for Agent A, Agent B and Agent S respectively. The interaction between the Agents in the protocol is shown in Figures 6-8. The execution sequences are as following:

- (1) Start Agent B to wait for communication.
- (2) Start Agent A to make initial contact with Agent B. Agent A is then awaiting an input from keyboard after sending the message to Agent B. If **CON** is inputted, the state will be transmitted to the next state.
- (3) Start Agent S to wait for communication with Agent B.
- (4) Agent B contacts with Agent S.
- (5) Agent S replies to Agent B's request.
- (6) Agent B receives Agent S's message.
- (7) Agent B verifies the message.
- (8) Agent B passes the Agent S's message to Agent A.
- (9) Agent A verifies the message.
- (10) Secure communication is then established between Agent A and Agent B if satisfied.

Communication with Agent\_B:

State 1: retrieve secret key key\_AS for communication between Agent\_A and Agent\_S  
State 2: Generate message to Agent\_B: M, A, B, {Na, M, A, B}k\_AS  
State 3: Encrypt {Na, M, A, B} using Key\_AS  
State 4: Send the message to Agent\_B

Enter END to exit, Enter CON to continue

CON

State 5: Received encrypted Agent\_S's message via Agent\_B  
State 6: Decrypt the message using K\_AS  
State 7: Verify the received message

Verification satisfied

State 8: Unwrap the wrapped key\_AB if satisfied  
State 9: Decrypt the message from Agent\_B using key\_AB

Message received from Agent\_B: Hello, message from Agent\_B

State 10: Send an encrypted message to Agent\_B using key\_AB

Message sent to Agent\_B: Hello, message from Agent\_A

Figure 6 Agent A's execution in Otway-Rees Protocol

Communication with Agent\_A:

State 1: Received Agent\_A's message

Communication with Agent\_S:

State 2: Retrieve secret key key\_BS for communication between Agent\_B and Agent\_S  
State 3: Generate Agent\_B's message  
State 4: Send the message (including Agent\_A's message) to Agent\_S  
State 5: Received Agent\_S's message  
State 6: Verify the received message from Agent\_S

Verification satisfied

State 7: Unwrapped session key: key\_AB

Communication with Agent\_A:

State 8: Generate Agent\_B's message to Agent\_A

Message sent to Agent\_A: Hello, message from Agent\_B

State 9: Encrypted Agent\_B's message using key\_AB

State 10: Send encrypted Agent\_B's message and pass Agent\_S's message to Agent\_A

State 11: Received encrypted Agent\_A's message

State 12: Decrypt Agent\_A's message using key\_AB

Message received from Agent\_A: Hello, message from Agent\_A

Figure 7 Agent B's execution in Otway-Rees Protocol

Communication with Agent\_B:

State 1: Retrieve secret key Key\_AS and Key\_BS  
State 2: Received Agent\_B's message (including Agent\_A's message)  
State 3: Verify the received message

Verification satisfied

State 4: Generate message including session key key\_AB for communication between Agent\_A and Agent\_B

State 5: Encrypt the message with key\_AS and key\_BS respectively

State 6: Send the message to Agent\_B

Figure 8 Agent S's execution in Otway-Rees Protocol

## APPLICATION OF THE SIMULATION

The simulation of cryptographic protocols could have various applications. For example, the efficiency of the protocols including encryption/decryption versus message length can be assessed. Attack activity can be also modelled to assess the security design of the protocols. An attack Agent has been developed to simulate the attack in Kerberos protocol, which retrieved the session key (*key\_AB*) and then contacted Agent A using this key, as shown in Figure 9. In the simulation, an extra window was opened to run the attack Agent. Once Agent A received the communication encrypted by *key\_AB*, the validation of the key has been checked as there is a lifetime for the key imposed by Agent S (Server) in Kerberos protocol. In the case, the attacker has used an out of date session key and therefore the verification has failed, as shown in Figure 10.

```
C:\MS0\Model>javac Agent_ATTACK_KP.java
```

```
C:\MS0\Model>java Agent_ATTACK_KP
```

Try to communicate with Agent\_A using key\_AB:

State 1: Retrieve key\_AB:

State 2: Generate a message to Agent\_A:

State 3: Encrypt the message using key\_AB:

State 4: Send the message to Agent\_A

Figure 9 Screen output of Agent Attack's execution in Kerberos Protocol

```

C:\MSc\Model>javac Agent_A_KP.java
C:\MSc\Model>java Agent_A_KP
Communication with Agent_S:
State 1: Generate Agent_A's message to Agent_S
Message sent to Agent_S: A, B
State 2: Send the cleartext message to Agent_S
State 3: Received encrypted Agent_S's message
State 4: Retrieve key_AS
State 5: Decrypt the message using key_AS
State 6: Verify the received messages: Lifetime L and B
Lifetime received from Agent_S, the key_AB is invalid after: 2002-12-07
15:06:36.154
Current time: 2002-12-07 15:04:57.638
Verification satisfied
Communication with Agent_B:
State 7: Generate Agent_A's message to Agent_B: {A, Ta}Key_AB
Timestamp Ta: 2002-12-07 15:05:01.326
State 8: Using key_AB to encrypt Agent_A's message
State 9: Send the messages including Agent_S's to Agent_B
Message sent to Agent_B: {Ts, L, key_AB, A}key_BS, {A, Ta}key_AB
State 10: Received Agent_B's reply message
State 11: Verification
Current time: 2002-12-07 15:05:02.622
Verification satisfied
Reply message received from Agent_B: 2002-12-07 15:05:01.357 + 1
State 12: Received further message encrypted by key_AB
State 13: Verification
Current time: 2002-12-07 15:07:03.06
Verification failed: the key is out-of-date!

```

Figure 10 Screen output of Agent A under attack in Kerberos Protocol

## CONCLUSIONS

Dynamical environment of Agent-based cryptographic protocols simulation system has been achieved using Java technology. The Java's networking and multithreading features have been used to establish communications between Agents in the protocol. The Java's security technology has been applied to generate keys (secret key and public key schemes), encrypt and decrypt messages.

A general formalism of Agent for simulation of cryptographic protocols has been proposed and implemented. Otway-Rees protocol has been successfully modelled using the Agent-based simulation system.

It was suggested that the Agent-based cryptographic protocols simulation system could have various application. The simulation system could be used to analyse, verify and assess design of the protocols, including correctness, weakness, reliability, efficiency of protocols. The simulation system could be also used to model attack activity.

## REFERENCES

- [1] [Http://www.ssh.com/support/cryptography/](http://www.ssh.com/support/cryptography/)
- [2] Michael Burrows, Martin Abadi and Roger Needham, A Logic of Authentication, SRC Research Report 39, February, 1990
- [3] Otway, D. & Rees, O. 1987 Efficient and Timely Mutual Authentication. Operating Systems Review Vol. 21, No.1, pp. 8-10.
- [4] Agent Technologies (<http://www.insead.fr/CALT/Encyclopedia/ComputerSciences/Agents/>)
- [5] Frantz O. Iwu, PhD Thesis, Manchester University, 2003.
- [6] Frantz O. Iwu and Richard N. Zobel, UK SIM Conference, Cambridge, April 2003.
- [7] B. P. Zeigler, Multi-Faceted Modelling and Discrete Event Simulation. Academic Press, 1984.
- [8] N. Su, Simulation In Cryptographic Protocol Design And Analysis, MSc Thesis, Department of Computer Science, University of Manchester, 2003.
- [9] Java2 SDK, Standard Edition, Documentation Version 1.4.0
- [10] [Http://www.bouncycastle.org/index.html](http://www.bouncycastle.org/index.html)

**NING SU** obtained his BSc and MSc in Engineering Mechanics at Zhejiang University and Xian Jiaotong University in China respectively. He got his PhD in Civil Engineering at the University of Dundee and MSc in Computer Science at the University of Manchester in the UK. He is currently working for LaChine LinCom Limited and his interests include Java technology, distributed simulation and network security.

**RICHARD ZOBEL** graduated with BSc(Eng) (London) in 1962, and with PhD (Manchester) in 1970. He is a C.Eng, and Member of both BCS and IEE. He has 118 publications, including a book and a patent, and has supervised 100 postgraduate students. He retires fully from Manchester University Computer Science Department end October 2003 after 37 years service as Lecturer and Senior Lecturer. He is a Member of UKSim, and of SCS from whom he received an Outstanding Service Award in 2002. His interests are currently distributed simulation, distance learning and network security.

**FRANTZ IWU** is a research associate at the University of York. He obtained his MSc in Advanced Computer Science and his PhD from Manchester University, United Kingdom. His research interest includes distributed simulation systems, security for distributed simulation under commercial network protocols, application of HAZOP to computer systems, system safety analysis, software flow failures and fault propagation, systematic development of large software systems from reusable fragments using the B-Method. He is a member of the British Computer Society.

# SIMULATION ISSUES OF OPTICAL PACKET SWITCHING RING NETWORKS

Marko Lackovic and Cristian Bungarzeanu  
EPFL-STI-ITOP-TCOM  
CH-1015 Lausanne, Switzerland  
{marko.lackovic;cristian.bungarzeanu}@epfl.ch

## KEYWORDS

Discrete event simulation, message exchange, packet switching, ring.

## ABSTRACT

The paper focuses on discrete event simulation of a ring topology based on optical packet switching. The cell loss ratio was used to compare two simulation mechanisms. The first one utilizes the message exchange domain, while the other represents a message exchange simplification as the traffic for each node is generated separately. Results have been verified using the analytical procedure and compared to justify the use of the simplified model. It has been shown that the influence of message exchange mechanism on traffic pattern introduces changes to the calculated performance which cannot be included in the analytical or the simplified simulation model.

## INTRODUCTION

A large number of telecommunication network evaluations by using simulation is based on evaluating just one node and drawing conclusions from these results. If the whole network's performance is to be evaluated by simulation, this model can be easily generalized by simulating each node independently. This simple model doesn't take into account changes introduced by intermediate nodes which serve traffic going from source to destination. Aggregation due to space switching and buffering in those nodes can significantly change traffic pattern. This is of the most importance, because the existence of traffic burstiness, or periods where packets arrive in a stream can impact the rejection ration in limited size buffers (Lackovic et al. 2003).

The aim of this work is to compare two simulation models of the same network. The first one is a simple generalization of a single node simulation, and regards each node independently. This is a simplification of the second simulation model, which uses the message exchange domain, and thus incorporates changes in traffic flows introduced by buffering and aggregation in

nodes. Simulation results are compared to analytical procedure in order to justify the use of the simplified simulation model, and to determine the possible restrictions in cases where obtained results differ too much.

The network is based on the optical packet switching paradigm (Yao et al. 2000), which is considered to be a long term solution for the broadband optical networks. Fine packet level granularity combined with WDM and intelligence introduced in optical nodes are giving the answer to the increasing demand for QoS aware increasing communication demands.

## ASSUMPTIONS

A ring topology (Figure 1) has been chosen to compare simulation and analytical results, because of transparent traffic demand structuring and parameterization.

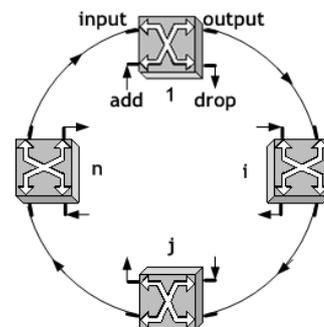


Figure 1: Network Structure

Each node has an optical packet switching capability. Switch is structured as an unblocking space switch with output buffering and full wavelength conversion. Nodes are capable of inserting (add) and extracting (drop) traffic from the ring (Figure 1). A cell based synchronous communication model was chosen, implying exchange of fixed sized packets (cells) in fixed time points (slot beginnings).

Traffic demands are equal between any node pair implying that each node generates the same traffic volume for each node in the network. This results in the same traffic load on all links in the ring.

The cell loss ratio (CLR) was chosen as a communication evaluation property. The goal was to determine CLR dependency on topological parameters (number of stations), network parameters (number of wavelengths) and node structure (memory capacity). The analytical procedure and simulation will be used to calculate these dependencies.

Figure 2 depicts general node model. Input traffic flows are demultiplexed (demux section) and their wavelength is adjusted to match a free wavelength in the appropriate fiber delay line, or a free wavelength on the output (cell encoding section). Cell encoding section comprises tunable wavelength converters. A control unit determines the output wavelength according to the information extracted from packet headers, and information on the occupation of the buffer on the required output. Set of internally used wavelengths is the same as the set of input/output wavelengths. Therefore wavelengths used for potential cell buffering correspond to output wavelengths. Switching section does the space switching to the appropriate output. Switching is done on the demultiplexed packet level. Outputs are multiplexed and sent to the chosen fiber delay line.

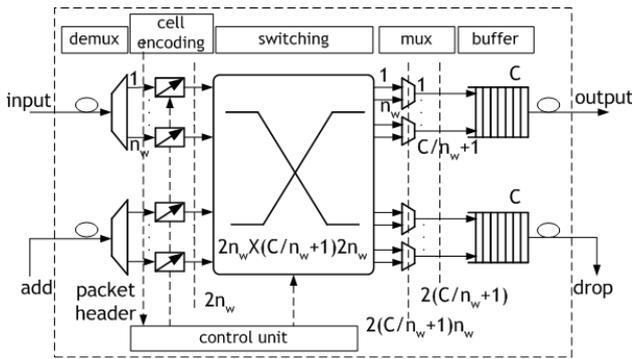


Figure 2: General Node Model

Each node output is buffered using fiber delay lines. Each buffer always contains a direct connection to the output which doesn't introduce any delay. Buffer capacity expressed in the number of cell that can be stored ( $C$ ) is determined by the number of FDLs that introduce a delay:

$$C = n_{FDL} n_w, \quad (1)$$

where  $n_{FDL}$  stands for the number of delay lines (not counting the direct connection), and  $n_w$  for the number of wavelengths used in the buffers. This number is equal to the number of wavelengths used in the network.

### ANALYTICAL MODEL

Analytical model (Lackovic and Bungarzeanu 2003) is based on the Markov chain describing number of cells in each buffer. Each Markov chain state represent a number of cells in one buffer. Each transition represent the change of the number of cells in a buffer (Figure 3).

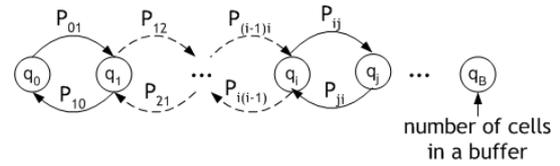


Figure 3: Markov Chain for one Buffer

For the CLR calculation the probability of each state (number of cells in a buffer) has to be calculated. This calculation is based on calculating the probability that a determined number of cells will arrive to a buffer in one time slot

### SIMULATION MODEL

Simulation was performed on a simplified model, but with all characteristics that influence the network performance in terms of CLR. These include full wavelength conversion on all inputs and output buffering on all outputs. Switching is performed by strictly unblocking space switch.

This simulation model is based on the discrete event simulation. Modelling was performed using object-oriented paradigm of the *Cosmos* tool (Lackovic and Inkret 2001). Figure 4 depicts a class taxonomy of the packet switching simulation. The implemented model is generalized implying that any topology can be structured and analyzed/simulated. The system contains (inherits) basic structural and discrete event simulation properties from the *Cosmos* base classes. It uses network algorithms for network/demand structuring. Base module contains structural properties (Lackovic and Inkret 2002) and the message exchange domain properties (ability to send/receive messages). Classes implementing concrete network components have been inherited from the base module class.

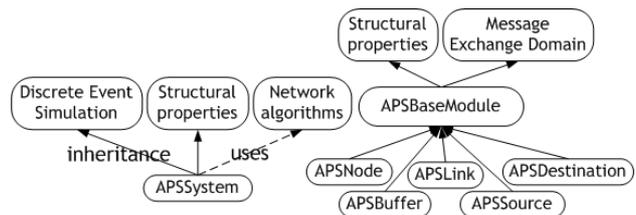


Figure 4: Class Taxonomy

### Cell Generation

The simplification of the cell generation model is important as it can introduce a considerable simulation speed-up. Large number of simulation iteration is required, as large number of time slots have to be simulated to obtain satisfactory CLR value accuracy, or to obtain any CLR value in the case of very rare cell rejections.

Cells are generated on each channel (wavelength) independently using a binomial distribution with the probability of cell generation equal to the channel load.

All channels on the same fiber/link have the same load because of the load balancing of all traffic on one fiber/link over its channels. Cell generation algorithm can be described as follows:

```

for all wavelengths on a fiber
  generate randomly  $0 \leq n < 1$ 
  if  $n \leq$  channel load
    randomly select a demand
    generate cell for the demand
    switch the cell to the output buffer
    if buffer not full
      store cell in buffer
    else
      discard cell
      notify demand/link
    end if
  end if
end for

```

Two different simulation models regarding cell generation have been defined. The first called independent traffic generation (ITG) generates cells using described method on all channels on all links, while the other utilizes message exchange domain (ME) and generates cells only on source links (connecting source and switch). These cells are exchanged between nodes using message exchange mechanism (Lackovic and Inkret 2002).

### Independent Traffic Generation

This method represents a network simulation conducted as a generalization of a single node simulation. A lot of studies have been focused on just one node simulation due to the long simulation execution time needed to obtain feasible results. A way to shorten execution time is to regard each node as an independent simulation entity. Its connection to the other nodes is realized through analytical determination of link loads on its inputs and outputs. In each iteration (time slot) a cell generation function is called for all fibers on all links. The speed-up is achieved by the memory-less simulation because no real packets are exchanged in the network.

After each iteration a release function is called to release appropriate number of cells from each buffer:

```

for all buffers
  if cell number  $< n_w$ 
    remove all buffered cells
  else
    remove  $n_w$  cells
  end if
end for

```

This exchange of generate and release function calls is a simulation of the bourn and dying processes of the Markov chain. Figure 5 depicts a simple scheme of basic generate and release model. The part in dotted lines is omitted in this model. Markov states are determined by the number of cells in each buffer.

### Message Exchange Model

Message exchange based model assumed cell generation only on network sources (source links). Generation of cells on ring links is substituted by the

message exchange mechanism. Generate function is modified to:

```

for all wavelengths on source fiber
  generate randomly  $0 \leq n < 1$ 
  if  $n \leq$  channel load
    randomly select a demand
    generate cell for the demand
    switch the cell to the output buffer
    if buffer not full
      store cell in buffer
    else
      discard cell
      notify demand/link
    end if
  end if
end for

```

Message exchange function can be defined as:

```

for all buffers
  send  $n_w$  cells to egress node
  switch cells to the output buffer
  if buffer not full
    store cell in buffer
  else
    discard cell
    notify demand/link
  end if
end for

```

Release function is the same as in the previous case.

Basic generate-release model is not appropriate for the message exchange simulation because it gives advantage to some demands in terms of priority. The order of release function calls determines whether a node will receive cells by the message exchange mechanism before or after it has released its cells from the buffer. The cells generated on the source link in this simplified model will always arrive before or after the cell release and incoming exchanged cells what would produce too large or too small CLR for source links. Therefore the generate-release model has to be adjusted to eliminate the possible cell discrimination. Figure 5 with the buffer in dotted lines depicts a modified fair generate release model with added pre-buffering which using an order randomizer assures that no cells will be discriminated.

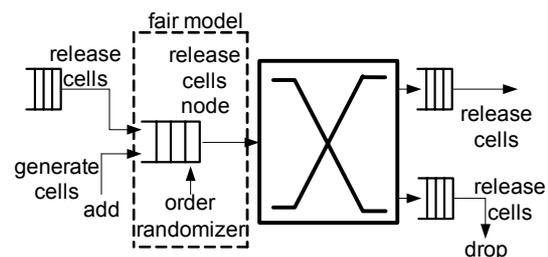


Figure 5: (Fair) Generate-Release Model

Modified generate function can be defined as:

```

for all wavelengths on source fiber
  generate randomly  $0 \leq n < 1$ 
  if  $n \leq$  channel load
    randomly select a demand
    generate cell for the demand
    store the cell in input buffer
  end if
end for

```

Modified message exchange is equal to:

```

for all buffers
  send  $n_w$  cells to egress node
  store cells in input buffer
end for

```

Release function also changes, as it includes the release of the pre-buffered cells, what is the actual cell generation for the switch inputs:

```

for all input buffers
  randomize buffer order
  switch cells to the output buffer
  if buffer not full
    store cell in buffer
  else
    discard cell
    notify demand/link
  end if
end for

```

### CALCULATIONS

Calculations include link and demand CLR by using analytical procedure and simulation. Simulation results were obtained by the ITG and ME simulation to verify the simplified simulation model. Both nodes with no buffering and with buffering capabilities have been taken into account.

#### No Buffering

Analyzed ring comprises 5 nodes without buffering capabilities. A 4 wavelength WDM system has been used to make the calculation shorter.

Figure 6 depicts dependency of the short demand CLR on the ring load. The graph contains analytical results (A), simulation results done by ITG (S), and simulation results obtained by ME simulation (S(ME)). A good match was obtained on smaller ring loads, but on the loads above 0.7 the difference between ME simulation results and those obtained by analytical procedure and ITG simulation grows. It is interesting to notice the change of relative difference sign between A/S and S(ME) results. S(ME) results are larger than A/S for smaller loads, but become smaller for larger ring loads.

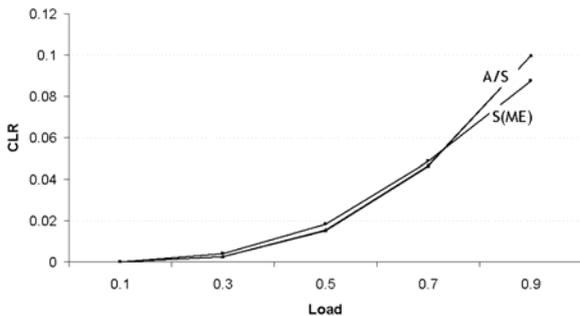


Figure 6: Short Demand CLR (1 Hop, no FDL)

Figure 7 depicts the longest demand CLR. Demand uses four ring links. The trends are the same as in the case of short demand with the visible differences between A and S results for the large ring loads.

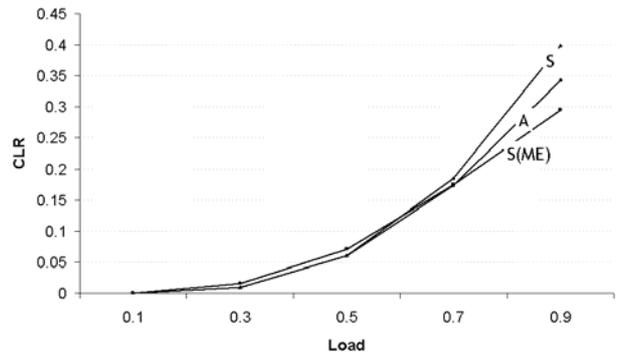


Figure 7: Long Demand CLR (4 Hops, no FDL)

Difference between A and S results is the consequence of inaccuracy introduced by the ITG simulation. Larger CLR for longer demand is caused by more links used by the large CLR. As the cells are not actually exchanged in the simulation, there is no continuity of communication in the network which would reflect the fact that the longer CLR transverses more links. This is implicitly assumed by the cell generate function. As all cells are generated independently on all channels, there is larger probability to generate a cell belonging to the demand that transverses larger number of links. If the cells for some demand are generated more often, the number of their rejections increases. The term cell generation has to be taken conditionally, because those cells just exist in the current simulation iteration (slot). Figure 8 depicts a case of 5 nodes in the ring. As all the demands have the same capacity, the probability of generating a cell for demand 1->5 will be 4 times larger than generating cell for the demand 1->2.

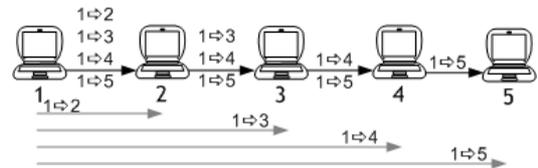


Figure 8: Influence of Generate Function on Long Demand CLR

Average link CLR is shown in Figure 9. The same conclusions as for short demand apply here.

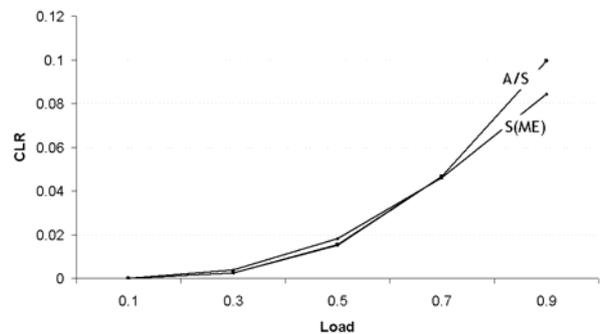


Figure 9: Average Link CLR (no FDL)

Smaller CLR values for the ME simulation for large ring loads can be explained by examining the link loads

in the case of analytical procedure, ITG and ME simulation. Analytical procedure assumes calculated link loads, just like the ITG simulation. As all cells are generated independently, fluctuations introduced by buffering and traffic aggregation which occurs on the output switch ports cannot be taken account. This is especially visible for large ring loads, where the large CLR values (even up to 50%) influence the link load after buffer. Figure 10 depicts differences in link loads introduced by ME simulation. Simulation load is the analytic load influenced by rejection in aggregation points/buffers. The simulation load is thus present in simulation after the buffering, but only in the ME simulation this load is the actual load that enters the next node. In the ITG simulation the egress node load doesn't influence the next node, because the ingress traffic for the next load is generated independently according to the average link load obtained by analytical calculation.

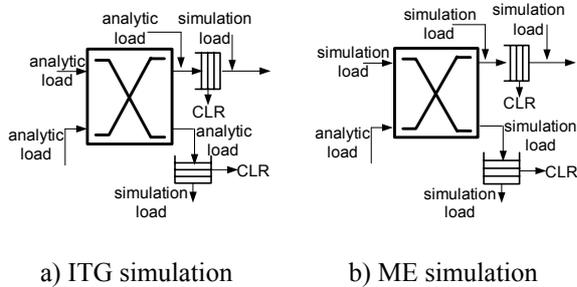


Figure 10: Analytic and Simulation Load

Figure 11 depicts the link loads obtained from simulation. Two links have been taken into account. The A1 link is the access link in the node 1, while the 1->2 link is the ring link connecting nodes 1 and 2. It is visible that the A1 link has the same load in ITG and ME simulation, while the difference between 1->2 link loads grows with the ring load increase. This is the consequence of growing CLR.

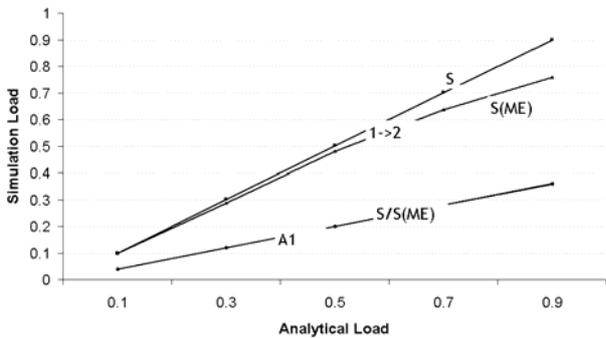


Figure 11: Mean Simulation Link Load (no FDL)

**Buffering**

The second calculation group focuses on nodes with buffering capabilities. Analyzed network has 5 nodes and 4 wavelengths with 1 FDL buffers. Figure 12 -

Figure 14 depict the same calculations as in the previous case.

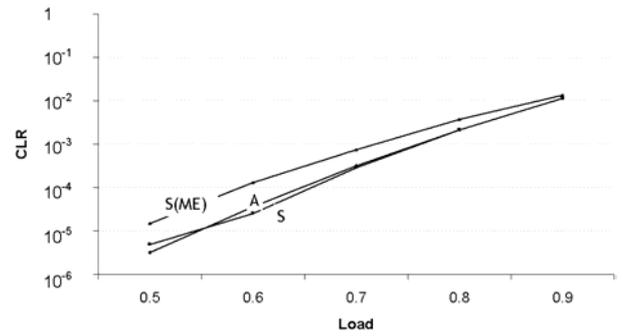


Figure 12: Short Demand CLR (1 Hop, 1 FDL)

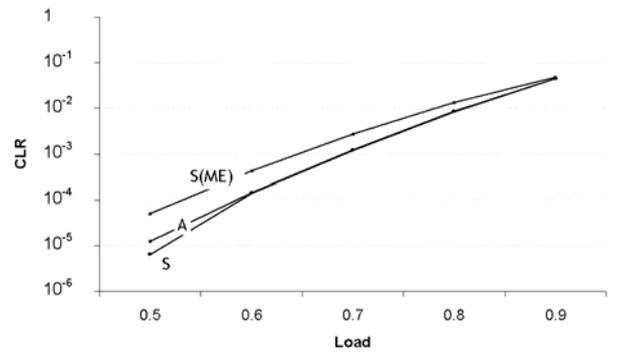


Figure 13: Long Demand CLR (4 Hops, 1 FDL)

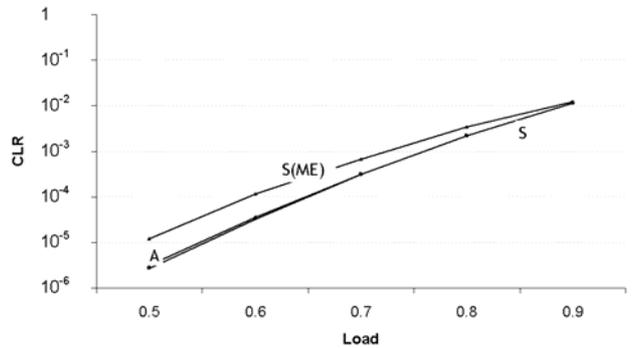


Figure 14: Average Link CLR (1 FDL)

A good match is obtained between analytical and ITG simulation results on almost all loads. The difference on large loads for long demand, which was present for the no buffering case, is now suppressed by small rejection probability which eliminates the described inaccuracy of the ITG simulation for longer demands. The difference exist for small loads where the simulation inaccuracy is caused by the very rare cell rejection events.

Difference for the ME simulation are reflection from the previous calculation group where ME simulation produced larger CLR values for the lower ring loads. In this case the CLR considerably influences the simulation link load. As the simulation load is equal to the analytical load, there is no influence on the CLR which was present in the previous calculation group.

Only the buffering and aggregation issues on the switch outputs influence the CLR. Larger CLR result can be explained by changes in the traffic characteristics imposed by nodes. These changes are not present in the ITG simulation as traffic is generated in each slot, and there is no influence of one slot to the other, except in the number of buffered and rejected cells. Figure 15 depicts histogram of arrived number of cells in a buffer. It is clear that the distribution of cell number changes with the ME mechanism introduction. The number of slots with larger number of incoming cells increases, what increases the rejection probability and CLR. Figure 16 depicts histogram of buffered cells with the number of cells which are going to be directly transmitted. An increase in probability of larger number being buffered is visible. Large number of buffered cells increases the probability of cell rejection in the next slot.

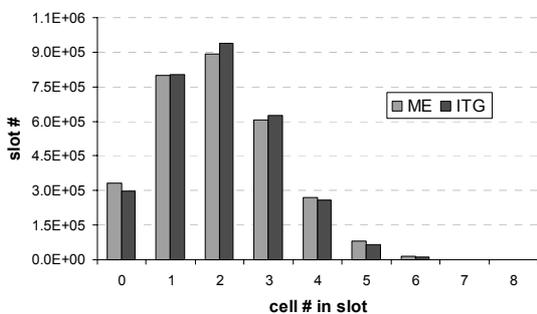


Figure 15: Arrived Cell Number Histogram (4 wl, 1 FDL)

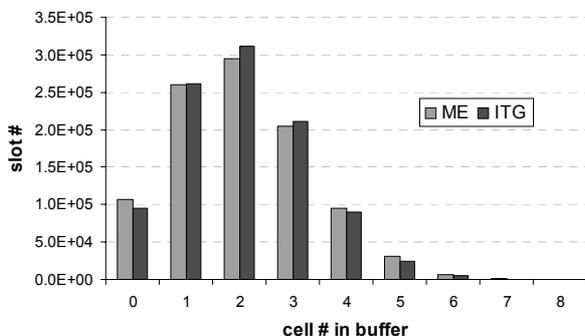


Figure 16: Buffered Cell Number Histogram (4 wl, 1 FDL)

## CONCLUSION

This work was focused on investigating the properties of the packet switched uniform ring topology. PSUR nodes have the optical packet switching capability. Ring links are equally loaded due to the unidirectional communication and same traffic demands between all ring nodes. CLR was chosen as the performance evaluation criteria. CLR calculation was performed using analytical procedure based on the Markov chain, and discrete event simulation. Simulation was based on

independent generation of traffic (ITG) for each node, and on message exchange simulation (ME).

The CLR results for all the cases were compared in order to verify the simulation model, and to evaluate the simplification introduced by the ITG simulation. A good fit between ITG simulation and analytical procedure results was achieved. The ME simulation produced larger CLR values, showing the CLR underestimation by other methods. This difference was caused by changes in traffic characteristics which cannot be taken into account by analytical procedure and ITG simulation. These procedures produced CLR overestimation on very large ring loads and no buffering capabilities. In those cases the CLR becomes very (unrealistically) large and affects effective (simulation) load, which becomes smaller than the analytically calculated load. These findings show that the simple generalization of the simulation of one node to the network simulation is sometimes not good enough, as it does not take into account the influence of nodes on the traffic model.

## ACKNOWLEDGEMENTS

This work has been conducted within the Cost 266 Action and with the support of the Swiss Federal Office for Education and Science.

## REFERENCES

- Lackovic, M.; B. Mikac; and V. Sinkovic, "Network Performance Evaluation by Means of Self Similar Traffic Model", In *Proceedings of Mipro* (Opatija, Croatia, May 19-23, 2003), 82-87.
- Yao, S.; B. Mukherjee; and S. Dixit, "Advances in Photonic Packet-Switching: An Overview", *IEEE Communication Magazine*, February 2000, 84-93
- Lacković, M.; and C. Bungarzeanu, "Planning Procedure and Performance Analysis of Packet Switched All-optical Network", In *Proceedings of ONDM* (Budapest, Hungary, February 3-5, 2003), 253-271.
- Lackovic, M. and R. Inkret, "Network Design, Optimization and Simulation Tool Cosmos", In *Proceedings of WAON*, Zagreb, Croatia, June 13-14, 2001), 37-44.
- Lackovic, M.; R. Inkret; and B. Mikac, "An Object-oriented Approach to Telecommunication Network Modeling", In *Proceedings of ESM*, June 3-5, 2002, Darmstadt, Germany

## AUTHOR BIOGRAPHIES

**MARKO LACKOVIC** was born in Zagreb, Croatia where he obtained an electrical engineering degree at the Telecommunication department of Faculty of Electrical Engineering and Computing in 2002. He worked there as an assistant before moving as a PhD student to Telecommunication laboratory at EPFL. His research interests include optical transmission network design, simulation and optimization techniques. He is one of the authors of the object oriented simulator *Cosmos*. His e-mail address is marko.lackovic@epfl.ch and his Web-page is <http://itop.epfl.ch/TCOM/Membres/Lackovic/>.

# CALCULATING THE VOLTAGE ACROSS A TURNED OFF SEMI-CONDUCTOR MODELLED BY PERFECT SWITCHES

Faouzi BOULOS  
Department of Physics  
University Lebanese of Science II  
Fanar BP 90656, Jdeidet-El-Maten, Lebanon  
Email: faboulos@ul.edu.lb

## KEYWORDS

Semi-conductor modeling, power electronic circuits, numerical simulation, variable topology method.

## ABSTRACT

The aim of this paper is to describe a numerical simulation method of power electronic circuits. The problem of the method is due to the fact that the semi-conductors are simulated by perfect switches. That is two nodes connected by a conducting semi-conductor are merged, and two nodes connected by a turned off semi-conductor have their branch removed, sometimes leaving some nodes pending. The branches terminated by pending nodes are also removed. A method, which will be shown further is based on a topological analysis of the circuit allowing the extraction of the relations and the calculation of the voltage across a turned off semi-conductor. The method consists of simplifying the initial figure of the circuit and setting up the mathematical formulation and development of the automatic simulation.

## INTRODUCTION

The choice of the electric model of the semi-conductor is thus very important. There are several ways to represent the semi-conductor which are all electrically equivalent (Antognetti and Massobrio 1988). Similarly, all semi-conductors may be represented by equivalent electric circuits more or less complex, which can be used in the electronic circuit simulation for example (Nagel and Pederson 1973). Some of the most frequently used models are:

binary variable resistance, depending whether the semi-conductor is "off" (high resistance) or "on" (low resistance). Controlled voltage source or current source (Laktos 1979), inductance in series with a parallel RC circuit (Rajagopalan 1978), inductance and resistance (Eisenar and Hofmenister 1972).

In these cases the topology is fixed, the circuit and the size of the calculating matrix is fixed. Another method is to represent the semi-conductor by replacing it by an open circuit when it is "off" and by a short circuit when it is "on". Thus, the topology is variable. Each condition has its set of equations, which reduces the size of the calculating matrix considerably (Boulos 1988). On the other hand in the variable topology method the semi-conductors are simulated by perfect switches, the circuit

and the equation system are variable (Boulos 1988). The difficulty is due to the fact that in such a method the semi-conductors are simulated by perfect switches, that is two nodes connected by a conducting semi-conductor are merged and two nodes connected by a turned off semi-conductor have their branches removed, sometimes leaving some nodes pending. The branches terminated by pending nodes are also removed. This point is resolved by an algorithm (Boulos 2001); that makes it possible to modify the complete topology of the circuit at each sequence of operation. This makes the knowledge of the current in a conducting semi-conductor and the knowledge of the voltage across a turned off semi-conductor difficult to obtain. In a previous publication (Boulos 2002), I have presented a solution for the method based on a topological analysis of the circuit allowing the extraction of the relations calculating the current in a conducting semi-conductor. In this paper I present solutions for the extraction of the relations and the calculation of the voltage across a turned off semi-conductor.

## THE FIXED TOPOLOGY METHOD

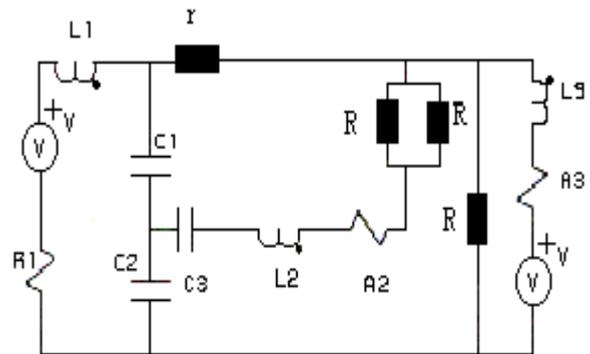


Figure 1: The semi-conductor is "off" (high resistance R) or "on" (low resistance r).

In this type of simulation, the circuit is fixed and the equation system is unique, yet, some values of the coefficients may change according to the operating point of the semi-conductors. They are considered high resistance when they are "off" (R) and low resistance when they are "on" (r). For the principal function, see figure 1.

## THE VARIABLE TOPOLOGY METHOD

In the variable topology method, the semi-conductors are simulated by perfect switches, shown in figure 2. When the semi-conductor is "on", we link the two nodes by a semi-conductor, when the semi-conductor is "off"; we take off the link between the two nodes.

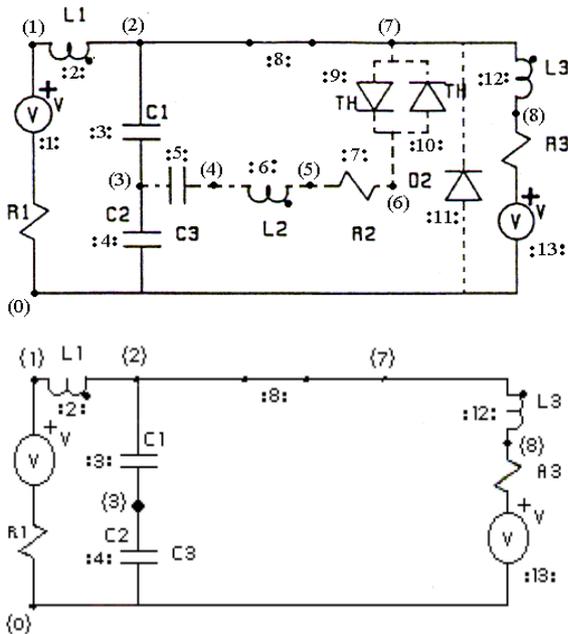
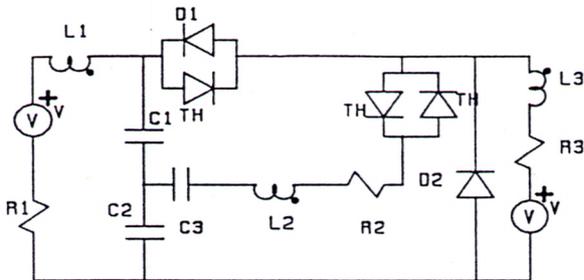


Figure 2: The semi-conductors are simulated by perfect switches.

At each step, the topology of the electric circuit is simplified compared to the original circuit. The number of differential equations is lower than the one obtained by the fixed topology simulation. Main difficulty of the general variable topology method: this method makes the knowledge of the voltage across a turned off semi-conductor difficult to obtain. The algorithm resolves this point it is possible to modify the complete topology of the circuit to each sequence of operation.

That is to say the figure 3 with:

(E) Is the whole of information represented tables and which relates to the configuration of the circuit (standard elements, numbers of the branches and nodes, state of the semiconductors, value of the voltage of all the capacitive branches and generators of voltage and value of the current of the inductive branches and generators of current). After passage of this information in “

TOPOVAR “, we obtain at the exit the information used during all this phase of operation as follows:

- (S1) which represents the form of minimal topology with a new classification and which allows us the setting in automatic equation.
- (S2) which is a whole of vectors and tables enabling us automatically to know the blocked value of the terminal voltages of the semiconductors.
- (S3) which is a whole of vectors and tables allowing us to know the value of the currents automatically which crosses the conducting semiconductors.

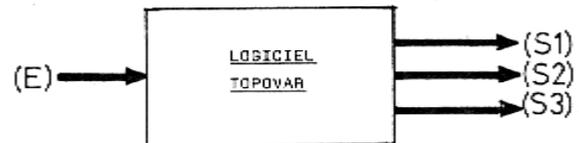


Figure 3: The “TOPOVAR”.

The main steps of this algorithm

1. Remove automatically all turned off semi-conductors and all branches leading to pending nodes.
2. Short circuit conducting semi-conductors.
3. Determine automatically at each step the new number of all the nodes that are either anode or cathode of turned off semi-conductors in order to know the voltages across those semi-conductors.
4. Determine automatically at each step the number of the branches connecting the anode nodes, and the number of the branches connecting the cathode nodes of a conducting semi-conductor, in order to know the current in the semi-conductor.
5. Determine automatically the form of the minimal topology. This topology contains a new number of nodes and branches. After applying the algorithm, only passive elements such as resistances, self-inductances, capacitors, voltage and current generators remain present in the structure figure 2.

## GENERAL PRESENTATION OF THE METHOD SUGGESTED

From the data (topology and components) and for each phase of operation, it is necessary to determine minimal topology, the new value of the nodes of entry and exit of the semiconductors after removal of the branches traversed by a null current and the value of the voltage for each blocked semiconductor.

## AUTOMATIC REMOVAL OF THE BLOCKED SEMICONDUCTORS AND OF THE BRANCHES ENDING BY PENDING NODES

Consider a complete circuit containing a certain number of active and passive branches. For each blocked semi-conductor, one removes the corresponding branch. Then the nodes of input and output are separated, which gives a circuit with two parts: a side node for the input, and another for the output figure 4. The main problems to be solved at this stage are:

- 1) Determination of the new numbers of nodes of input and output for each blocked semiconductor (and removed) when there is removal of branches in series, terminated by pending nodes.
- 2) Determination of the value of the terminal voltage of each semiconductor removed. This value can depend on the voltage of the removed branch if it is capacitive or if it contains a voltage source.

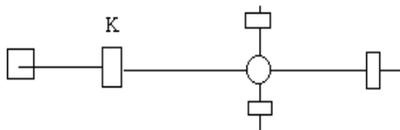
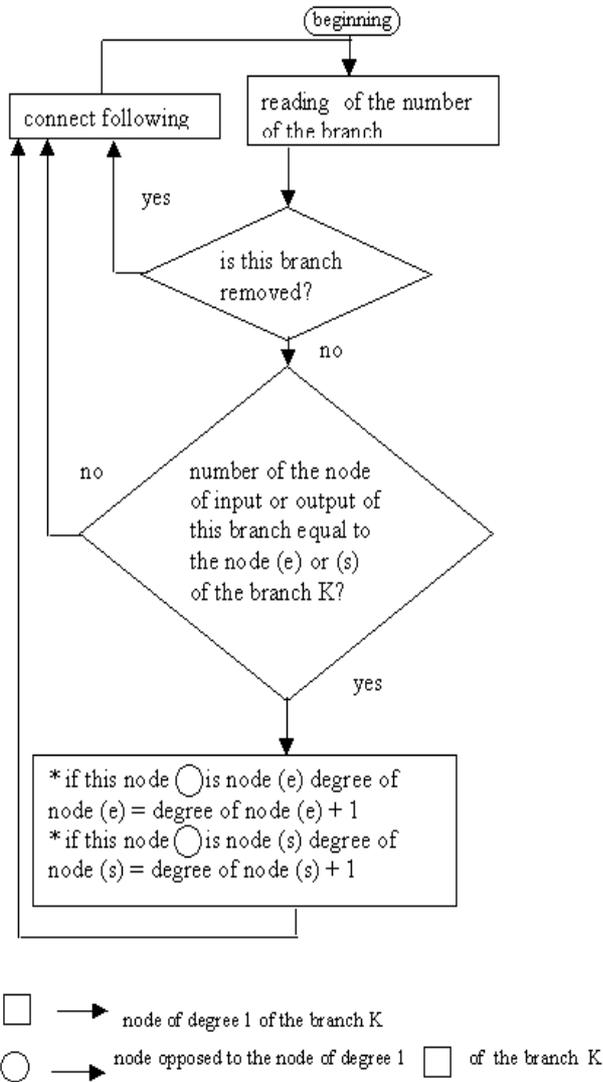


Figure 4 : The flow chart above presents the search for degree of the node opposed to the branch K connected to a node of degree 1.

### AUTOMATIC DETERMINATION OF THE NEW NODES OF INPUT AND OUTPUT FOR EACH REMOVED BLOCKED SEMICONDUCTOR

This is done in a first step analysing the circuit branch by branch and remove every branch containing a blocked semiconductor. In a second step the process is repeated checking we seek if the output of this branch is equal to the input node (e) of the blocked semiconductor.

- a) If so, we keep the number of this branch and assign a degree for the node (e) = degree of the node (e)+1.
- b) If not, we seek if the input of this branch is equal to the node (e)

-If yes we keep the number of this branch and we assign a degree for the node (e) = degree of the node (e) +1.  
- If not, we move to the next branch.

The same process is repeated for the output node (s). If the degree of the node (e) or the node (s) is equal to 1, the branch that connects this node is pending and this branch is removed. When the degree of the node is not equal to 1, this node is the new number for this blocked semi-conductor which is stored in the matrix G (X, Z).

### AUTOMATIC DETERMINATION OF THE VOLTAGES ACROSS BLOKED SEMI-CONDUCTORS

The objective is to determine the voltage across each blocked semiconductor. It should be noted that the search for the new input and output nodes is not sometimes sufficient to know the variation of the voltage across the removed blocked semiconductors for each branch containing a capacitor or a voltage source contributes to the determination of this voltage. For that, we will show how, starting from the removed passive branch we calculate the value of the voltage across each blocked semiconductor. We start by defining a function f (X, Y) which determine the terminal voltage of each blocked semiconductor.

$$f(X, Y) = VC(X, Y) + VE(X, Y) \quad (1)$$

With:

X = number of branch containing a blocked semiconductor,

Y = 1 or 2 according to whether one works on the corresponding node output or the node input of the blocked semiconductor at each instant of time, we are able to define the values of the functions f (X, 1) and f (X, 2). These two functions allow is to know the voltage across each blocked semiconductor by applying the following relation:

$$F(K) = (U(G(K,2)) + f(K,2)) - (U(G(K,1)) + f(K,1)) \quad (2)$$

With:

F (K): Voltage across the blocked semiconductor numbered K.

U (G (K, 2) + f (K, 2) Voltage of the input node (E) of the blocked semiconductor compared to the reference node (O)

U (G (K, 1) + f (K, 1) Voltage of the output node (S) of the blocked semiconductor compared to the reference node (O). It should be noted that at each step of

calculation the value of the voltage of each blocked semiconductor must be calculated.

We study now the construction of matrices VC (X, Y) and VE (X, Y).

### Case of the capacitive branches VC (X,Y).

Each capacitive branch connected to a node of degree 1 is removed. The value of the voltage across these terminals is stored in VC (X, 1) for output node of the blocked semiconductor numbered X and in VC (X, 2) for the input node of this blocked semiconductor.

For example:

$$- VC(X, 1) = VC(X, 1) + F(K) \quad (3)$$

$$- VC(X, 2) - VC(X, 2) + F(K) \quad (4)$$

During a whole phase of operation, the values of VC (X, 1) and VC (X, 2) remain constant. However, for each new phase of operation the matrix is initialized with zero value. We analyse in order to better understand the step to study the example of the chopper drawn figure 2. We are interested in the phase of operation during which thyristors of branches :9: and :10: as well as the diode of branch :11: are blocked whereas the switch of branch :8: is conducting and the capacitor of branch :5: is pending. The voltage across its terminals acts on the input node of branch :10: and the output node of the branch :9: . We will seek for the blocked semiconductor numbered 9 of which its side of the node of entry or node of exit depends the tension of the capacitive branch number :5: is pending. We find that it is the node of exit of the branch :9: that takes the value of the node of entry of the capacitive branch numbered :5:., then  $VC(9,1) = VC(9,1) + F(5)$ .

For the blocked semiconductor of the branch :10:, it is the node of entry which becomes equal to the node of entry of the capacitive branch numbered :5:.

Therefore:  $VC(10,2) = VC(10,2) + F(5)$

### Case of the voltage source

The difference between a capacitive branch and a branch containing a voltage source comes owing to the fact that the tension of a voltage source can vary as a function of time. Thus, its value is  $F(K) = VM(K)$  where VM(K) is the amplitude of the voltage source. It is thus necessary to recompute the voltage source for each step of calculation and to store it in VE(X,Y).

## AUTOMATIC DETERMINATION OF THE VALUE OF THE CURRENTS THAT CROSS EACH REMOVED CONDUCTING SEMI-CONDUCTOR

The current in a semiconductor is the algebraic sum of the currents of the branches that connect the input node or the output node. It is to be noticed that the knowledge of the information stored in matrices MAENT (X, X1), MASORT (X, X1), SIGC (X, X1) enable us automatically to know the value of the currents which at

every moment cross all the conducting semiconductors of calculation. For that, we apply one of two following equations:

$$ISC(X) = ISC(X) + SIGC(X,X1) * IBB(MASORT(X,X1)) \quad (5)$$

$$ISC(X) = ISC(X) + SIGC(X,X1) * IBB(MAENT(X,X1)) \quad (6)$$

With:

ISC(K) Vector containing the value of the current of the conducting semiconductor number K.

SIGC(X1,X2) Matrix containing the values of the signs of the current that cross the branches connected to the input node or the output node of a conducting semiconductor. (n) Number of node and :n: number of branch.

MAENT(X1,X2) Matrix containing the numbers of the branches related to a conducting semiconductor on the side of the input node with: integer variable X2 varying from 0 with the maximum number of branches connected to this node.

MASORT(X1,X2) Even definition that MAENT(X1,X2) but for the output side.

## FORMING AND SOLVING THE EQUATIONS

In order for the program to be automatic, it must establish the equations by itself. This is achieved in two steps:

the first step consists of the topological study of the circuit, and the second step consists of establishing the equations. An electronic circuit is represented by a group of branches connected together and the equations are established using mathematical techniques. This enables the analysis of a great number of different circuits, for which the formation and solving of the equations would otherwise require long and tiring calculations. It is not possible to list all mathematical techniques and numerical calculations in this article. Only the most frequently used methods will be mentioned, without going into detail (Chue L.O. and Lin P.M. 1975) and (Pelletier 1971): nodal method, state variable method.

For the application, we chose the setting in equation by the nodal method for it asks less memory.

### Nodal analysis method

Several simulators use this method, which is based on the solution of the matrix associated to the equations representing the electronic circuit. The matrix is solved by the method of the pivot. SPICE (Antognetti and Massobrio 1988). The nodal analysis method is used by several simulators because it is simple and powerful. In this method, the equation system is in the form of:

$$(I) = (G) * (V) \quad (7)$$

(I) Vector of electrical current.

- (V) Vector of the voltages of the nodes compared to the reference nodes.
- (G) Matrix of the admittance.

**THE PRINCIPAL OF THE SIMULATION SOFTWARE VARIABLE TOPOLOGY METHOD**

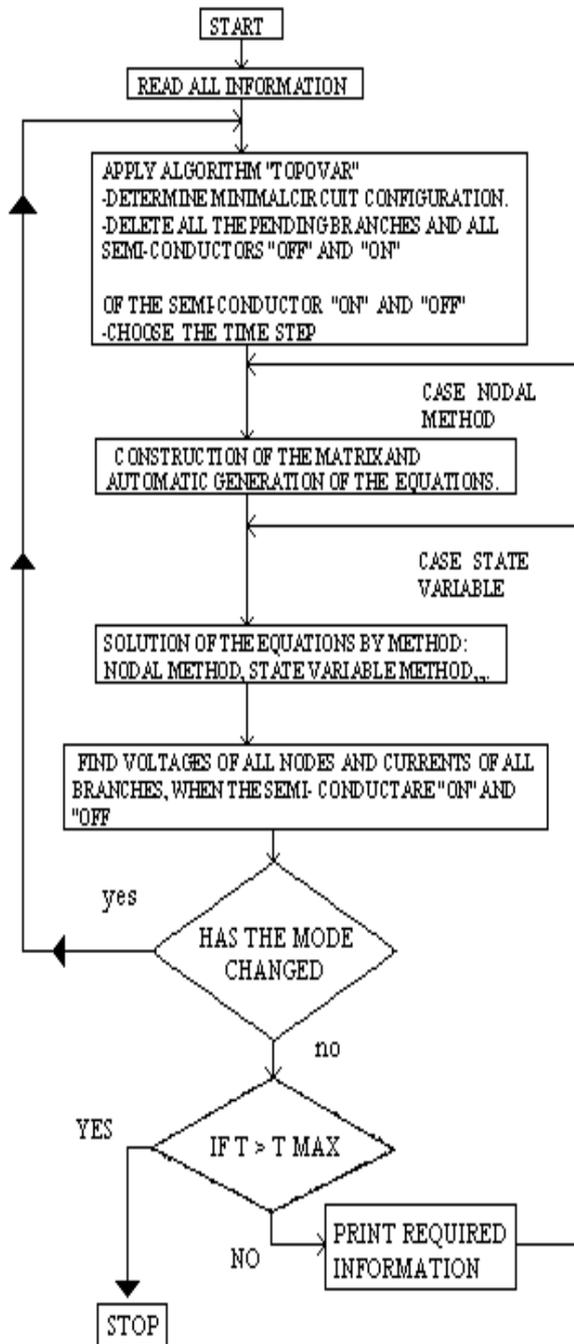


Figure 5: General flow-chart for variable topology.

**RESULTS**

In order to illustrate the mentioned methods, the results obtained by simulating a chopper figure 6 are shown.

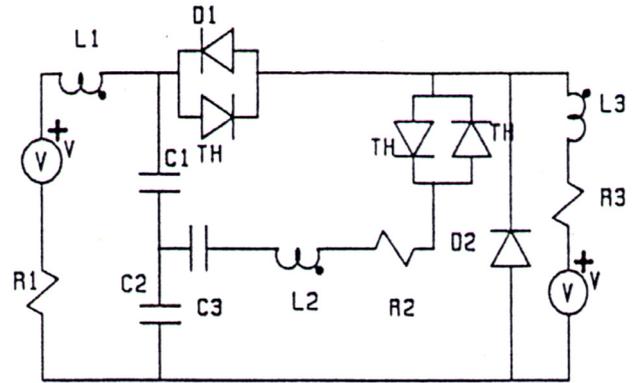


Figure 6: Schema of the simulated chopper.

The states of the semiconductors are represented by 0 for the semiconductors that are “off” and 1 for the semiconductors which are “on”..

$\bar{W}(n-1,n)$ : Represents the calculation matrix corresponding to the minimal topology having (n-1) lines and (n) columns.

$\delta t$ : Is the simulation time used for  $\bar{W}(n-1,n)$  during a simulation period  $2.10^{-4}$  s

For:	D2	TH1	TH2	THP	
T=0.000E+0	1	0	0	0	=> VD=8 , $\bar{W}(4,5)$
T=1.000E-5	1	0	0	1	=> VD=9 , $\bar{W}(4,5)$
T=1.376E-5	0	0	0	1	=> VD=1 , $\bar{W}(6,7)$
T=3.000E-5	0	0	1	1	=> VD=3 , $\bar{W}(4,5)$
T=5.927E-5	0	0	0	1	=> VD=1 , $\bar{W}(6,7)$
T=1.200E-4	0	1	0	1	=> VD=5 , $\bar{W}(7,8)$
T=1.513E-4	0	1	0	0	=> VD=4 , $\bar{W}(6,7)$
T=1.733E-4	1	1	0	0	=> VD=12 , $\bar{W}(4,5)$

We regard a thyristor and an antiparallel diode as only one switch THP.

Table I shows the time ratios obtained when using the example of the chopper figure 6. The gain is between 2 and 3.

Table 1: The time ratios obtained.

$\frac{\bar{W}(\cdot)}{W(\cdot)}$	$\delta_n\%$	$\delta_t \mu_s$	$\frac{\delta t}{\mu = TT} \%$
$\frac{(20)}{(72)}$	27%	$95 \mu_s$	47,5 %
$\frac{(42)}{(72)}$	58%	$75 \mu_s$	37.5%
$\frac{(52)}{(72)}$	77%	$30 \mu_s$	15 %

Where:

$\bar{W}(n-1,n)$ : calculating matrix corresponding to the minimal topology.

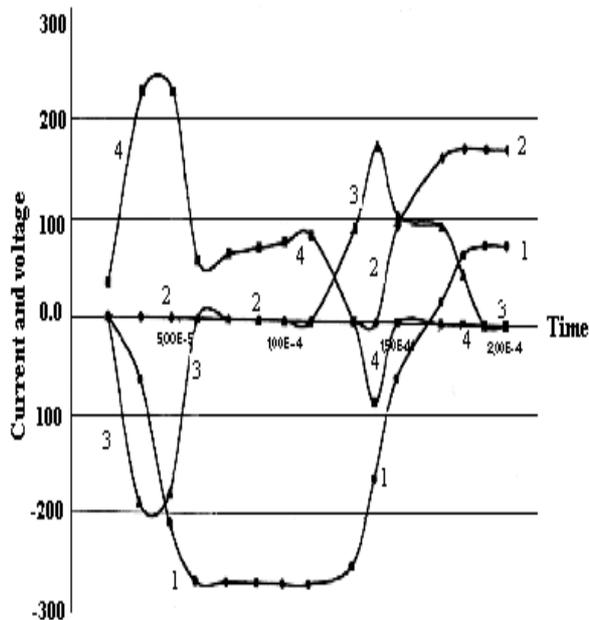
$W(n-1,n)$ : calculating matrix corresponding to the fixed topology.

$\delta_n\%$ : Percentage of the number of elements used for the variable topology.

$\delta_t$ : Simulation time for  $\overline{W}(n-1,n)$  during a certain simulation period.

$\mu$ : Percentage of the simulation time.

TT: Simulation period.



- 1- Voltage across a condensator C3.
- 2- Voltage across a diod D1 and the antiparallel TH.
- 3- Current in a condensator C3.
- 4- Current in a diod D1 and the antiparallel TH.

Figure 7: Represents the results of the simulation using the variable topology.

We present the form of the currents and the tensions for the capacitive branch C3 and the diode D1 and the antiparallel thyristor TH.

## CONCLUSION.

The aim of this article was the study and the realization of a general program of numerical simulation of static inverters by the method of variable topology. The characteristic of this method is the representation of the semiconductors in the form of perfect switches. Work was carried out in two stages. The first stage is the determination of minimal topology according to the state of the semiconductors. Then the reconstitution of the terminal voltages of the blocked semiconductors is removed. The second stage is the setting in automatic equation and the resolution of these equations.

The advantage, which this modeling type brings, is a notable simplification of the system of equation to be solved. The secondary time constants of the modeling

semiconductor by fixed topology are avoided, which implies that an execution time of variable topology is shorter and result of simulation more precise.

## REFERENCES

- Antognetti P., G. Massobrio 1988. "Semi-conductor Device Modeling with SPICE" *McGraw-Hill Book*.
- Boulos F. 1988. "Etude et réalisation d'un programme de simulation de convertisseurs statiques. Utilisation d'une méthode à topologie variable". *Thèse de docteur de l'Université Paris VI*.
- Boulos F. 2001. "Numerical simulation of power electronic Circuit. Use of variable topology method. Modelling of the Semiconductor as perfect switches". *SCS MESM 3<sup>rd</sup> Middle East Symposium on Simulation and Modelling*, (September 3-5), PP. 97-100.
- Boulos F. 2002. "Calculating the current in conducting semiconductors modelled by perfect switches". *SCS MESM, 4<sup>th</sup> Middle East Symposium on Simulation and Modelling*, (September 28-30), PP. 217-222.
- Chue L.O., Lin P.M. 1975 "Computer aided analysis of Electronic circuits" *Prentice-Hall New-Jerssay,USA*.
- Eisenar H. and Hofmenister H 1972. "Digitale nachbildung von elektrischen netz werken mit dioden und thyristoren" *archin fur Electrotechnik* n55.
- Lakatos L. 1979. "A New method for simulating power semiconductor circuits". *IEEE Trans. Ind. Electro. Vont Inst.* Vol IECI - 26 n 1 (Feb).
- Nagel L. N., Pederson D-O.1973. "SPICE" Simulation program with integrated circuit emphasis" *Berkeley California: University of California Electronics Research laboratory Memorandum ERL,M 382* (Apr 12).
- Pelletier P. 1971. "Techniques numériques appliquées au calcul scientifique". *Masson*.
- Rajagopalan 1978. "Simulateur digital des convertisseurs de puissance à thyristors" *Can. Elec. Eng. J.* n 1PP 5-10(Jan).

## BIOGRAPHY OF AUTHOR

**FAOUZI BOULOS** was born in Byblos in Lebanon. He studied in France at the university of Pierre and Marie Curie (Paris VI). He obtained a thesis in electronics in 1988. He worked a few years in France and returned to Lebanon. Since 1995, he is professor at the department of physics of Lebanese University of sciences II. His e-mail address is: faboulos@ul.edu.lb

# PARALLEL SIMULATION MADE EASY WITH OMNeT++

Y. Ahmet Şekercioglu<sup>†</sup> András Varga<sup>‡</sup> Gregory K. Egan<sup>†</sup>

<sup>†</sup>Centre for Telecommunication and Information Engineering, Monash University, Melbourne, Australia

<sup>‡</sup>Omnest Global Inc., Budapest, Hungary

## KEYWORDS

Parallel simulation, discrete-event simulation, PDES

## ABSTRACT

This paper reports a new parallel and distributed simulation architecture for OMNeT++, an open-source discrete event simulation environment. The primary application area of OMNeT++ is the simulation of communication networks. Support for a conservative PDES protocol (the Null Message Algorithm) and the relatively novel Ideal Simulation Protocol has been implemented. Placeholder modules, a novel way of distributing the model over several logical processes (LPs) is presented. The OMNeT++ PDES implementation has a modular and extensible architecture, allowing new synchronization protocols and new communication mechanisms to be added easily, which makes it an attractive platform for PDES research, too. We intend to use this framework to harness the computational capacity of high-performance cluster computers for modeling very large scale telecommunication networks to investigate protocol performance and rare event failure scenarios.

## INTRODUCTION

Telecommunication networks are increasingly becoming more complex as the trend toward the integration of telephony and data networks into integrated services networks gains momentum. It is expected that these integrated services networks will include wireless and mobile environments as well as wired ones. As a consequence of the rapid development, reduced time to market, fusion of communication technologies and rapid growth of the Internet, predicting network performance, and eliminating protocol faults have become an extremely difficult task. Attempts to predict and extrapolate the network performance in small-scale experimental testbeds may yield incomplete or contradictory outcomes. Application of analytical methods is also not feasible due to the complexity of the protocol interactions, analytical intractability and size (Bagrodia et al., 1998). For large scale analysis in both the spatial and temporal domain, accurate and detailed models using parallel simulation techniques offer a practical answer. It should be noted that simulation is now considered as a tool of equal

importance and complementary to the analytical and experimental studies for investigating and understanding the behavior of various complex systems such as climate research, evolution of solar system and modeling nuclear explosions.

This paper reports about the results of implementing parallel simulation support in the OMNeT++ discrete event simulation tool (Varga, 2001). The project has been motivated by and forms part of our ongoing research programs at CTIE, Monash University on the analysis of protocol performance of large-scale mobile IPv6 networks. We have developed a set of OMNeT++ models for accurate simulation of IPv6 protocols (Lai et al., 2002). We are now focusing our efforts to simulate mobile IPv6 networks in very large scale. For this purpose, we intend to use the computational capacity of APAC (<http://www.vpac.org>) and VPAC (<http://www.apac.edu.au>) supercomputing clusters. In a series of future articles, we will be reporting our related research on synchronization methods, efficient topology partitioning for parallel simulation, and topology generation for mobile/wireless/cellular Internet.

## PARALLEL SIMULATION OF COMMUNICATION NETWORKS TODAY

Discrete event simulation of telecommunications systems is generally a computation intensive task. A single run of a wireless network model with thousands of mobile nodes may easily take several days and even weeks to obtain statistically trustworthy results even on today's computers, and many simulation studies require several simulation runs (Bagrodia et al., 1998). Independent replicated simulation runs have been proposed to reduce the time needed for a simulation study, but this approach is often not possible (for example, one simulation run may depend on the results of earlier runs as input) or not practical. Parallel discrete event simulation (PDES) offers an attractive alternative. By distributing the simulation over several processors, it is possible to achieve a speedup compared to sequential (one-processor) simulation. Another motivation for PDES is distributing resource demand among several computers. A simulation model often exceeds the memory limits of a single workstation. Even though distributing the model over several computers and controlling the execution with PDES algorithms may result in slower execution than on a single workstation (due to communication and synchronization overhead in the PDES mechanism), but at least it is possible

to run the model.

It is a recent trend that clusters (as opposed to shared memory multiprocessors) are becoming an attractive PDES platform (Pham, 1999), mainly because of their excellent price/performance ratio. Also, very large-scale network simulations demand computing capacity that can only be provided with cluster computing at affordable costs.

Despite about 15-20 years on research on parallel discrete event simulation (see e.g. (Chandy and Misra, 1979)), PDES is today still more of a promise than part of everyday practice. Fujimoto, a PDES veteran (Fujimoto, 1990), expressed this only last year (Fujimoto, 2002) as: “Parallel simulation provides a benefit, but it has to be transparent, automatic, and virtually free in order to gain widespread acceptance. Today it ain’t. It may never be.”

What parallel simulation tools are available today for the communication networks research community? A parallel simulation extension for the traditionally widely used ns2 simulator has been created at the Georgia Institute of Technology (PADS Research Group), but it is not in wide use. Also, ns2 is apparently losing its popularity because its architecture makes it difficult to simulate wireless networks, a prime interest area today. SSFNet (ssfnet) claims to be a standard for parallel discrete event network simulation. SSFNet’s commercial Java implementation (Renesys Raceway) is becoming popular in the research community, but SSFNet for C++ (DaSSF) does not seem to receive nearly as much attention, probably due to the lack of network protocol models. JavaSim (javasim), another popular network simulation environment does not have PDES support. Parsec (Bagrodia and Meyer, 1998) has to date failed to gain strong foothold in the scientific community outside UCLA, and went into commercial direction instead (Qualnet (qualnet)). The optimistic parallel simulation tool SPEEDES (speedes)(Steinman) has similarly become commercial, and it is apparently not being used for simulation of communication networks. Also, SPEEDES is available for use only within the USA. The best-known commercial network simulation tool, OPNET (opnet) claims to support parallel simulation, but nothing has been published about it. It appears that OPNET simulations can make use of multiprocessor architectures, but cannot run on clusters.

Apparently, the choice is limited for communication networks research groups that intend to make use of parallel simulation techniques on clusters. SSFNet for Java appears to be a feasible choice, but in the C/C++ world there is probably no really attractive choice today. The project effort published in this paper attempts to improve this situation, and there is a good chance that OMNeT++ can fill this niche.

## PARALLEL SIMULATION SUPPORT IN OMNeT++

### About OMNeT++

OMNeT++ (Varga, 2001) is a discrete event simulation environment. The primary application area of OMNeT++ is the simulation of communication networks, but because of its generic and flexible architecture, it has been successfully used in other areas like the simulation of complex IT systems, queueing networks or hardware architectures as well. OMNeT++ is rapidly becoming a popular simulation platform in the scientific community as well as in industrial settings. The distinguishing factors of OMNeT++ is its strongly component-oriented approach which promotes structured and reusable models, and its extensive graphical user interface (GUI) support. Due to its modular architecture, the OMNeT++ simulation kernel (and models) can be easily embedded into your applications. OMNeT++ is open-source and free for academic and non-profit use.

An OMNeT++ model consists of modules that communicate with message passing. The active modules are termed simple modules; they are written in C++, using the simulation class library. Simple modules can be grouped into compound modules. Both simple and compound modules are instances of module types. While describing the model, the user defines module types; instances of these module types serve as components for more complex module types. Finally, the user creates the system module as an instance of a previously defined module type.

Modules communicate with messages which—in addition to usual attributes such as timestamp—may contain arbitrary data. Simple modules typically send messages via gates, but it is also possible to send them directly to their destination modules.

Gates are the input and output interfaces of modules: messages are sent out through output gates and arrive through input gates. An input and an output gate can be linked with a connection. Connections are created within a single level of module hierarchy: within a compound module, corresponding gates of two submodules, or a gate of one submodule and a gate of the compound module can be connected.

Due to the hierarchical structure of the model, messages typically travel through a chain of connections, to start and arrive in simple modules. Compound modules act as ‘cardboard boxes’ in the model, transparently relaying messages between their inside and the outside world. Connections can be assigned properties such as propagation delay, data rate and bit error rate.

### PDES Features

This section introduces the new PDES architecture in OMNeT++ (OMNeT++ has had some support for parallel simulation before, but it was sufficient only for experimental purposes). In its current form it supports conservative synchronization via the classic Chandy-Misra-Bryant (or Null Mes-

sage) Algorithm (Chandy and Misra, 1979). As of September 2003, the implementation is not yet publicly available (beta versions are expected to be published in the first quarter of 2004).

The OMNeT++ design places a big emphasis on *separation of models from experiments*. The main rationale is that usually a large number of simulation experiments need to be done on a single model before a conclusion can be drawn about the real system. Experiments tend to be ad-hoc and change much faster than simulation models, thus it is a natural requirement to be able to carry out experiments without changing the simulation model itself.

Following the above principle, OMNeT++ allows simulation models to be executed in parallel without modification. No special instrumentation of the source code or the topology description is needed, as partitioning and other PDES configuration is entirely described in the configuration files (in contrast, ns2 (PADS Research Group) requires modification of the Tcl source, and SSFNet requires modification of the DML file(s)).

OMNeT++ supports the Null Message Algorithm (NMA) with static topologies, using link delays as lookahead. The laziness of null message sending can be tuned. Also supported is the Ideal Simulation Protocol (ISP) introduced by Bagrodia in 2000 (Bagrodia and Takai, 2000). ISP is a powerful research vehicle to measure the efficiency of PDES algorithms, optimistic or conservative; more precisely, it helps determine the maximum speedup achievable by any PDES algorithm for a particular model and simulation environment. In OMNeT++, ISP can be used for benchmarking the performance of the NMA. Additionally, models can be executed without any synchronization, which can be useful for educational purposes (to demonstrate the need for synchronization) or for simple testing.

For the communication between logical processes (LPs), OMNeT++ primarily uses MPI, the Message Passing Interface standard (MPI). An alternative communication mechanism is based on named pipes, for use on shared memory multiprocessors without the need to install MPI. Additionally, a file system based communication mechanism is also available. It communicates via text files created in a shared directory, and can be useful for educational purposes (to analyze or demonstrate messaging in PDES algorithms) or to debug PDES algorithms. Implementation of a shared memory-based communication mechanism is also planned for the future, to fully exploit the power of multiprocessors without the overhead of and the need to install MPI.

Nearly every model can be run in parallel. The constraints are the following:

- modules may communicate via sending messages only (no direct method call or member access) unless mapped to the same processor
- no global variables
- there are some limitations on direct sending (no sending to a *submodule* of another module, unless mapped to the same processor)

- lookahead must be present in the form of link delays
- currently static topologies are supported (we are working on a research project that aims to eliminate this limitation)

PDES support in OMNeT++ follows a modular and extensible architecture. New communication mechanisms can be added by implementing a compact API (expressed as a C++ class) and registering the implementation – after that, the new communications mechanism can be selected for use within the configuration file.

New PDES synchronization algorithms can be added in a similar way. PDES algorithms are also represented by C++ classes that have to implement a compact API to integrate with the simulation kernel. Setting up the model on various LPs as well as relaying model messages across LPs is already taken care of and not something the implementation of the synchronization algorithm needs to worry about it (although it can intervene if needed, because the necessary hooks are present).

The implementation of the NMA is also modular in itself in that a lookahead discovery mechanism can be plugged in via a defined API. Currently implemented lookahead discovery uses link delays, but it is possible to implement more sophisticated ones and select them through the configuration file.

## Parallel Simulation Example

For demonstrating PDES capabilities of OMNeT++, we will use the closed queuing network (CQN) model described in (Bagrodia and Takai, 2000). The model consists of  $N$  tandem queues where each tandem consists of a switch and  $k$  single-server queues with exponential service times (Figure 1). The last queues are looped back to their switches. Each switch randomly chooses the first queue of one of the tandems as destination, using uniform distribution. The queues and switches are connected with links that have nonzero propagation delays. Our OMNeT++ model for CQN wraps tandems into compound modules.

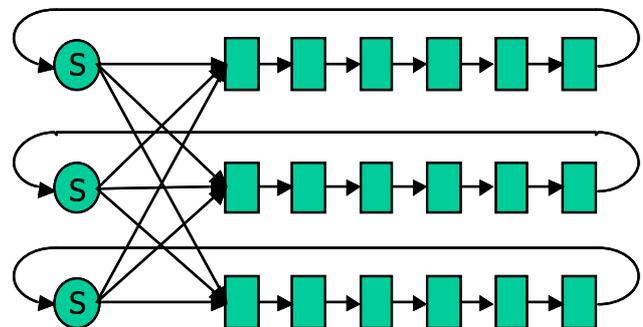


Figure 1: The Closed Queueing Network (CQN) model

To run the model in parallel, we assign tandems to different LPs (Figure 2). Lookahead is provided by delays on the marked links.

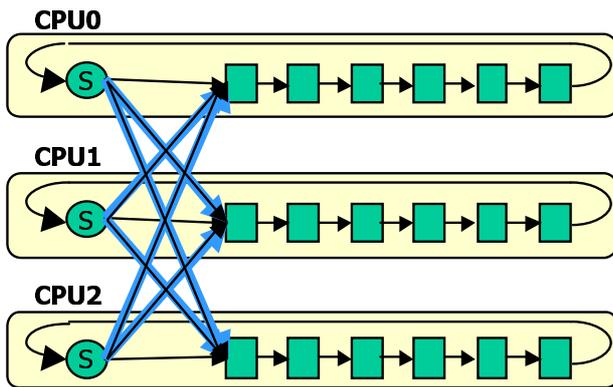


Figure 2: Partitioning the CQN model

To run the CQN model in parallel, we have to configure it for parallel execution. In OMNeT++, the configuration is in a text file called `omnetpp.ini`. For configuration, first we have to specify partitioning, that is, assign modules to processors. This is done with the following lines:

```
[Partitioning]
*.tandemQueue[0].segment-id = 0
*.tandemQueue[1].segment-id = 1
*.tandemQueue[2].segment-id = 2
```

The numbers after the equal sign identify the LP. Also, we have to select the communication library and the parallel simulation algorithm, and enable parallel simulation:

```
[General]
parallel-simulation=true
parsim-communications-class =
    "cMPCCommunications"
parsim-synchronization-class =
    "cNullMessageProtocol"
```

When the parallel simulation is run, LPs are represented by multiple running instances of the same program. When using LAM-MPI (`lam-mpi`), the `mpirun` program (part of LAM-MPI) is used to launch the program on the desired processors. When named pipes or file communications is selected, the `opp_prun` OMNeT++ utility can be used to start the processes. Alternatively, one can launch the processes manually:

```
./cqn -p0,3 &
./cqn -p1,3 &
./cqn -p2,3 &
```

Here, the `-p` flag tells OMNeT++ the index of the given LP and the total number of LPs. For PDES, one will usually want to select the command-line user interface of OMNeT++, and redirect the output to files (OMNeT++ provides the necessary configuration options.)

The GUI of OMNeT++ can also be used (as evidenced by Figure 3), independent of the selected communication

mechanism. The GUI interface can be useful for educational or demonstration purposes as OMNeT++ shows the operation of NMA in a log window, and one also can examine EIT and EOT values.

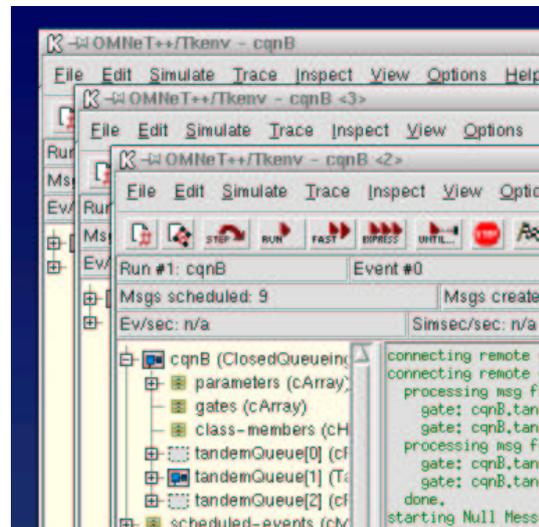


Figure 3: Screenshot of CQN running in three LPs

## Instantiation of Modules

When setting up a model partitioned to several LPs, OMNeT++ uses placeholder modules and proxy gates. In the local LP, placeholders represent sibling submodules that are instantiated on other LPs. With placeholder modules, every module has all of its siblings present in the local LP – either as placeholder or as the “real thing”. Proxy gates take care of forwarding messages to the LP where the module is instantiated (see Figure 4).

The main advantage of using placeholders is that algorithms such as topology discovery embedded in the model can be used with PDES unmodified. Also, modules can use direct message sending to any sibling module, including placeholders. This is so because the destination of direct message sending is an input gate of the destination module, thus if the destination module is a placeholder, the input gate will be a proxy gate which transparently forwards the messages to the LP where the “real” module was instantiated. A limitation is that the destination of direct message sending cannot be a *submodule* of a sibling (which is probably a bad practice anyway, as it violates encapsulation), simply because placeholders are empty and so its submodules are not present in the local LP.

Instantiation of compound modules is slightly more complicated. Since its submodules can be mapped to different LPs, the compound module may not be “fully present” on any given LP, and it may be forced to be present on several LPs (on all LPs where if one or more submodules instantiated). Thus, compound modules are instantiated wherever they have at least one submodule instantiated, and are represented by placeholders everywhere else (Figure 5).

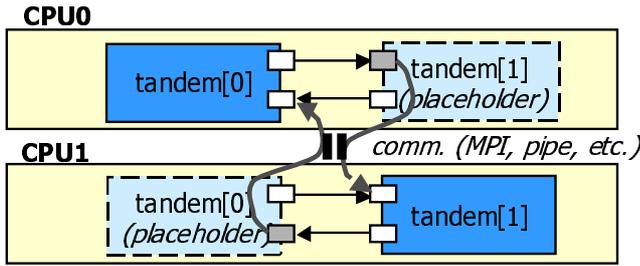


Figure 4: Placeholder modules and proxy gates

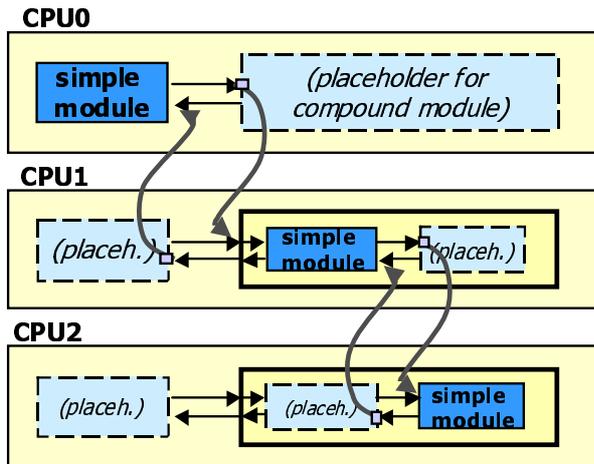


Figure 5: Instantiating compound modules

## Performance Measurements

We have made several runs with the CQN model on 2 and 4 processors, with the following parameters:  $N = 16$  tandem queues,  $k = 10$  and 50 queues per tandem, with lookahead  $L = 1, 5$  and 10. The hardware environment was an Linux cluster (kernel 2.4.9) of dual 1 Ghz Pentium III PCs, interconnected using a 100Mb Ethernet switch. The communication library was LAM-MPI (lam-mpi). The MPI latency was measured to be  $22 \mu\text{s}$ . Sequential simulation of the CQN model achieved  $P_{seq} = 120,000$  events/sec performance.

We executed simulations under NMA and (for comparison) under ISP. The results are summarized in Table 1.  $P_{ISP}$ ,  $P_{NMA}$  are the performances (ev/sec) under the ISP and the NMA protocol, and  $S_{ISP}$ ,  $S_{NMA}$  are the speedups under ISP and NMA, respectively. It can be observed that the  $L$  lookahead strongly affects performance under NMA. An analysis of NMA performance versus lookahead and other performance factors can be found in (Varga et al., 2003). However, it is probably too early to draw conclusions from the figures below about the performance of the OMNeT++ parallel simulation implementation, because we are still optimizing the code.

## DESIGN OF PDES SUPPORT IN OMNeT++

Design of PDES support in OMNeT++ follows a layered approach, with a modular and extensible architecture. The overall architecture is depicted in Figure 6.

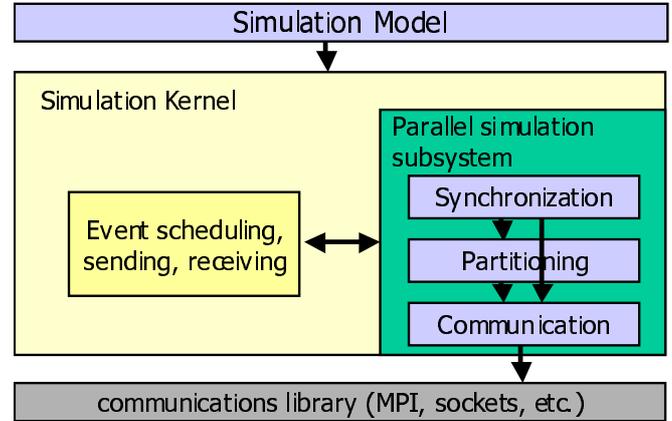


Figure 6: Architecture of OMNeT++ PDES implementation

The parallel simulation subsystem is an optional component itself, which can be removed from the simulation kernel if not needed. It consists of three layers, from the bottom up: communication layer, partitioning layer and synchronization layer.

The purpose of the *Communication layer* is to provide elementary messaging services between partitions for upper layer. The services include send, blocking receive, non-blocking receive and broadcast. The send/receive operations work with *buffers*, which encapsulate packing and unpacking operations for primitive C++ types. The message class and other classes in the simulation library can pack and unpack themselves into such buffers. The Communications layer API is defined in the `cFileCommunications` interface (abstract class); concrete implementations like the MPI one (`cMPICommunications`) subclass from this, and encapsulate MPI send/receive calls. The matching buffer class `cMPICommBuffer` encapsulates MPI pack/unpack operations.

The *Partitioning layer* is responsible for instantiating modules on different LPs according to the partitioning specified in the configuration, for configuring proxy gates. During the simulation, this layer also ensures that cross-partition simulation messages reach their destinations. It intercepts messages that arrive at proxy gates and transmits them to the destination LP using the services of the communication layer. The receiving LP unpacks the message and injects it at the gate pointed to be the proxy gate. The implementation basically encapsulates the `cParsimSegment`, `cPlaceholderModule`, `cProxyGate` classes.

The *Synchronization layer* encapsulates the parallel simulation algorithm. Parallel simulation algorithms are also represented by classes, subclassed from the `cParsimSynchronizer` abstract class. The parallel sim-

LPs	k	L	P <sub>ISP</sub>	P <sub>NMA</sub>	S <sub>ISP</sub>	S <sub>NMA</sub>
2	10	1	147618	76042	1.23	0.63
2	10	5	151250	143289	1.26	1.19
2	10	20	157200	153600	1.31	1.28
2	50	1	168830	131398	1.41	1.09
2	50	5	170289	164563	1.42	1.37
2	50	20	172811	173249	1.44	1.44
4	10	1	300479	45190	2.50	0.38
4	10	5	311392	148007	2.59	1.23
4	10	20	314892	271648	2.62	2.26
4	50	1	359517	144979	3.00	1.21
4	50	5	364663	284978	3.04	2.37
4	50	20	372844	352557	3.11	2.94

Table 1: Comparison of NMA and ISP simulations

ulation algorithm is invoked on the following hooks: event scheduling, processing model messages outgoing from the LP, and messages (model messages or internal messages) arriving from other LPs. The first hook, event scheduling is a function invoked by the simulation kernel to determine the next simulation event; it also has full access to the future event list (FEL) and can add/remove events for its own use. Conservative parallel simulation algorithms will use this hook to block the simulation if the next event is unsafe, e.g. the null message algorithm implementation (`cNullMessageProtocol`) blocks the simulation if an EIT has been reached until a null message arrives (see (Bagrodia and Takai, 2000) for terminology); also it uses this hook to periodically send null messages. The second hook is invoked when a model message is sent to another LP; the NMA uses this hook to piggyback null messages on outgoing model messages. The third hook is invoked when any message arrives from other LPs, and it allows the parallel simulation algorithm to process its own internal messages from other LPs; the NMA processes incoming null messages here.

The null message protocol implementation itself is modular as it employs a separate, configurable lookahead discovery object. Currently only link delay based lookahead discovery has been implemented, but it is possible to implement more sophisticated ones.

The ISP implementation, in fact, consists of two parallel simulation protocol implementations: the first one is based on the NMA and additionally records the external events (events received from other LPs) to a trace file; the second one runs the simulation using the trace file to find out which events are safe and which are not.

Note that although we implemented a conservative protocol, the provided API itself would allow implementing optimistic protocols, too. The parallel simulation algorithm has access to the executing simulation model, so it could perform saving/restoring model state if the code of the simulation model supports this (unfortunately, support for state saving/restoration needs to be individually and manually added to each class in the simulation, including user-

programmed simple modules).

We also expect that because of the modularity, extensibility and clean internal interfaces of the parallel simulation subsystem, the OMNeT++ framework has the potential to become a preferred platform for PDES research.

## CONCLUSION

The paper presented a new parallel simulation architecture for OMNeT++. A merit of the implementation is that it features the “separation of experiments from models” principle, and thus allows simulation models to be executed in parallel without modification. It relies on a novel approach of placeholders to instantiate the model on different LPs. The placeholder approach allows simulation techniques such as topology discovery and direct message sending to work unmodified with PDES. The architecture is modular and extensible so it may serve as a potential framework for research on parallel simulation.

## References

- R. Bagrodia and R. Meyer. PARSEC: A parallel simulation environment for complex systems. *IEEE Computer Magazine*, pages 77–85, oct 1998. URL <http://citeseer.nj.nec.com/bagrodia98parsec.html>.
- R. L. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H. Y. Song. Parsec: A parallel simulation environment for complex systems. *IEEE Computer*, pages 77–85, October 1998.
- R. L. Bagrodia and M. Takai. Performance evaluation of conservative algorithms in parallel simulation languages. *IEEE Transactions on Parallel and Distributed Systems*, 11(4):395–414, 2000. URL [citeseer.nj.nec.com/bagrodia98performance.html](http://citeseer.nj.nec.com/bagrodia98performance.html).
- M. Chandy and J. Misra. Distributed simulation: A case study in design and verification of distributed programs. *IEEE Transactions on Software Engineering SE-5*, (5):

440–452, 1979. URL <http://citeseer.nj.nec.com/context/58222/0>.

R. M. Fujimoto. Parallel discrete event simulation. *Communications of the ACM*, 33(10):30–53, October 1990.

R. M. Fujimoto. Parallel and distributed simulation in the 21th century. In *Grand Challenges for Modeling and Simulation (Seminar 02351), 26-30 August 2002, Dagstuhl Castle, Germany, 2002*. URL <http://www.informatik.uni-rostock.de/~lin/GC/>.

javasim. JavaSim home page. URL <http://www.javasim.org>.

J. Lai, E. Wu, A. Varga, Y. A. Şekercioğlu, and G. K. Egan. A simulation suite for accurate modeling of IPv6 protocols. In *Proceedings of the 2nd International OMNeT++ Workshop*, pages 2–22, Berlin, Germany, January 2002.

lam-mpi. LAM-MPI home page. URL <http://www.lam-mpi.org/>.

MPI. MPI: A message-passing interface standard. *International Journal of Supercomputer Applications*, 8(3/4): 165–414, 1994. Message Passing Interface Forum.

opnet. OPNET Technologies, Inc. home page. URL <http://www.opnet.com/>.

Atlanta PADS Research Group, Georgia Institute of Technology. PDNS - Parallel/Distributed NS home page. URL <http://www.cc.gatech.edu/computing/compass/pdns>.

C. D. Pham. High performance clusters: A promising environment for parallel discrete event simulation. In *Proceedings of the PDPTA'99, June 28-July 1, 1999, Las Vegas, USA, 1999*.

qualnet. QualNet home page. URL <http://www.qualnet.com/>.

speedes. SPEEDES home page. URL <http://www.speedes.com/>.

ssfnet. SSFNet home page. URL <http://www.ssfnet.org>.

J. Steinman. Scalable parallel and distributed military simulations using the SPEEDES framework. ELECSIM '95, 2nd Electronic Simulation Conference, Internet, May-June, 1995.

A. Varga. The OMNeT++ discrete event simulation system. In *Proceedings of the European Simulation Multi-conference (ESM'2001), June 6-9, 2001, Prague, Czech Republic, 2001*.

A. Varga, Y. A. Şekercioğlu, and G. K. Egan. A practical efficiency criterion for the null message algorithm. In *Submitted to European Simulation Symposium (ESS2003), Oct. 2003, Delft, The Netherlands*. Society for Computer Simulation, 2003.

## AUTHOR BIOGRAPHIES

**Y. Ahmet Şekercioğlu** is a researcher at the Centre for Telecommunications and Information Engineering (CTIE) and a Senior Lecturer at Electrical and Computer Systems Engineering Department of Monash University, Melbourne, Australia. He also holds the position of Program Leader for the Applications Program of Australian Telecommunications Cooperative Research Centre (ATCRC, <http://www.atcrc.com>). He completed his PhD degree at Swinburne University of Technology, Melbourne, Australia (2000), MSc (1985) and BSc (1982) degrees at Middle East Technical University, Ankara, Turkey (all in Electrical Engineering). He has lectured at Swinburne University of Technology for 8 years, and has had numerous positions as a research engineer in private industry.

His recent work focuses on development of tools for simulation of large-scale telecommunication networks. He is also interested in application of intelligent control techniques for multiservice networks as complex, distributed systems.

His e-mail address is : [ASekerci@ieee.org](mailto:ASekerci@ieee.org) and his Web-page can be found at <http://titania.ctie.monash.edu.au>.

**András Varga** received his M.Sc. in computer science with honors from the Technical University of Budapest, Hungary in 1994. He worked for several years as software architect for Encorus (formerly Brokat Technologies), which has provided distributed application server technologies for financial institutions in Europe and Asia, and now focusing on Internet and mobile payment solutions.

He is the author of the OMNeT++ open-source network simulation tool currently widely used in academic and industrial settings, and founder of Omnest Global, Inc. which provides commercial licenses and services for OMNeT++ worldwide. He is currently working towards PhD, his research topic being large-scale simulation of communication networks. Between February and September 2003 he visited CTIE at Monash University (Melbourne, Australia) to participate in the parallel simulation research project.

**Gregory K. Egan's** principal research interests are the design, programming and the application of high-performance parallel distributed computer architectures.

He is currently Professor of Telecommunications and Information Engineering, Director of the Centre for Telecommunications and Information Engineering and Head of the Department of Electrical and Computer Systems Engineering at Monash University in Australia.

# PARTITIONING AND FPGA-BASED CO-SIMULATION OF STATECHARTS

Rico Dreier  
Georg Dummer  
Guoxing Zhang  
Klaus D. Müller-Glaser

FZI Forschungszentrum Informatik (Research Center for Information Technologies)  
Department of Electronic Systems and Microsystems (ESM)  
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany  
E-mail: dreier@fzi.de

## KEYWORDS

Partitioning, FPGA, Co-Simulation, Statecharts, VHDL Code Generation.

## ABSTRACT

With the rising complexity and distribution of integrated circuits and embedded systems, the requirements for their development increase as well. Above all, the verification of a large design proves to be the bottleneck of a conventional computer-based simulation in the design flow, requiring several hours to several days. A possible approach to accelerate a simulation is to process different events on several processors at the same time. Assuming a one-processor-architecture, this allows real concurrency of the execution. In a second step the model or a part of the model is executed on reconfigurable hardware. This method is marked by the unique costs of partitioning but takes advantage of the significantly higher execution speed and real parallelism on one chip. This paper presents a framework for a Field Programmable Gate Array (FPGA) based co-simulation of electronic systems as well as an efficient partitioning strategy for Statecharts as a basis for co-simulation. An FPGA based simulation acceleration and the Statechart simulator JStateSim have been developed at FZI/ESM as well as the tool JVHDLGen that enables Very high speed integrated circuit Hardware Description Language (VHDL) code generation out of Statecharts, considering also concurrent charts.

## INTRODUCTION

Dependent on the level of abstraction and the different views, a model can be described in different forms. Finite State Machines (FSM) are suitable to specify the behavior of discrete event systems. Extended FSM additionally introduce the concept of state hierarchy, concurrent states and broadcast. In order to model a complex behavior, Harel extended finite state machines by Statecharts including concepts like combination of Mealy and Moore machines (hybrid state machine), conditions, hierarchical states, history junctions, concurrent states and broadcast (Harel 1987).

The CASE tool MATLAB/Simulink/Stateflow (The Mathworks) is used to work on Statecharts, starting from the point, where the model or a part of the model shall be mapped to hardware and then embedded into

the simulation. Stateflow is a graphical extension for Simulink and allows the modeling and simulation of state machines (as for example BetterState (Wind River) (Stingl and Dreier 1999)). In contrast to Simulink systems, Stateflow is event triggered, which means that a chart is only processed if a previous defined event occurs. Inside a Simulink model a Stateflow diagram is represented by a Stateflow block. A model can have several of them that can communicate through its interface. A Stateflow machine is defined as the set of all Stateflow blocks in a model. During simulation, signals can be exchanged with Simulink. The Stateflow concept extends Harel-Statecharts, e.g., by junctions (allow representing logical structures such as a `for` loop), scopes (to define the valid scope for events and data), directed event broadcasts (allows sending an event explicitly to a given state), implicit events and condition actions (executed, if the condition is evaluated as true) (Dreier et al. 2001).

The simulation acceleration is based on the integration of a hardware supported simulation (emulation) in a heterogeneous network of multiple simulators. Java Remote Method Invocation (RMI) is a mechanism hiding this from the user. It is designed as client-server architecture between local and remote objects. The communication between both is taken over and almost completely hidden by RMI. In order to accomplish a distributed simulation, the model must be partitioned into several concurrently simulated submodels. Each of them is executed by an own simulator. Goal of the partitioning is to maximize parallel execution and to minimize the communication costs at the same time. Furthermore, the static load balancing and the respective synchronization of the simulators are taken into account (Mehl 1994). Since Statecharts have a complex semantic, they cannot be converted directly into graphs, like logic circuits. Possible partitions, the cost of computation per partition and the communication between the partitions must be analyzed a priori. Then, the Statecharts are modeled in hypergraphs, so that partitioning algorithms, that have been proved useful for logic circuits, can also be applied to Statecharts.

## PARTITIONING OF STATECHARTS

### Heuristic Iterative Partitioning Algorithms

The algorithm of Kernighan and Lin (KL) (Kernighan and Lin 1969) is an algorithm of bi-partitioning. KL begins with an arbitrary partitioning and exchanges the nodes from the two partitions in pairs, as long as the cut size can be reduced. The KL algorithm treats only nodes with uniform weights, and the size of a partition is fixed. This algorithm cannot be used for partitioning of hyper graphs. The algorithm of Fiduccia and Mattheyses (FM) (Fiduccia and Mattheyses 1982) is an extension of the KL algorithm. The FM algorithm permits the movement of single nodes between partitions. Thus, the size of a partition is changeable. With the FM algorithm the selection of the node from those with the same gain, which can be moved next, is arbitrary. Krishnamurthy (Krishnamurthy 1984) introduced the concept of "level gains", in order to distinguish between such nodes according to their gains from later movements. Sanchis (Sanchis 1989) extended the concept of Krishnamurthy to a k-way partitioning (K-FM). A node is marked as free, if it is not moved in a pass of the algorithm. Otherwise it is marked as locked. The K-FM algorithm of Sanchis can be used for an arbitrary number of partitions and nodes with different weights.

All iterative methods specified above model a circuit as non-directional graph or hyper graph, without considering the direction of signals. In some situations the information about the direction of signals is useful, in order to improve the partitioning (Cong et al. 1994, Chen et al. 1997).

### Partitioning of Logic Circuits

Before partitioning a logical circuit, its net list is first modeled in graphs. The partitioning of logical circuits can then be realized as partitioning of graphs or hyper graphs. The partitioning of a hyper graph is formalized in (Lengauer 1990). If a given net list contains sequential components (e.g., flip-flops), such components must be removed first. After the acyclic k-way partitioning these components are assigned to suitable partitions on the basis of the same computing load as well as the connections between sequential components and partitions.

A logical block is modeled with nodes and a net between logical blocks with directed hyper edges. The primary inputs and outputs are represented also with nodes. Figure 1 shows a logic circuit with four gates modeled in MATLAB/Simulink.

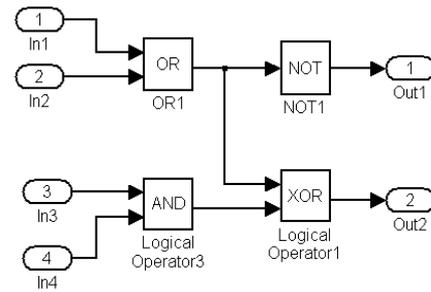


Figure 1: Logic Circuit with Four Gates

The corresponding hyper graph is illustrated in figure 2. For a hyper edge an additional node is added. Here it is marked by a black point. The directed edges between the nodes of a logic gate and the node of a hyper edge represent the input and output nodes of this hyper edge.

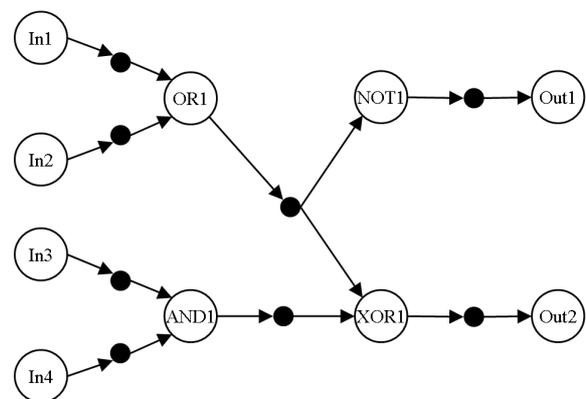


Figure 2: Directed Hyper Graph of the Circuit in Fig. 1

### Analysis of a Stateflow Model

Because Statecharts have a complex semantic, they cannot be converted directly in graphs as it is possible with logic circuits. The possible partitions, the computation cost of a partition and communication between partitions must be analyzed first, which is based on the analysis results of the analyzed Statecharts modeled in hyper graphs. Partitioning algorithms for logic circuits can then be used for the partitioning of Statecharts. The concurrency of a distributed simulation depends on the concurrency of the model. A Stateflow machine can contain several charts. Within Statecharts the concurrency depends on the state hierarchy. Figure 3 shows a Stateflow machine with two chart blocks. The two charts are illustrated in figure 4 and figure 5.

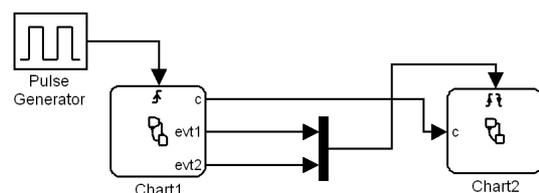


Figure 3: Stateflow Machine with two Statecharts

The state hierarchy can be represented by a tree, whereas the root node is the node for the Stateflow machine. The successors are the nodes for charts. The concurrency of a chart is defined as the number of sub-

states of the chart, if the chart is a state with AND decomposition. If the chart is a state with XOR decomposition, its concurrency is defined as 1, in other words this chart can be simulated on one simulator only.

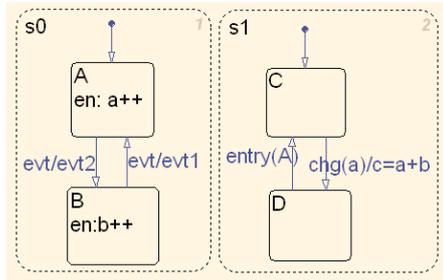


Figure 4: Chart1 of Stateflow Machine in Figure 3

The concurrency of a model is the sum of the concurrencies of all its charts. The maximum number of partitions of a model is decided by its concurrency.

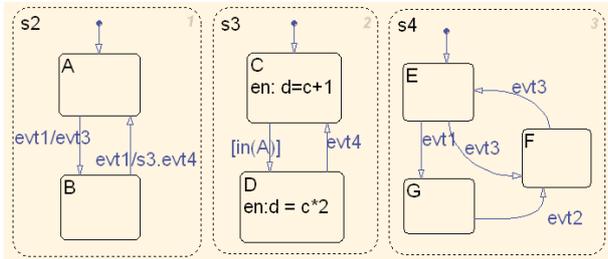


Figure 5: Chart2 of Stateflow Machine in Figure 3

Since only one substate can be active in a state with XOR decomposition at each time, the partitioning of a state with XOR decomposition will produce no parallelism. The possible partition can be either a chart or substates of a chart, depending on whether the chart has XOR or AND decomposition. Figure 6 illustrates the state hierarchy of the model shown in figure 3. The node  $m$  represents the Stateflow machine; the nodes  $c1$  and  $c2$  represent the chart Chart1 and Chart2. The concurrency of the Statecharts Chart1 and Chart2 are thus 2 and 3. Therefore, the concurrency of this model is 5. Thus, this model can be divided in 5 partitions at most.

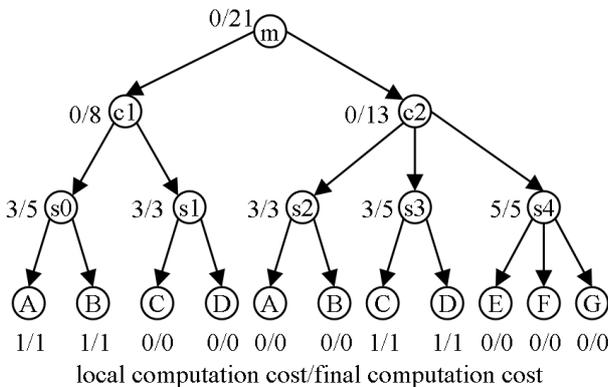


Fig. 6: State Hierarchy of a Model with Concurrency 5

The cost of computation of a model with Statecharts is mainly caused by the execution of state actions and transitions. It is assumed that all transitions and state

actions are executed with the same frequency, the computation cost of each transition and state action is the same, and the computation cost of a state is defined as the sum of the number of its state actions and transitions. Since transitions between states can exist in different hierarchy levels, the computation costs are calculated in two steps. First, local computation costs are calculated. The local computation costs of a state are defined as the sum of the number of state actions and the number of transitions between direct substates. This result will be marked first as weight of a node in the state tree. The final computation costs of a state are calculated by accumulating the final costs of its direct substates and its local costs. This calculation is accomplished gradually from the leaves to the root. For the leaves the local costs are also the final costs. Figure 6 shows, how the computation costs of the model given in figure 3 are calculated. Since the root node does not contain state actions and transitions, its local costs are 0. The substates A and B of the state  $s0$  each contain a state action. The state  $s0$  contains 3 transitions including the default transition. The final computation costs of the state  $s0$  are the sum of the costs of the nodes A, B and its local costs 3.

### Communication Costs Between Partitions

The partitions of states or charts must communicate with each other, if the events generated by a partition need to be passed on to another partition, or if two partitions access common variables. Therefore, the data objects exchanged between two partitions differ from events and variables. A data object can be accessed either by a state action or a transition. In order to determine the communication costs between partitions, a communication graph for a partitioning is defined. A communication channel between two partitions P1 and P2 exists if one of the following conditions is fulfilled:

- A variable will be changed by a state action or transition in P1, simultaneous it is operand of an expression of state action or transition in P2.
- A normal or implicit event in P1 must be sent via broadcast to P2.
- A directed event of a transition in P1 is sent to another transition in P2.
- P1 must inform P2 about the validity of a state within P1 as a state condition.

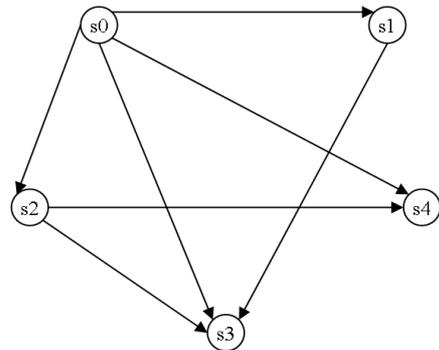


Figure 7: Communication Graph between Partitions

The nodes of the communication graph represent a possible partition. If one of the conditions specified above is fulfilled, a directed edge is added from P1 to P2 as a communication channel. It contains all data objects that shall be sent from P1 to P2. It is assumed that all data objects are exchanged with the same frequency between the partitions and the communication costs between two partitions in one direction are defined as the number of different data objects. Figure 7 shows the communication graph of the model given in the last section.

## FPGA-BASED CO-SIMULATION

### Conversion to VHDL Code

In order to execute a MATLAB/Stateflow model on an FPGA it must be converted to a hardware description language. After the synthesis it can then be executed on an FPGA and merged into a distributed simulation.

SF2VHD (Camera 2001) is a MATLAB M-script that uses the MATLAB API and translates graphically represented Statecharts to a textual representation in VHDL. The VHDL code is output as character string by the individual functions of an M-script. A second M-script serves for the syntax conversion of the action label into appropriate VHDL operations. A powerful language like Statecharts does not map easily into hardware. Therefore, not all possible expressions are considered in SF2VHD. An important constraint regarding Stateflow designs is that no AND decomposition is supported, i.e. states can only have XOR decompositions. In JVHDLGen, that was developed in Java at FZI/ESM, the functionality has been extended by states with AND decomposition. JVHDLGen bases on a parser that is capable to parse MATLAB (mdl) files. The objects are converted to Java objects.

In Statecharts concurrently executed states can communicate with each other by broadcast or common data. In VHDL concurrency is supported by processes. Processes can communicate through signals. In VHDL'93 the data type `shared variable` can be used for communication as well as the instruction `wait` for synchronizing processes. An alternative method is putting concurrent states into one process that is executed in one clock cycle.

An enumeration type is defined for each concurrent state. The possible values of this type are the names of the basic states. A virtual state is additionally added, in order to represent the source state of the default transition. A state inside the virtual state means that this state is inactive, too. Since the history junction is not implemented, yet, all default transitions of the concurrent states are executed when entering the appropriate state. When leaving the state with AND decomposition all concurrent substates are set to the respective virtual state. Thus, with the next entry to the state with AND decomposition, the default transitions are always executed. In order to easily identify state names, the names of all of its super states are added as a prefix. In figure 8

an example Statechart is shown that can be converted to VHDL code by JVHDLGen.

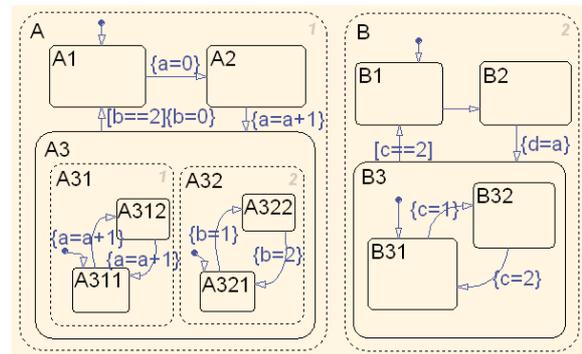


Figure 8: Chart with Nested AND Decomposition

### Development Environment and Tools

The rapid prototyping system RP.2002 (Kühl 2002) is used as an evaluation platform for the development of FPGAs as well as FPGA Advantage of Mentor Graphics. The CORE generator of Xilinx supplies adjustable COREs which are optimized for Xilinx FPGAs. A feature of the RP.2002 is the extensive use of standard hardware (COTS) components. This does not only lead to low costs, the system can be also easily integrated into an existing network and is suitable for distributed simulation. A backplane integrates a PCI controller and an FPGA and allows the access of external actuators and sensors. As operating system Linux is used together with the real-time application interface RTAI.

By means of special blocksets, such as SystemGenerator (Xilinx) and AccelFPGA (AccelChip) as well as the tool JVHDLGen VHDL code can be generated from MATLAB/Simulink/Stateflow models for the configuration of an FPGA. AccelFPGA requires fixed point MATLAB or Simulink files as input. The model either has to be developed in this form or must be transformed to it manually. MATLAB models offer the opportunity to do this automatically by executing an auto quantize command. Additionally, compiler directives must be given to specify which part of the model shall be converted to VHDL. They provide model specific knowledge for the compiler and offer the opportunity for optimization. AccelFPGA generates a VHDL description on register transfer level out of the modified Simulink or MATLAB design, which can then be synthesized and simulated. MATLAB/Stateflow blocks are not supported. Similar limitations exist for SystemGenerator. Therefore, the tool JVHDLGen is used, together with the tools from Mentor Graphics and Xilinx.

JStateSim allows a distributed discrete event simulation of Stateflow charts, that are similar to Harel Statecharts (Harel 1987), analogue to MATLAB/Stateflow, but additionally offers the possibility for a distributed simulation using conservative or optimistic protocols (Schmerler et al. 1997, Fujimoto 2000). It is implemented in Java and is for that reason platform independent. The communication is realized using RMI. Simulator instances can be coupled as many as desired.

Transition notations are restricted to the following formal description:

- if or if/then
- if := [condition] or event
- then := (event)\* and/or (equation)\* (separated by ";")
- equation := VK ◦ VK
- VK := variable or constant
- ◦ := + or -

In order to use JStateSim as part of a distributed simulation with hardware acceleration, a part of the simulator was modified so that the hardware is completely hidden from the other simulators and the control process. In this case the simulator does not get a model as input, since it is loaded while the FPGA is configured. Otherwise the behavior is equivalent to that of an instance of the simulator of JStateSim, but can only be started with according hardware (FPGA) and software (library for accessing the FPGA). The communication between the object instances of JStateSim as well as the CPU on the FPGA rapid prototyping system with the other simulators that are involved in the simulation is taken over by Java RMI.

### Simulation Structure and Design Flow

The simulation structure is shown in figure 9. The communication between the C interface of MATLAB (S function) and Java is done using the Java Native Interface (JNI). A control process communicates with the RMI part of the simulators that are implemented in Java.

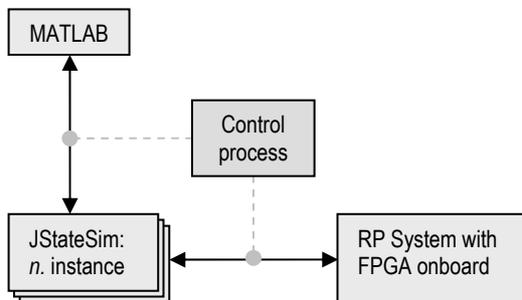


Figure 9: Simulation Structure

A bidirectional connection between MATLAB and the RP system is also possible since the interface is the same as between MATLAB and JStateSim. The JNI is used as an interface between Java and the hardware as well as for the communication between JStateSim and MATLAB. It allows Java code running in a Java Virtual Machine (JVM) to operate with libraries written in C/C++.

The VHDL design flow (figure 10) is divided into three fields: creation, implementation and verification. The starting point of each design is its specification. In VHDL this could be done on abstraction layers like algorithms and function blocks or converted to from a non-formal description. Already on this layer the correctness of the design should be tested by logic simulation. This high level description is translated to the reg-

ister transfer level (RTL) and handed over to the synthesis tool. The supported language by this tool is the first of the dependencies appearing in the tool chain. The synthesis itself converts the VHDL RTL description to a mostly technology independent logic description, e.g., by using special features of a certain FPGA; even the RTL code is not target independent. This step is followed by place and route, which maps the design to the technology of a certain vendor. At this point, precise timing information is available in SDF (standard delay format) and a timing simulation is possible to verify whether all constraints are met. If an FPGA is the target platform, a bitstream for its configuration is generated. The modules are placed on the FPGA and are interconnected.

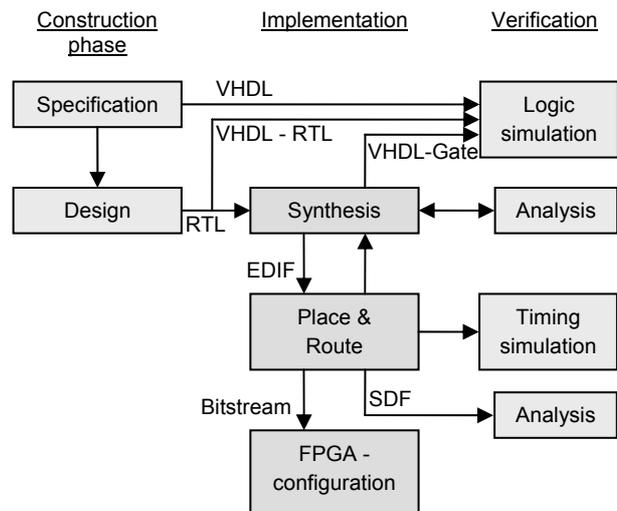


Figure 10: VHDL Design Flow

### Implementation of the Interfaces

A JNI shared library provides an interface to the part of the simulator running on the FPGA. It translates and forwards the messages from JStateSim to the kernel module. Therefore, two FIFOs are installed for communication purposes and cleared at the end of a communication step. A kernel module sets up the communication between the simulator and the FPGA and uses the functionality of the PCI-driver and the FIFOs. The kernel module offers functions for the initialization and cleanup which are executed after loading and before unloading the module. The other functions provide read and write access to the dual port RAM and commands like model reset and execution of a simulation step. These recurrent operations are implemented at this low level to reduce the communication overhead. The bus interface is the link between the local bus, the part of the simulator running, and the FPGA. Basically, it consists of a clock former and a dual-port RAM, instantiated using the Xilinx CORE generator. This type of RAM is used because the local bus and the simulator have to read and write simultaneously to the RAM.

## Control Unit

The control unit (figure 11) makes data available and controls the MATLAB model. Its design should be flexible and serve models without significant changes as much as possible. The simulated model is attached to the control unit. By default, it offers a clock input as well as a synchronous reset and enable input. Data in-

are VHDL statements. The encircled numbers at the beginning of the outgoing transitions wait and continue represent the priorities. In figure 11 only one variable is checked exemplarily. The decision to write the status of the control unit in a status bit can be replaced in the future by an implementation of an interrupt mechanism for the FPGA on the RP.2002.

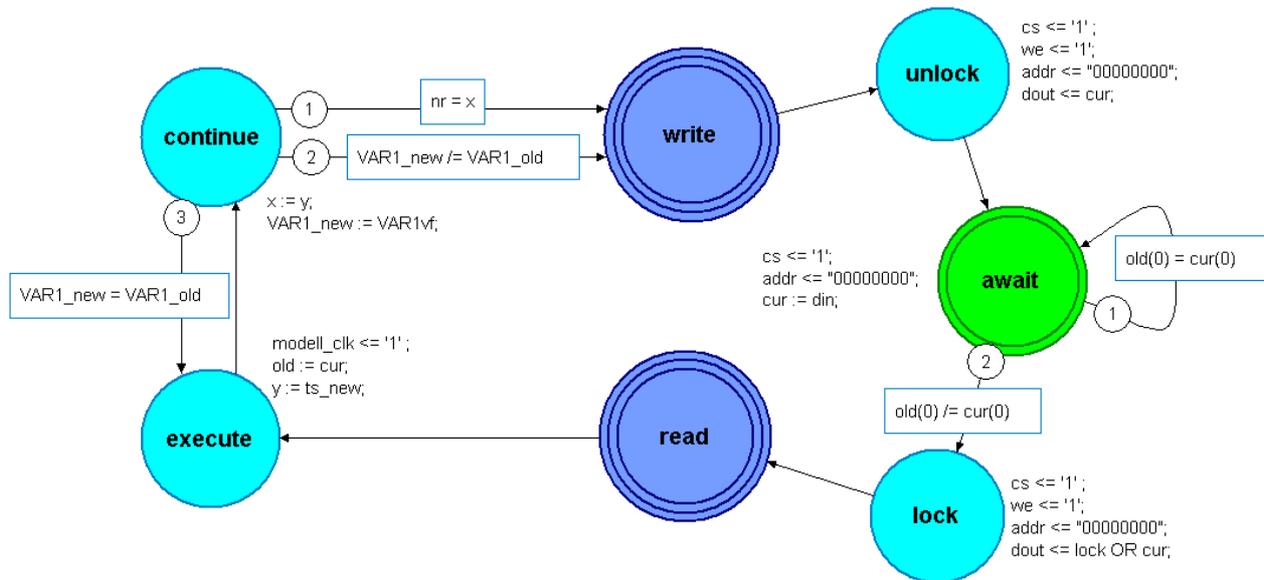


Figure 11: Control Unit

puts and outputs are dependent on the particular model. The control unit is the part of the simulator implemented in hardware. Corresponding to the data stored in the RAM it provides control signals and data to the simulated model. It is composed of different states. The initial state wait reads the control register (address: 0x00) until bit 0 changes. In this case state lock sets bit 3 in the control register to prevent the software from reading invalid values or sending new commands to the control unit. Afterwards the hierarchical state read reads the variables and hands them over to the model. Also, a maximum simulation time is read from a certain register in the RAM. The state exc simulates the model for one time step and increases the time by one. The state continue decides if the simulation can be carried on or not. The break conditions are the simulation time and modified variables. Afterwards the state write writes back the results and the simulation time. After resetting bit 3 in state unlock the control unit returns to the initial state and starts a new cycle. The structure of the control unit arises from the requirements of the distributed simulation and the model. JStateSim requires the control unit to simulate the model to a certain time. If the output variables change, the simulation must be canceled and the new values and time stamp must be handed over to the control process. In order to describe these requirements in VHDL finite state machines are a recommended choice. The state actions and conditions

## CONCLUSION

A platform independent co-simulation framework has been presented that allows for simulation acceleration by emulating a system using FPGAs. A requirement for executing a co-simulation are appropriate partitioned models, so that each participant is as efficient as possible, and the additional communication overhead induced by distributing a model remains as small as possible. Therefore, a proceeding for the partitioning of Statecharts has been suggested that integrates proven partitioning algorithms for logic circuits. Another way to increase the simulation speed is using dedicated simulators like JStateSim with optimistic synchronization protocols as an option. Since the framework has been designed in a flexible way, it can be extended to support also continuous systems. Both will be investigated in the future as well as the supported Statechart features of JVHDLGen extended.

## REFERENCES

- Camera, K. 2001. "SF2VHD: A Stateflow to VHDL Translator". Master Thesis, Department of Electrical Engineering and Computer Science, R. Brodersen, University of California, Berkeley, California, USA.
- Chen, Y.; V. Jha; and R. Bagrodia. 1997. "A Multi-dimensional Study on the Feasibility of Parallel

- Switch-Level Circuit Simulation". *11th Workshop on Parallel and Distributed Simulation PADS '97*.
- Cong, J.; L. Zheng; and R. Bagrodia. 1994. "Acyclic Multi-Way Partitioning of Boolean Networks". *Design Automation Conference 1994*, 670-675.
- Dreier, R.; E. Sax; and K.D. Müller-Glaser. 2001. "Requirements and State of the Art of Automated Software Development for Embedded Systems Based on CASE Tools". *Proc. of IEEE Design, Automation and Test in Europe Conference 2001*, Munich, Germany, 44-48.
- Fiduccia C. and R. Mattheyses. 1982. "A Linear Time Heuristic for Improving Network Partitions". *ACM/IEEE Design Automation Conference 1982*, 175-181.
- Fujimoto, R.M. 2000. "*Parallel and Distributed Simulation Systems*". John Wiley & Sons, Inc.
- Harel, D. 1987. "A Visual Formalism for Complex Systems". *Science of Computer Programming*, No. 8, 1987.
- Kernighan, B. and S. Lin. 1969. "An Efficient Heuristic Procedure for Partitioning Graphs". *The Bell System Technical Journal*, 1969, 291-307.
- Krishnamurthy, B. 1984. "An improved min-cut algorithm for partitioning VLSI networks". *IEEE Transactions on Computers*, 1984, 438-446.
- Kühl, M. 2002. "*Rapid-Prototyping System RP.2002*". developed at the Institute for Information Processing Technology, University of Karlsruhe, Karlsruhe, Germany.
- Lengauer, T. 1990. "*Combinational Algorithms for Integrated Circuit Layout*". John Wiley & Sons, Inc.
- Mehl, H. 1994. "*Methoden verteilter Simulation, Vieweg-Verlag*". Braunschweig, Germany.
- Sanchis, L. 1989. "Multiple-Way Network Partitioning". *IEEE Transaction on Computers*, 1989, 62-81.
- Schmerler, S.; Y. Tanurhan; and K.D. Müller-Glaser. 1997. "Predictive Time Warp". *11th European Simulation Multiconference ESM '97*, Istanbul, Turkey.
- Stingl, T. and R. Dreier. 1999. "Modellierung und Simulation mit Zustandsautomaten - Schwerpunkt BetterState". *ASIM Fachtreffen 1999*, Aachen, Germany.

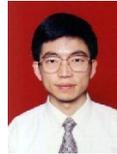
## AUTHOR BIOGRAPHIES



**Rico Dreier** was born in Sinsheim, Germany. He received the Dipl.-Ing. degree in electrical engineering in 1997 from the Technical University of Karlsruhe, Germany. Since 1998 he works at the FZI Research Center for Information Technologies, Dept. of Electronic Systems and Microsystems. He participated in projects in the area of MPEG-4 standardization, CASE methods and analysis of real time operating systems. His research interests include simulator coupling and distributed simulation of electronic systems.



**Georg Dummer** was born in Wittenberg, Germany. Since 1997 he studies computer science at the University of Karlsruhe, Germany. During the study he focused on the design of embedded systems. He is now completing his degree with a diploma thesis.



**Guoxing Zhang** was born in Shanxi, China. He received his BS degree in electrical engineering from the Tongji University in Shanghai in 1991. 1999 he joined the company NEC in Tianjin, China. Since 1999 he studies computer science at the University of Karlsruhe, Germany, and is now working on his diploma thesis.



**Klaus D. Müller-Glaser** received the Dipl.-Ing. and Dr.-Ing. degree in 1972 and 1977 from the University of Karlsruhe, Germany. From 1977 to 1986 he worked for Siemens AG, Synertek Inc., Honeywell Inc. and Bell Labs, Allentown, PA, before he became responsible for setting up the first commercial U.S. AT&T ASIC Design Center in Sunnyvale, CA. In 1986 he was appointed Full Professor at the University of Erlangen-Nürnberg, Germany, in April 1993 he became Full Professor and Director of the Institute for Information Processing Technologies, Department of EE and IT, University of Karlsruhe. He is a Director of the Computer Science Research Center (FZI) in Karlsruhe. From 1996 till 2002 he served as president of FZI, currently he is the dean of the department.

# **SIMULATION IN LOGISTICS, TRAFFIC AND TRANSPORT**



# ANALYSIS OF DYNAMIC PROPERTIES OF AN INVENTORY SYSTEM WITH SERVICE-SENSITIVE DEMAND USING SIMULATION

Yuri Merkurjev and Julija Petuhova  
Department of Modelling and Simulation  
Riga Technical University  
1 Kalku Street  
LV-1658 Riga, Latvia  
E-Mail: merkur@itl.rtu.lv, julija@itl.rtu.lv

Janis Grabis  
Department of Operation Research  
Riga Technical University  
1 Kalku Street  
LV-1658 Riga, Latvia  
E-Mail: grabis@itl.rtu.lv

## KEYWORDS

Inventory management, service-sensitive demand, hybrid modelling.

## ABSTRACT

Complexity of many problems solvable by analytical methods quickly increases under additional assumption. Inventory management under the service-sensitive demand is one of such practical problems. This paper considers application of the hybrid simulation/analytical approach for dealing with this problem. The appropriate closed loop model, that incorporates both simulation and analytical models, has been developed. It has been applied to study behaviour of the inventory system under the service-sensitive demand. The regression analysis conducted indicates that the service-sensitive demand causes substantial deviations of the provided service level from the target service level. The target service level, the demand variability and the lead time are factors substantially influencing the difference between the target service level and the provided service level. The results obtained are to be used for design of a mechanism for adjusting the parameters of the inventory system in order to maintain the target service level.

## INTRODUCTION

The comparative advantages and disadvantages of analytical versus simulation models are well known (for instance, see Nolan and Sovereign 1972). Hybrid simulation/analytical models are used to attain some of the advantages of both types of models, while avoiding the disadvantages. Shanthikumar and Sargent (1983) identify four classes of hybrid simulation/analytical models. Simulation and analytical models are independent parts of a model for the first class of hybrid models. These models are used sequentially. The second class includes hybrid models consisting of simulation and analytical models operating in parallel. The third class comprises hybrid models with dominant analytical models, which use subordinate simulation models for performing special tasks. Finally, in the fourth class a simulation model is a primary model of the system, and it uses inputs from one or more secondary analytical models during the modelling process. Such utilization of analytical models is frequently encountered in complex simulation models. For instance, analytical models are

used to generate demand forecasts (Bhaskaran 1998) and for inventory management (Ganeshan et al. 2001). In order to simplify usage of analytical models in simulation, Baker (1997) develops a methodology for incorporating classic algorithms of operations research into simulation models.

This paper considers a special case of the fourth class hybrid simulation/analytical models. This case describes a situation, where a simulation model is built around an analytical model in order to extend functionality of the analytical model. This type of combination of analytical and simulation models is used to evaluate analytical models under realistic conditions and to consider model and environmental parameters not represented in the traditional formulation of analytical models. Cerda (1997) uses simulation to select the most appropriate ordering options for the complex multi-item re-order point inventory management policy. Clay & Grange (1997) simulate the supply chain of automotive service parts. The simulation model is used to evaluate the impact of different forecasting methods on the supply chain performance. Such analysis provides means for direct evaluation of forecasting methods by evaluating a resulted value of a specified goal function or variable (e.g., a service level) instead of evaluation of forecasting methods according to the forecasting accuracy criterion. Enns (2002) investigates the impact of forecasting accuracy on efficiency of materials requirements planning. In order to overcome limitations of previous research in this area, the author develops a shop floor simulation model for more realistic evaluation of elaborated production schedules. The application of the realistic production schedule evaluation procedure has enabled identification of complex interaction among properties of demand forecasts and characteristics of demand process. Takakuwa & Fujii (1999) develop a standardized simulation model for analysis of transshipment inventory systems. The simulation model is used to provide a more realistic representation of the transshipment problem comparative to the traditional mathematical programming representation. For model building purposes, the authors identify and standardize modules defining the transshipment problem and parameters of these modules.

A modelling problem considered in this paper is inventory management under service-sensitive demand. The service-sensitive demand implies that demand for future periods depends upon the service level observed at the current period. A traditional formulation of analytical inventory models does not consider such dependence.

The existing research on inventory management under service-sensitive demand has been restricted to either situations with deterministic demand or two-period problems (Baker and Urban 1988; Ernst and Powell 1995). These limitations can be explained by an explicitly dynamic character of the problem leading to a complicated analytical analysis. Simulation modelling allows analysing a multi-period problem under stochastic demand. The demand parameters change from one period to another in the case of the multi-period problem. Such behaviour can be observed in highly dynamic and competitive inventory systems. For instance, wholesalers of computer chips often are not able to meet demand due to insufficient supplies from upstream supply chain levels. In the case of the shortage, customers are likely to seek alternative vendors and may choose to place orders to a newly selected vendor for following periods as long as the service level is maintained. The customers may switch back to the initial vendor, if the newly selected vendor reduces its service level. Similar relationships between demand and performance of the inventory system are also observed in retail (Silver and Peterson 1985). The research on service-sensitive demand also relates to research on inventory level dependent demand (see Chung (2003) for a recent account).

Main objectives of this research are to expand modelling of inventory systems with service-sensitive demand by considering multi-period stochastic problems, identify properties of such systems and to test ability of traditional inventory models to meet service level requirements in the case of service-sensitive demand. The service level sensitive demand is modelled similarly to Ernst and Powell (1995). A simulation model is built around analytical models used for inventory management and updating demand parameters according to the observed short term service level. Experimental studies are conducted with the model, and the regression analysis is used to determine impact of service-sensitive demand on performance of the inventory system. The remaining part of the paper discusses issues in inventory management under service-sensitive demand, describes a simulation model for evaluation of such inventory systems and provides preliminary experimental results.

## INVENTORY SYSTEM

A single-item, single-stage, multi-period inventory system is considered. The traditional re-order point policy is used for inventory management. The order size is fixed independently of the re-order point level.

The service level is measured using a proportion of demand satisfied directly from the inventory. Any unsatisfied demand is lost. External demand is normally distributed with mean  $\bar{D}_t$  and the standard deviation  $\sigma_t$ , where  $t = 1, \dots, T, \dots$ . The parameters of the external demand change dynamically according to short-term fluctuations of the service level provided.

The inventory management objective is to preserve the service level at a fixed level. Alternatively, one could try to increase his or her market share. However, the market is assumed to be highly competitive, and other players are expected to take analogous actions preventing one player to achieve a permanent increase of the market share.

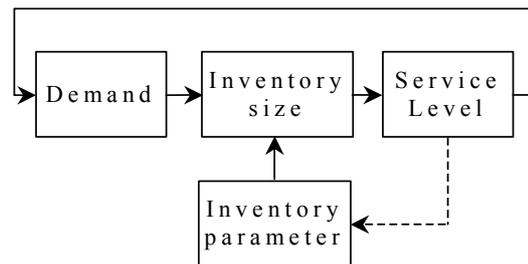


Figure 1: Interactions among Demand, Inventory Size and Service Level

Relationships between the demand, service level and inventory parameters are shown in Figure 1. The external demand causes depletion of the inventory level. The inventory is replenished according to the re-order point policy specified by a set of the inventory parameters (re-order point, order size, mean demand, demand standard deviation, lead time and target service level). The service level achieved during a relatively short time period is observed. This service level is most likely to differ from the target, required service level. In the case of the service-sensitive demand, this causes changes of the demand parameters. A higher than target service level causes increase of the mean demand and the standard deviation. A lower than target service level causes decrease of the mean demand and the standard deviation. The increase of the demand parameters may result in a declining service level in forthcoming periods unless the inventory parameters are properly adjusted. The decrease of the demand parameters may cause overstocking unless the inventory parameters are properly adjusted. Therefore, a link representing updating of the inventory parameters according to the observed short-term service level should be established.

## SIMULATION MODEL

The inventory system described above has an explicitly dynamic character. Simulation is used to capture this behaviour of the system. A simulation model developed describes the inventory system and incorporates an analytical model for implementing a feedback between the simulated short-term fluctuations in the service level

and the customer demand parameters. A structure of the considered hybrid simulation/analytical model is given in Figure 2.

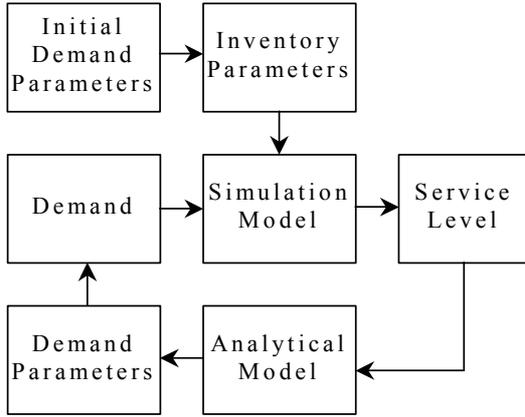


Figure 2: Structure of the Hybrid Simulation/Analytical Model

Simulation is used for analysis of properties of the system. Results of the analysis are expected to create a basis for developing a mechanism for updating of the inventory parameters in order to maintain the service at the target level.

Main steps of the simulation analysis being performed are as follows:

1. Initialise the input data module, including the initial (received by traditional forecasting techniques) forecast of the customer demand;
2. Perform simulation of inventory control processes;
3. Calculate the observed service level for a current time period;
4. Calculate the customer demand distribution parameters based on the observed service level;
5. Update the demand parameters in the simulation model;
6. Go to step 2 until simulation is completed.

Inventory management is based on the re-order point policy, where the order size  $Q$  is fixed independently of the re-order point level. The re-order point level is calculated using a formula:

$$ROP = LT * \bar{D}_0 + z * \sqrt{LT} * \sigma_0, \quad (1)$$

where  $LT$  is the lead time,  $\bar{D}_0$  is an initial mean demand during one period,  $\sigma_0$  is an initial standard deviation of the demand during one period,  $z$  is a safety factor that depends on a specified target service level.

The short term service level is calculated each period using a formula:

$$SL_t = 1 - \frac{SO_t}{D_t}, \quad (2)$$

where  $t$  is a current time period,  $SO_t$  is an unsatisfied demand in period  $t$ ,  $D_t$  is an observed actual demand in period  $t$ .

An impact of the customer service level on a future customer demand is quantified similar to Ernst and Powell (1995). In this approach, a linear relationship between the service level and the demand parameters is assumed. This dependence is evaluated based on parameters estimated by experts. This means that the mean demand increases/decreases by  $\alpha$  points if the change in the service level doesn't exceed a certain threshold:

$$\bar{D}_t = (1 + \alpha * (SL_t - SL_{t-1})) * \bar{D}_{t-1}, \quad (3)$$

where  $SL_{t-1}$  is the short term service level in the previous time period,  $\bar{D}_{t-1}$  is the mean demand of the previous time period,  $\alpha$  is a coefficient of the change in mean demand with increased/decreased service level.

The standard deviation of the demand for the new demand level is expressed as a function of the parameters  $\alpha$ ,  $\beta$  and the standard deviation  $\sigma_{t-1}$  from the previous time period:

$$\sigma_t = \left[ 1 + \beta^2 \alpha (SL_t - SL_{t-1}) \right]^{\frac{1}{2}} \sigma_{t-1}, \quad (4)$$

where  $\beta$  is a coefficient of the change in standard deviation of demand with changed service level.

In case if the increase/decrease in the service level exceed a restricted constant the mean demand is calculated by a formula:

$$\bar{D}_t = (1 + \alpha * (SL_t - SL_{t-1}) * MaxChange) * \bar{D}_{t-1}, \quad (5)$$

where  $MaxChange$  is a constant of the maximal change in the service level.

The standard deviation of the demand in this case is found by a formula:

$$\sigma_t = \left[ 1 + \beta^2 \alpha (SL_t - SL_{t-1}) * MaxChange \right]^{\frac{1}{2}} \sigma_{t-1} \quad (6)$$

If the short term service level is equal to one for two consecutive periods and the demand parameters do not exceed their initial values, the demand parameters are updated using the following expressions:

$$\bar{D}_t = \left( 1 + \frac{\alpha}{10} \right) * \bar{D}_{t-1}, \quad (7)$$

$$\sigma_t = \left[ 1 + \frac{\beta^2 * \alpha}{10} \right]^{\frac{1}{2}} \sigma_{t-1}. \quad (8)$$

If this restriction is not imposed, the system may settle for providing a high service level on expense of carrying excessive inventory.

The simulation model is developed using the ARENA simulation modelling environment. Evaluation of the service level and updating of the demand parameters are implemented using Visual Basic.

## EXPERIMENTAL EVALUATION

### Experimental Design

Objective of experimental studies is to determine the short term customer service level, to identify parameters of the inventory system influencing disagreement between the target and observed service levels and to evaluate changes of the customer demand parameters. Therefore, a set of experiments with a feedback from the simulation model to the analytical model, when the demand parameters are updated taking into consideration the observed service level (service-sensitive demand), is performed. Performance of the inventory system is evaluated under various factors such as initial end customer mean demand, signal to noise ratio, target service level, lead time, and order size coefficient (Table 1).

Table 1: Experimental Design

Factors	$\bar{D}_0$	Signal to Noise	Target Service Level	LT	Q
Values					
Min	50	2	0.9	2	1xLT
Max	250	10	0.99	6	2xLT

The *Signal to Noise* factor describes variability of the demand process. A value of this factor normally should be in range between 1 and  $\bar{D}_0$ . Given the value of  $\bar{D}_0$ , the initial standard deviation ( $\sigma_0$ ) is found by a formula:

$$\sigma_0 = \frac{\bar{D}_0}{\text{Signal to Noise}}, \quad (9)$$

where  $\bar{D}_0$  is the initial mean demand.

The *Target Service Level* factor describes the required service level of the inventory system considered. The *LT* factor value corresponds to time between the order placement time and the order arrival. It is measured in days. Two values of the fixed order size are considered. The minimum value is equal to the initial mean lead time demand, the maximum value is equal to the double initial mean lead time demand. The short term service level is observed every 5 days. The demand parameters are re-evaluated at the same time interval. The difference between the target ( $SL_{Target}$ ) and the provided ( $SL_{Provided}$ ) service level is the main performance measure. The provided service level represents the overall system service level and is calculated at the end of each run by a formula:

$$SL_{Provided} = 1 - \frac{\sum_{t=1}^T SO_t}{\sum_{t=1}^T D_t}, \quad (10)$$

where  $T$  is a replication length.

Experiments are conducted according to a factorial experimental design with resolution IV. This design consists of 16 experimental cells. The model was run for 5 replications. Each replication length is defined as 250 weeks and a warm-up period is 20 weeks. Thus, simulation results are independent from the empty-and-idle initial state; there is no predetermined starting and finishing point for a simulation run.

### Experimental Results

The simulation results are summarized using the regression analysis. We analyse how a difference between target and provided service levels is affected by values of the experimental factors considered. The dependent variable  $Y$  is defined as  $(SL_{Target} - SL_{Provided})$ . Estimated coefficients of the regression equation and associated  $p$ -values are reported in Table 2.

Table 2: Results of the Regression Analysis

Independent variables	Coefficients	$p$ -value
Constant	-0.795	0.00
$\bar{D}_0$	0.00002	0.05
Signal to Noise	-0.004	0.00
LT	0.006	0.00
Order Size Coefficient	0.003	0.15
Target Service Level	0.848	0.00

Ideally, the difference between  $SL_{Target}$  and  $SL_{Provided}$  should be equal to zero. However, the service-sensitive demand causes deviation of the provided service level from the target service level. Values larger than zero indicate that the required, target service level is not reached. Values smaller than zero mean that reached service level is higher than expected. The regression equation suggests that the *Signal to Noise* factor, *LT* and *Target Service Level* have the most significant impact on the difference between the target and provided service levels. The order size does not have the significant impact on this difference. The provided service level is likely to be smaller than the target service level, if the target service level is high, the lead time is long and the demand is highly variable. With a high degree of confidence (95%) the observed service level averaged over all experimental cells differs from the required service level by 5%, and the mean customer demand differs from the initial mean demand on average by 13%.

Dynamic behaviour of the mean demand and the observed short term service level during simulation is shown in Figures 3, 4, and 5. The results are obtained for different values of the target service level. Values of the initial mean demand, the signal to noise ratio, the lead time and the order size coefficient are fixed at 250, 2, 2, 1, respectively.



Figure 3: Mean Demand with Target Service Level 0.9



Figure 4: Mean Demand with Target Service Level 0.95

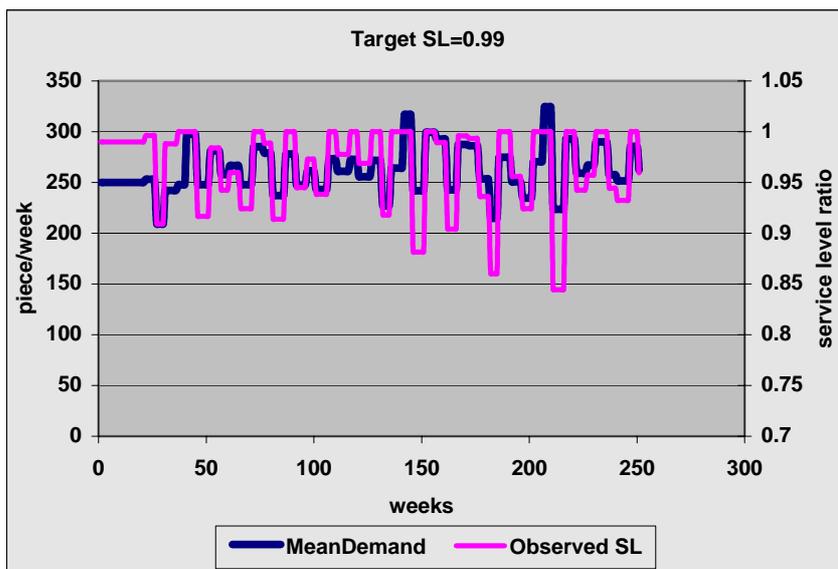


Figure 5: Mean Demand with Target Service Level 0.99

Experimental results show that the observed service level achieved each week often differs from the target service level. This leads to changes in the end customer demand. The demand volume changes are the main cause of the decreasing/increasing of the service level, because of reaching a high service level the demand volume become large and inventory control system is not able to adapt to a new environment during the short time. This leads to a lower service level in the next period and the demand volume becomes smaller. That increases the service level in the next period. This sequence is kept during a simulation run.

The target service level has an impact on the demand parameters. The demand parameters increase, if the target service level is high. The demand parameters decrease, if the target service is low. The increase of the demand parameters is observed because, in the case of the high target service level, the observed short term service level is often equal to one, shortages occur less frequently causing fewer possibilities for the demand parameters to decline.

## CONCLUSION

The analysis of re-order point inventory systems under the service-sensitive demand has been extended to multi-period, stochastic demand situation. The hybrid simulation/analytical modelling approach has been advocated as an appropriate technique for conducting this analysis. The appropriate simulation model, which incorporates analytical models for inventory management and modelling of the service-sensitive demand, has been developed. It has been applied to study behaviour of the inventory system under the service-sensitive demand. Analytical models provide a well-defined mechanism for inventory management. However these models not give an impression of the system operation over the time. Therefore, a simulation technique is used to perform the analysis of the system dynamic behaviour.

The regression analysis conducted indicates that the service-sensitive demand causes substantial deviations of the observed service level from the required, target one. Additionally, the observed service level is lower than the target level, if latter is larger, while the observed service level is higher than the target one, if latter is smaller. The demand variability and the lead time are other factors substantially influencing the difference between the target and the provided service levels. The results obtained are to be used for design of a mechanism for adjusting the parameters of the inventory system in order to maintain the target service level. The simplest mechanism for achieving this objective is recalculating of the inventory parameters according to new values of the demand parameters. However, preliminary studies of this mechanism suggest that a more complex preemptive approach is needed.

## REFERENCES

- Baker, G.S. 1997. "Taking The Work Out Of Simulation Modeling: An Application Of Technology Integration". In *Proceedings of the 1997 Winter Simulation Conference*, 1345-1351.
- Baker, R.C.; and Urban, T.L. 1988. "A deterministic inventory system with an inventory-level-dependent demand rate". *Journal of the Operational Research Society* 39, 9, 823-831.
- Bhaskaran, S. 1998. "Simulation analysis of a manufacturing supply chain". *Decision Sciences* 29, 3, 633-657.
- Cerda C.B.R.; and de los Monteros, A.J.E. 1997. "Evaluation of a (R,S,Q,C) Multi-Item Inventory Replenishment Policy Through Simulation". In *Proceedings of the 1999 Winter Simulation Conference*, 825-831.
- Chung, K.-J. (2003), An algorithm for an inventory model with inventory-level-dependent demand rate, *Computers & Operations Research* 30, 9, 1311-1317.
- Clay, G.R.; and Grange, F. 1997. "Evaluating Forecasting Algorithms And Stocking Level Strategies Using Discrete-Event Simulation", In *Proceedings of the 1997 Winter Simulation Conference*, 817-824.
- Enns, S. T. 2002. "MRP performance effects due to forecasting bias and demand uncertainty". *European Journal of Operational Research* 138, 87-102.
- Ernst, R. and S. G. Powell (1995), Optimal inventory policies under service-sensitive demand, *European Journal of Operational Research* 87, 316-327.
- Ernst, R. and S. G. Powell (1998), Manufacturer incentives to improve retail service level, *European Journal of Operational Research* 104, 437-450.
- Ganeshan, R., T. Boone and A. J. Stenger (2001), The impact of inventory and flow planning parameters on supply chain performance: An exploratory study, *International Journal of Production Economics* 71, 1-3, 111-118.
- Ho, P.-K. and J. Perl (1995), Warehouse location under service-sensitive demand, *Journal of Business Logistics* 16, 1, 133-162.
- Nolan, R. L. and M. G. Sovereign (1972), A recursive optimization and simulation approach to analysis with an application to transportation systems, *Management Science* 18, 12, 676-690.
- Shanthikumar, J. G. and R. G. Sargent (1983), A Unifying View of Hybrid Simulation/Analytic Models and Modeling, *Operations Research* 31, 6, 1030-1052.
- Silver, E.A.; and Peterson R. 1985. Decision systems for inventory management and production planning, 2nd ed. New York: Wiley.
- Takakuwa, S.; and Fujii, T. 1999. "A Practical Module-Based Simulation Model for Transshipment-Inventory Systems". In *Proceedings of the 1999 Winter Simulation Conference*, 1324-1332.

# SIMULATION OF CITY BUS OPERATION PROCESSES AS TRANSPORTATION SYSTEM REALIZATION

Wojciech Osmólski  
Institut of Machines and Vehicles  
Poznan University of Technology  
ul. Piotrowo 3, 60-965 Poznań  
e-mail: [wosm@sol.put.poznan.pl](mailto:wosm@sol.put.poznan.pl)

Waldemar Osmólski  
Ponetex Logistics Sp. z o.o.  
ul. Grodziska 50, 62-067 Rakoniewice  
e-mail: [w.osmolski@ponetex.com.pl](mailto:w.osmolski@ponetex.com.pl)

Paweł Kaczalski  
Solaris Bus&Coach Sp.z o.o.  
ul.Obornicka 1, Bolechowo, 62-005 Owińska  
e-mail: [kaczalski\\_p@solarisbus.pl](mailto:kaczalski_p@solarisbus.pl)

## KEYWORDS

Transportation system, city logistics, urban traffic, simulation.

## ABSTRACT

In the paper methods of modelling nonlinear transportation systems have been presented basing on urban buses in aspect of considering processes connected with city-logistics. In chapter 1 deterministic nonlinear model of functioning of urban traffic transportation system has been formulated with distinction of elements dependent on vehicle operating systems. In the next chapter has been presented test method as well as investigations carried out together with the results obtained for the NEOPLAN N4020 bus. The investigations were carried out in Warsaw.

## 1. NON-LINEAR MODEL OF URBAN TRAFFIC TRANSPORTATION SYSTEM FUNCTIONING.

In the assumed model of the transportation system we assumed one type of decision variables, two types of restricting conditions and six types of criteria. Decision variable in the model is traffic frequency in a given traffic line  $h$  ( $\omega_h$ ). The total number of decision variables occurring in the assumed model depends on number of traffic lines in the analyzed transportation system and equals  $H$ . All types of restricting conditions, which occur in the model, have of course a determining character and each of them is analysed for the particular traffic lines conditions. The total number of restricting conditions equals thus  $2H$ . The first Group of restricting conditions defines permissible values of traffic frequency in the particular communication lines according to relation  $60/i_6 \leq \omega_h \leq 60/i_0$

The following types of criteria were taken into consideration:

- 1) traffic interval  $IR_h$
- 2) waiting time  $CO_h$

- 3) bus change factor  $WP$
- 4) overcrowding degree  $SP_h$
- 5) running costs  $KE$
- 6) effectiveness  $EF_h$

In addition we calculated in the model III:

- 1) operating speed  $PE_h$
- 2) number of vehicles  $x_h$

From all of six types of criteria two of them  $WP$  and  $KE$  have a global character, that means that they are defined as general indicators for the whole urban traffic system.. The other four  $IR_h$ ,  $CO_h$ ,  $SP_h$  and  $EF_h$  have been analyzed in the particular traffic lines. Therefore the total number of criteria amounts to  $4H+2$ . Criteria 1) and 6) are maximized and criteria 2) ÷ 5) are minimized. The criterion traffic interval ( $IR_h$ ) is defined (in minutes) in the form of:

$$IR_h = 60p_m S_h / (P_h S_{sr}) \quad (1)$$

where:

$p_m$  - vehicles capacity (in person),

$S_h$  - length of communication line,

$P_{hr}$  - the number of passengers by the line,

$S_{sr}$  - average length of person trip (km).

This criterion is maximized and number equals is  $H$ .

The criterion of waiting time ( $CO_h$ ) is often formulated partially in a nonlinear function with use of an approximation with a multinomial of the sixth degree. The successive criterion, which is taken into consideration in nonlinear models, is the factor of changes ( $WP$ ). This is one of the two global criteria. As a minimized criterion it belongs to a group of criteria, which are essential for a passenger, it is determined on base of the equation

$$WP^{(h)} = d_2 CO_h^2 + d_1 CO_h + d_0 \quad (2)$$

where  $d_0$ ,  $d_1$  and  $d_2$  are constants determined for the specified traffic conditions.

The relation 2 will, after simple transformations, assume a form:

$$WP = \frac{1}{H} \sum_{h=1}^H [d_2 (b_6 w_h^6 + b_5 w_h^5 + \dots + b_1 w_h + b_0) + d_1 (b_6 w_h^6 + b_5 w_h^5 + \dots + b_1 w_h + b_0) + d_0] \quad (3)$$

The successive criterion considered in the non-linear model is the overcrowding degree ( $SP_h$ ) in a given traffic line in form of:

$$SP_h = P_{hmax} / (p_m \omega_h) \quad (4)$$

where:  $P_{hmax}$  – max. passengers flow,

$p_m$  – vehicles capacity (in person),

$\omega_h$  – traffic frequency.

This criterion is minimized and hence it is of great importance for a passenger. It is analyzed for each traffic line  $h$ .

Last but one from criteria running costs is a global criterion belonging to a group, which is essential one from the point of view of a transportation firm. This is a criterion, which as a minimized one tends to reduce general outlays spent. to secure functioning of urban communication systems. This system is determined as:

$$KE = \sum_{m=1}^L z_m n_{wkmm} \quad (5)$$

where:  $KE$  – Total running cost of functioning of the communication system.

Each of the magnitudes occurring in the formula (5), i.e.  $z_m$  and  $n_{wkmm}$  has been modelled in a different way. With reference to the cost of a bus-workday we assumed to calculate it according to an equation

$$z_a = g_{1a} PE^{g_{2a}} + g_{3a} \quad (6)$$

The operation speed  $PE_h$ , occurring in this relation can be calculated from the equation

$$PE_h = 60 S_h / (t_h + \Delta_h) \quad (7)$$

where:

$PE_h$  – operational speed,

$S_h$  – length of communication line,

$t_h, \Delta_h$  – time of trip and parking.

The operation speed of a transportation unit of  $m$  type operating in  $h$  communication line is not a criterion of this model as it is independent from the decision variable. This magnitude calculated according to the equation (7) is provided to the decision maker as an additional effectiveness indicator which is essential

from the point of view of the user as well as the operator of a communication system. After some transformations the cost of one bus-kilometre can be determined from the equation

$$z_a = g_{1a} (60 S_h / (t_h + \Delta_h))^{g_{2a}} + g_{3a}$$

$$\text{for } v_{1a} \leq (60 S_h / (t_h + \Delta_h)) \leq v_{3a} \quad (8)$$

and the number of bus-kilometres can be presented in form of:

$$n_{wkmm} = \sum_{k=k_{m-1}+1}^{k_{m-1}+k_m} (60 \alpha x_h S_h / (t_h + \Delta_h)) \quad (9)$$

$m=1, 2, \dots, L$

After further (simple) transformations we will obtain

$$n_{wkmm} = \sum_{k=k_{m-1}+1}^{k_{m-1}+k_m} (60 \alpha S_h \omega_h) \quad (10)$$

$m=1, 2, \dots, L$

$\omega_h$  – traffic frequency

Hence the final form of the criterion of operation costs will assume a form of:

$$KE = \sum_{h=1}^H 2 \alpha S_h (f_1 (60 S_h / (t_h + \Delta_h))^{f_2} + f_3) \omega_h \quad (11)$$

where  $f_1, f_2, f_3$  non-linear coefficients.

In effect a linear dependence of operation costs on traffic frequency has been obtained.

There is, however, some nonlinearity in this equation connected with the cost of bus-kilometre, though they are directly invisible. The last criterion – effectiveness ( $EF_h$ ) – is a quotient of number of passenger-kilometres and number of bus-kilometres in a given communication line.

$$EF_h = D_h / n_{wkmm} \quad (12)$$

Schedule speed and number of buses  $x_h$  assigned to a communication line  $h$  are provided to a decision maker as indicators aiding the analysis and the decision makes process.

## 2. SIMULATION OF RUNNING PROCESSES – DEFINING THE RUNNING (SCHEDULE) SPEED.

For simulation of running processes a computer system was created, the BASIC part of which is the data base in which are gathered the so called model signals for running vehicle processes under specific urban conditions.



Fig.1 Investigated bus NEOPLAN N4020

The investigations were carried out on a low-floor bus NEOPLAN N4020 (Fig. 1) which the specification is presented in table below.

DIMENSIONS	
Length	14 600 mm
Width	2 500 mm
Height	2 950 mm
Wheel space	7 000 mm
WEIGHTS	
Complete vehicle weight	13 800 kg
Permissible weight	25 000 kg
ENGINE	
Type	DAF GS 200M, lying
Power	200 kW
Max revolutions	2 300 revs/min
Max. torque	1010 Nm at 1500 revs/min

During the investigations the total weight of the bus amounted:

- 12 tonnes of cargo plus bus weight equal 13,8 t plus driver, three persons to make measurements and weight of the measuring equipment.
- 6 tonnes of cargo plus bus weight equal 13,8 t plus driver, three persons to make measurements and the weight of the measuring equipment.

The following sensors were used to make measurements:

Extensometric torque meter with a digital telemetric signal transmission, sensors to measure rotational speed of wheels provided with optoelectronic rotational converters MOL2500, sensor of longitudinal and transverse speed V-1 of DATRON mounted at the rear of the bus, which is presented in the drawing.

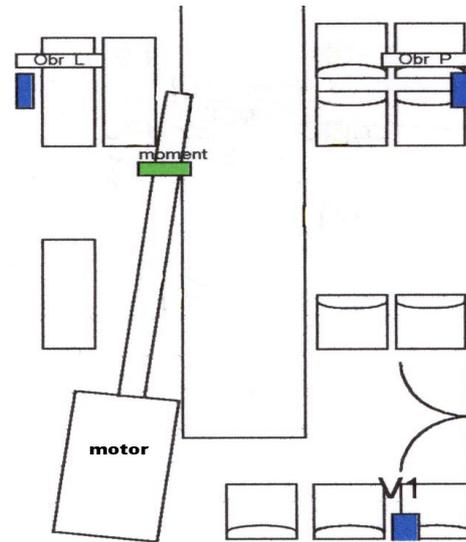


Fig. 2 Arrangement of sensors at the rear of the bus

Parameters of sensors used are provided in the table below.

OPTIC SENSORS V1	
Range of measurement speed-V1	0,25...310 km/h
Measurement terror	$\pm 0,5$ % in measurement INTERNAL
Outlet frequency	0 - 40 kHz
Digital outlets (RS 485)	3 - Vx, Vy, direction Vy
SENSORS OF ROTATIONAL SPEED OF WHEELS	
Type	MOL2500
Measurement range	4, 8, 16 i 32 revs/s
Resolution (number of graduations)	2500
Output signal	Analogue $\pm 5$ V

All sensors co-operated with a digital data acquisition unit of German DATRON – AEP 2.50 of resolution of 12 bits with a portable computer.

The investigations were carried out on 18-21 June 2001 in Warsaw on routes proposed by Urban Bus Company.

In the drawing 38 a map of investigation routes is presented. General investigations were carried out in two stages:

for traffic jams during morning rush hours for 50 % bus load,

for traffic jams during afternoon rush hours for 100 % bus load.



Fig. 3 Map of a ride along investigated routes

During the measurement the most characteristic bus operation periods in DRIVE in urban traffic were measured:

Rides in a street „jam” – starting and stopping a bus in result of traffic of a vehicle stream,

Rides along routes of urban bus lines – starting from bus bays with joining the intense urban traffic and pulling over into a bay with pulling up at bus stops,

Rides along routes in the streets of deteriorated surface conditions – uneven asphalt surfaces and surface of concrete slabs/tiles,

Average rides beyond rush hours through streets of city centre.

After having done investigations on chosen street sections in Warsaw modelling the ride along routes was started with consideration to the below given requirements:

It was assumed (on base of measurements), that there are two types of road slopes which amounted to 1% i 2%. Hence a test schedule has been defined on each ride through a given city bus route so that a bus drives 80% on flat roads and 10% -on 2% ascending roads.

It was assumed (basing on measurements carried out by MZA-Warszawa), that in rush hours (7-11 and 16-19) the bus load amounts to 100% (and most often to

125%), while beyond the rush hours it equals 50% only. According to European standards the load should be related to percent schedule of the ridden road and thus:

% LOADS	% OF A RIDDEN ROAD
0-50	60
50-75	25
75-100	10
100-125	5

But the Warsaw criterion refers to the time of implementation of a ride process which seems to be a more correct interpretation from the point of view of stream of people observed. Comparison of the both criteria (in percentage) allows to state that the Warsaw criterion is more „rigorous” in relation to the European one in aspect of bus load amounting to 125%;

Two types of bus stops were assumed; bus bay and start from a bus stop located directly on a road;

On base of the investigations it was found that about 50% of distances between the bus stops is being taken

by start/stop process and the other 50% takes the ride. This is anyhow a non-rigorous criterion, as at small distances between the bus stops (500 m) a start/stop process was basically observed on the whole section of the ride;

Types of surface (normal, moguls tram tracks, cobblestones and narrow streets);

Types of traffic: relatively smooth traffic, traffic with non-numerous bus stops and traffic with numerous bus stops during rush hours.

Such formulated requirements have been implemented during a drive simulation process in an 8 hours

workday with consideration made that a stop at a bus stop equals 15 sec. (for a 50% load) and 30 sec (for an 100% load), which results from measurements made by MZA-Warszawa. Moreover a 15 minutes long breakfast break is proposed. In the simulation process a process of driving from the bus depot to a ride route as well as driving from the route to bus depot after finished work have also been taken into account..

Routes Nos: 189, 172, 157 and 517 were taken for simulation.

Exemplary results for the assumed drive hours from 6:30 till 14:30 for the route 157 were presented in form of diagrams:

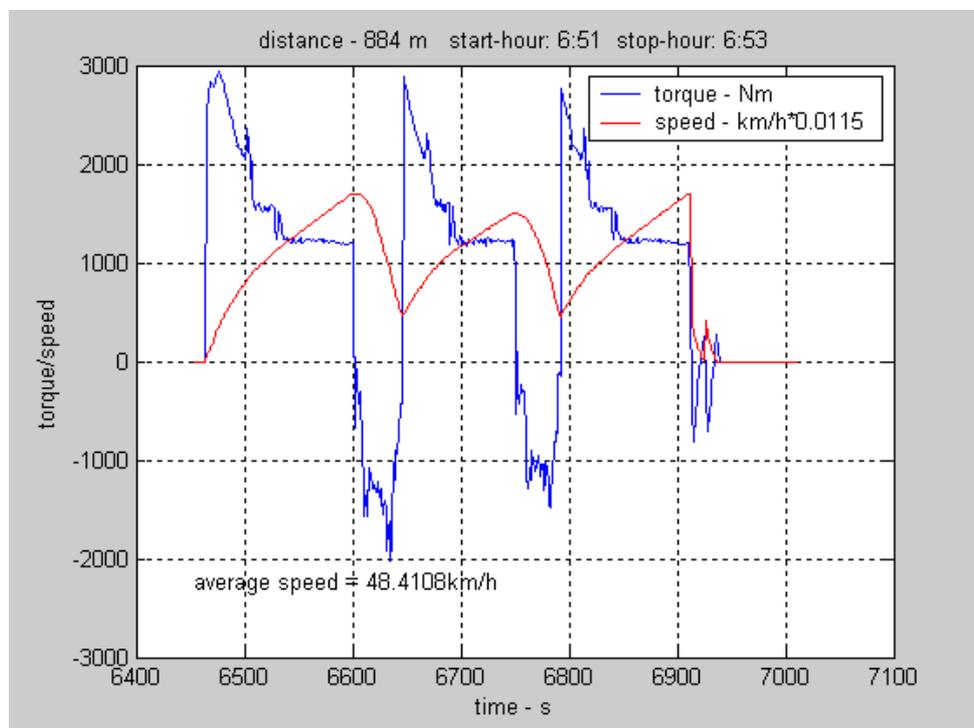


Fig. 4 Street traffic of small intensity

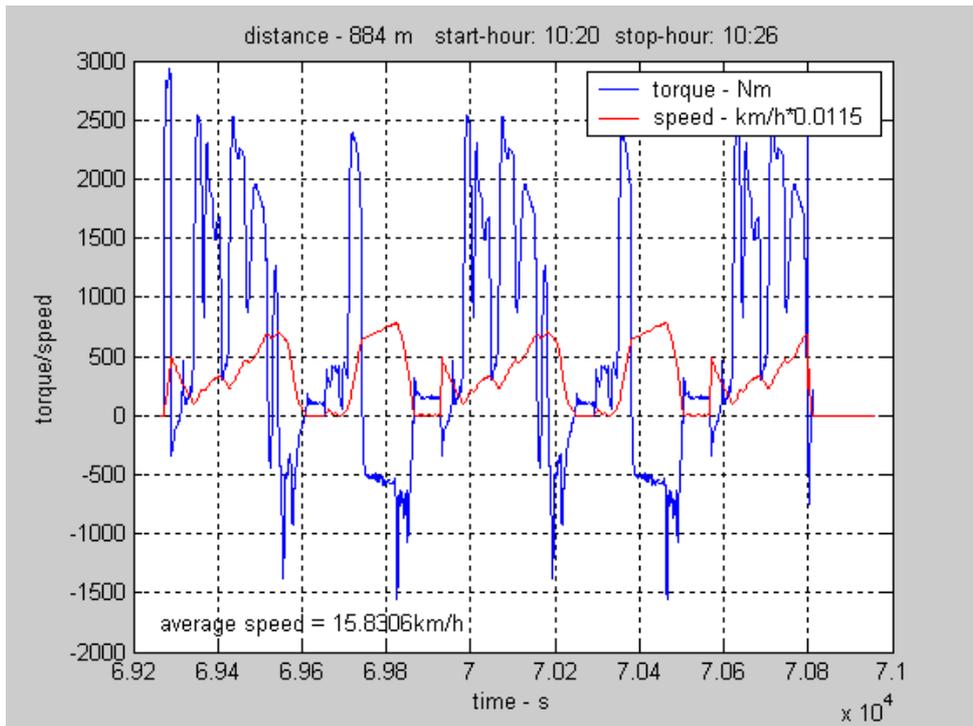


Fig.5 Street traffic of high intensity

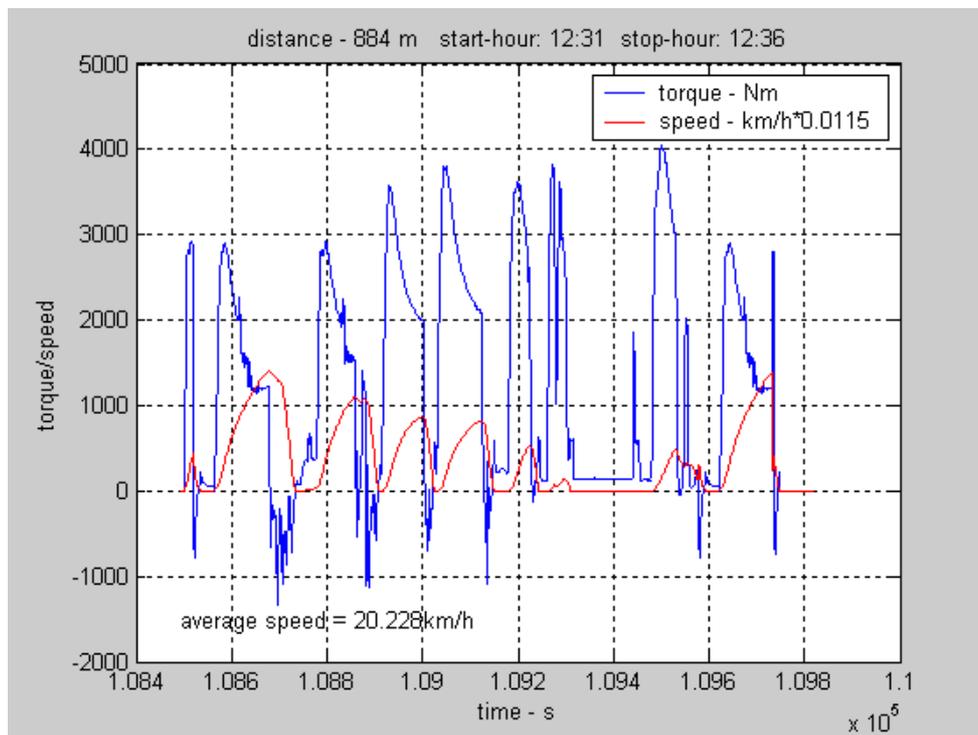


Fig. 6 Street traffic of medium intensity

It results from the diagrams shown in Fig. 4 - Fig. 6 That on the same section of the chosen route the average (medium) speed changes much in function of

hours of drive implementation.. Much more important significance in fuel consumption has a torque which increases non-linearity at the stop&go drive. Hence it

results that average running cost of a bus is a function of two variables that is the route of a specific road section as well as time of implementation of the drive.

Operational cost of this road section in function of time has been presented in fig.

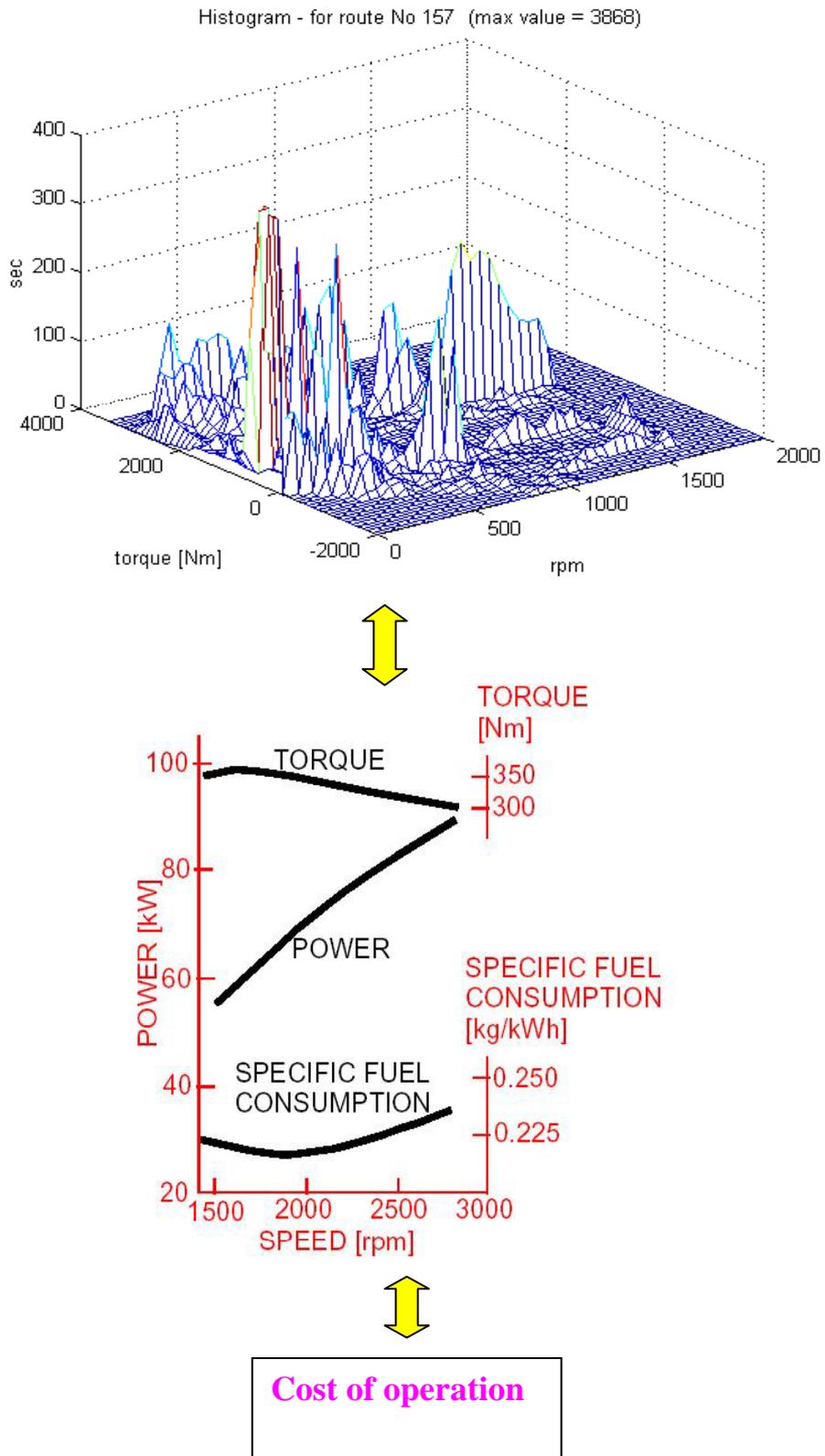


Fig. 7 Histogram of loads obtained for route 157 with operation cost of a bus in function of work time

### 3. CONCLUSIONS

On base of the analysis carried out of the processes occurring in buses operated the running costs should be assumed for modelling transportation systems in aspect of their optimization as a function of two variables – road and work time, which is directly connected with the traffic intensity.

### REFERENCES

- Bouzaiene-Ayari, B.; M. Dror, G. Laporte. "Vehicle Routing with stochastic demands and split deliveries. *Fundation of Computing and Decision Sciences*, 1993, vol. 18, No.2, 63-69.
- Osmólski, W.; S. Krzyżaniak. „Simulation Research Methods In First Joint Conference of International Simulation Societies”, Zurich, 1994, 525-529.
- Bielli M.; G. Improta. „Integration of operations research and artificial intelligence for Urban traffic control and management. *Proc. Of 6 WCTR (Abstracts)*, Lyon, 1992, No 858.
- Saxena, P.K. "Lexicographic bi-criteria transportation problem with additional restrictions. *Proceedings of 10 International Conference on MCDM*, Taipei, 1992, vol. IV, 241-249.
- Willumsen, L.,G. Simplified Transport Models based on Traffic Counts, *Transportation* 10, 257-278.

### AUTHOR BIOGRAPHIE



**WALDEMAR OSMÓLSKI** was born in Poznań, Poland and went to the Technical University of Poznań, where he studied cars and vehicles construction and obtained his degree in 1988. He worked for Transmeble SCANSPED Sp. z o.

o. ( Swedish carriage and forwarding company ) as logistics and development manager, then in Raben Transport Sp. z o. o. ( Dutch carriage and forwarding company ) as dispatcher , selling and administration manager and from 1-st December 1996 till today - managing director, Ponetex Logistics and Customs Agency Sp. z o.o. (Dutch logistic company).

# DECISION SUPPORT SYSTEM IN CITY LOGISTICS

Waldemar Walerjańczyk,  
Michał Maciejewski  
Instytut Maszyn Roboczych i Pojazdów Samochodowych  
Politechnika Poznańska  
ul. Piotrowo 3, 60-965 Poznań, Poland  
e-mail: [wal@sol.put.poznan.pl](mailto:wal@sol.put.poznan.pl),  
[michal-m@o2.pl](mailto:michal-m@o2.pl)

## KEYWORDS

City Logistics, Decision Support Systems (DSS), Data Center, Geographic Information Systems (GIS), Global Positioning Systems (GPS), Multiple Objective Optimization, Hybrid Genetic Algorithms

## ABSTRACT

Finding efficient and cost acceptable solution for supporting processes in city-logistics leads to multi-user multi-layer hybrid simulation systems incorporating latest technology achievements: from sophisticated computer and database systems thru latest large data set computing metaheuristic algorithms to Global Positioning Systems and other status update systems (exceptional traffic, hour/weekday traffic levels). In this paper example approach to design of Decision Support System (DSS) in city-logistic is presented. This includes proposition of multi-layer simulation system that incorporates existing hardware and software solutions (Data Center technology, TransCAD as pre and post processing GIS) and new metaheuristic solution algorithms (with experimental verification) and system topology design which make it more suitable for needs and economic capability of potential users. Although this paper concerns preliminary approach to creation of such system - currently extensive work is done to fully implement ideas presented and successfully verified on benchmarking sets.

## INTRODUCTION

Urban freight transport is an extremely important activity in the context of urban life. Efficient transport is an important element of the urban economy, both in terms of the income it generates and the employment levels it supports. However, freight transport is responsible for traffic and environmental impacts in urban areas. The research has attempted to gain tools for simulating and handling the problems experienced by goods and service vehicles in urban areas:

- Traffic flow/congestion problems
- Parking and loading/unloading problems
- Customer/receiver-related problems

The enormous computing power available on the desktop, at nearly the price of electricity, enables a graphic, interactive, GIS-based approach to transportation decision support systems that can be

accessible to almost every organization. This paper concerns the usage of advanced technologies that enable development of logistics decisions supporting software, proposes new optimization methods that have not yet been deployed in practice and shows achievements of metaheuristics algorithms for solving example problem: the Capacitated Vehicle Routing Problem (CVRP).

## PROBLEM IDENTIFICATION

Any decision support system designed for City-Logistics should incorporate tools/methods that can simulate and handle with typical problems concerning traffic flow/congestion:

- weekday traffic levels
- exceptional traffic incidents
- seasonal variation in traffic levels
- inadequate road infrastructure
- lack of traffic problems information
- vehicle access time restrictions
- vehicle weight (and other) restrictions
- permanent road closures
- road design/layout

This leads to very expensive and overloaded database systems (with frequent and various updates) neither accessible nor maintainable to many organizations. Even though data acquisition for such system seems to be possible nowadays it reaches economical barrier and can be implemented only in multi-user shared computer architecture based on third party Data Center technology as described in next chapter.

The only acceptable solution for this problem is creation of integrated computer system in one of two proposed architecture configurations: centralized or semi-distributed. First one gives economy savings while second gives better flexibility to potential user. Both assume usage of TransCAD from Caliper Corporation as core solution and visualization application with add-ins expanding its functionality and providing task-oriented user-friendly interface along with Mainframe Computers in Data Center as outsourcing solution. This approach gives great functionality and efficiency at reasonable price and guarantee professional maintenance and security of stored data. Participation of many companies in that project gives additional benefits – decision support system can easily obtain and use information about traffic generated by other participating companies. Of course this leads to another

problem with privacy of information – any participant is interested in usage of such information but no one is interested in giving information about his activity, clients, fleet status etc. System proposed in this paper gives complex solution to this problem through access control in communication module and professional solution provided by Data Center.

The application of transportation models and solutions has always been a computationally intensive process. System proposed in this paper requires application of new methods of optimization for many transportations problems such as Vehicle Routing Problem, which is a complex combinatorial problem (and the most representative one). Due to its NP-hardness it is almost impossible to find the optimal solution for big instances of the VRP even though mainframe computer power is used. That is why many heuristic algorithms have been proposed and used for solving the problem.

This paper relates to the usage of metaheuristics algorithms for solving the Capacitated Vehicle Routing Problem (CVRP) as an example solution that should improve computational efficiency of proposed multi-user decision support system.

### SYSTEM CHARACTERISTICS

System is designed as a multi-level scalable computer system based on Data Center technology and TransCAD application as core solution and visualization engine implemented in one of two possible topologies:

- Centralized – where participants can use terminal-like technology, which offers full functionality of the system but limits possibility to modify or match specific requirements of participants while offers lowest possible operating cost.
- Semi-distributed – giving great flexibility due to possibility of implementing completely new modeling procedures and joining additional data not available in central database while still having full functionality of the system but with higher operating cost (installation and maintenance of additional database systems on client computers)

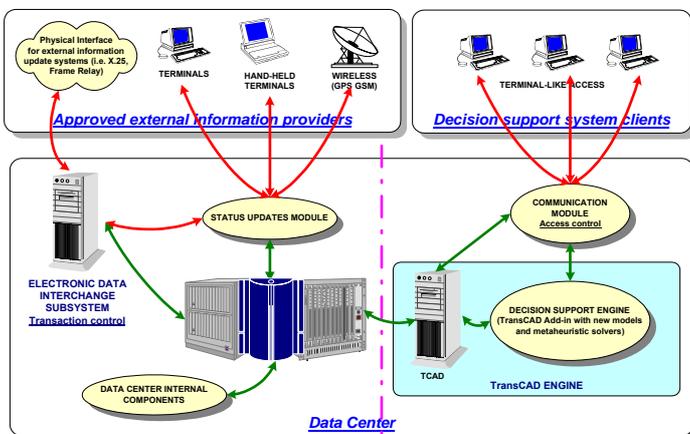


Fig. 1 Modular architecture of Centralized System

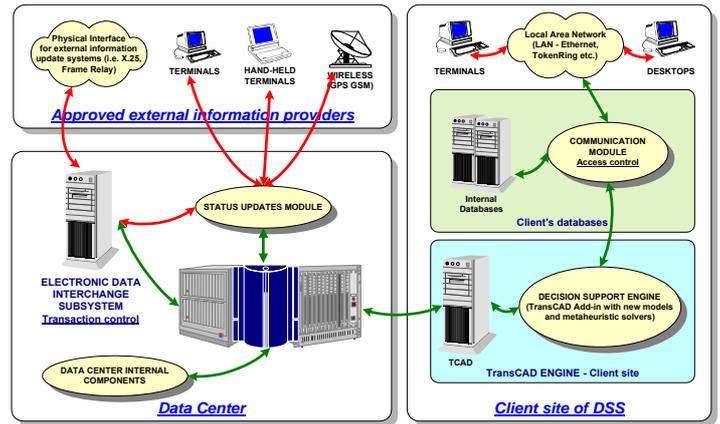


Fig. 2 Modular architecture of Semi-Distributed System

As shown on figures 1 and 2 in both solutions some elements are common so obvious that are not even detailed in this paper and only few are different and solution-dependent.

The most important element of the system is solution and visualization engine TransCAD from Caliper Corporation. This Geographic Information System has been chosen due to “natural gift” for transportation application. It can be replaced by any other GIS system offering similar functionality. TransCAD was designed as an open platform that facilitates the addition of user-written and third party extensions. The Geographic Information System Developer’s Kit (GISDK™) for TransCAD provides the tools for creating add-ins to TransCAD, macros to automate repetitive tasks, and the ability to implement custom model interfaces. Add-ins can implement completely new modeling procedures that can access TransCAD data or subroutines and thru the TransCAD functionality can access almost any database system. In this way, TransCAD is designed to avoid the requirement of massive code-writing for new analysis methods, and provides a cost-effective means of implementing new models.

The companies that have participated in the discussion held during the research have had the opportunity to express what actions they might want to be implemented in order to make the supply of goods and services easier to perform. As the result of this discussion Decision Support System architecture was designed (shown on fig.1 and fig 2) as the solution that fully satisfies potential participants of this project.

As proposed both solutions may coexist in single integrated multi-user multi-architecture system offering both (but exclusively) economic and flexible access to Decision Support System module and information stored and maintained in Data Center.

As another result of discussion a need for flexible user-friendly and task-oriented user interface has been denoted. Decision Support System should simplify and minimize user interaction with system when frequent and schematic functions are executed. In many cases user interaction can be minimized when information about past activity would be analyzed and reused – but

this leads to an expert systems not considered in this paper. On the other hand modern applications offer great flexibility at the cost of clarity or simplicity. Due to proposed architecture of DSS this inconvenience along with language/localization problem is solved – DSS is masking underlying solution and visualization system offering convenient user-friendly and task-oriented interface (see figure 3)

Not only the computer power and storage capacity is crucial when deciding which offer to accept – one of the most important is client access to this resources i.e. offered connection capability. Talex S.A. Data Center WAN has star topology based on Frame Relay Polpak-T. Central point (Poznań) is connected via PVC links to its divisions in Warsaw and Wrocław. Every division has additional PVC link to

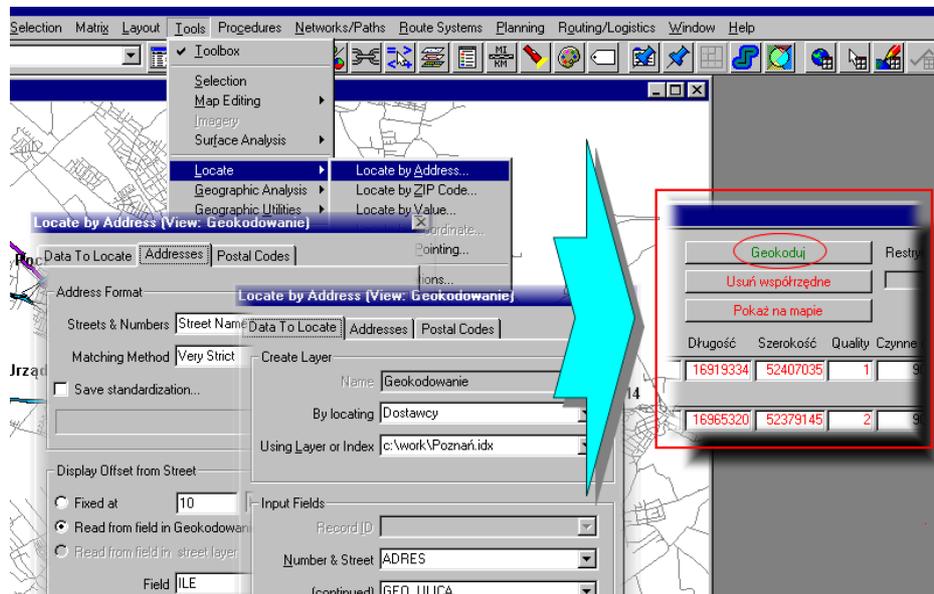


Fig. 3 DSS Masking Feature resulting with Localized and Optimized Interface

This approach creates real multi-level system where any underlying element can be replaced by any other solution providing compatible inter-level communication capability (i.e. when user interface calls sub layer for geocoding new localization it waits for execution status and after success assumes that all required data is written/updated in proper databases – existence of TransCAD in this example is not required nor checked, operation can be done by any other GIS application - MapInfo Professional for example)

## DATA CENTER

As previously mentioned implementation of proposed system for many reasons is possible and cost effective only by means of outsourcing specialized companies offering Data Center technology. In our region one of the most well-known companies offering such solutions is TALEX SA

Their Data Center as certified by IBM fulfills rigorous requirements concerning:

- Localization,
- Servers room construction,
- Operational continuity,
- Telecommunication infrastructure,
- Professional client care, etc.

and thus seems to be the best solution for proposed intensive, heavily loaded computer database system

Internet (TPNet). SDH link of 34 Mb/s bandwidth allows 17 clients connections of 2 Mb/s each. Although proposed system is designed for city logistics, possibility of wide network access can be helpful or required when decision centers are localized in other cities (in Warsaw as capital for example).

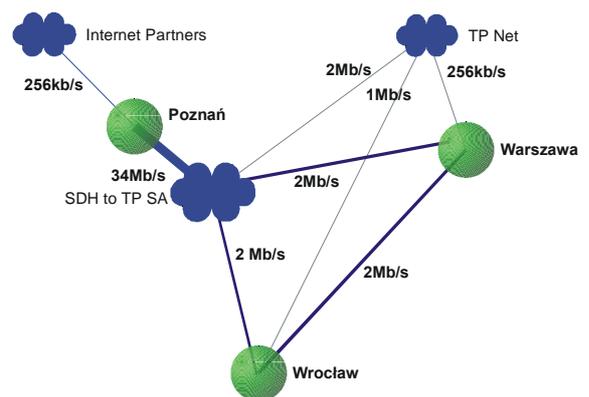


Fig. 4 Talex S.A. Data Center WAN

## MULTIPLE OBJECTIVE METAHEURISTICS IN VEHICLE ROUTING PROBLEM

Decision Support System as described in this paper requires new approach to standard modeling procedures. This section describes one of the most representative:

multiple objective metaheuristics in vehicle routing problem. (designed and verified on simulation data as add-in expanding functionality of TransCAD solver)

In many real-world problems it is necessary to take into account many different and often conflicting objectives. In such problems there is no single optimal solution, but a set of many alternative solutions. At present multiple objective metaheuristics are a very active field of research.

In this chapter author's achievements in this field of research are summarized (for details see [1]).

In our research an extended Capacitated Vehicle Routing Problem (CVRP) has been studied. This extension involves multiple objective optimization. Three independent objectives have been selected arbitrary: minimization of total traveling distance, traveling time and number of vehicles.

Our multiple objective metaheuristics for CVRP have been based on Multiple Objective MetaHeuristic Library in C++ (MOMHLib++). MOMHLib++ is a library of C++ classes that implements a number of multiple objective metaheuristics. It has been developed by Jaszkiwicz (2001). In this library main multiple objectives metaheuristics are implemented in a consistent way, which allows easy adaptation of the methods to a given problem.

As our main research interest lies in evolutionary computations, genetic algorithms have been selected from the MOMHLib++. These are (with working abbreviations used in the further part of this paper):

1. Strength Pareto Evolutionary Approach (**SPEA**),
2. Nondominated Sorting Genetic Algorithm (**NSGA**),
3. Nondominated Sorting Genetic Algorithm II (**NSGA II**),
4. Controlled Nondominated Sorting Genetic Algorithm II (**NSGA II C**).

Also hybrid genetic methods have been studied, where global optimization is hybridized with local optimization (local search method). These algorithms are:

1. Multiple Objective Genetic Local Search by Jaszkiwicz (**MOGLS**),
2. Multiple Objective Genetic Local Search by Ishibushi and Murata (**IMMOGLS**),
3. Pareto Memetic Algorithm (**PMA**).

For the comparative purposes Multiple Objective Multiple Start Local Search (**MOMSLS**) method has been analyzed.

During calculations two alternative mutation operation have been used: node exchange (NX) and arc exchange (AX). These two operations are adopted from the Traveling Salesperson Problem (TSP). In case of node exchange two points in the solution sequence are exchanged. In arc exchange operation the sequence between two points (including the second of the two selected points) is inverted. Each change in arcs is

connected with changes in time and distances traveled by vehicles.

Another operation is neighborhood search operation, which is necessary in local search optimization methods. For this purpose node exchange and arc exchange operators were used. But while in case of mutation the pairs of points are generated with randomness, in local search (where the neighborhood is searched in a systematical way) for each sensible pair the operation is performed.

Experiments have been conducted in two phases. In first phase the single objective optimization has been carried out (minimization of the total distance). It is due to the fact that there are no benchmarks for the multiple objective CVRP. The single objective optimization was conducted to prove (with success) that:

1. Our genetic algorithm operators and local search method operators are well suited to the CVRP
2. Our software (which implements the CVRP and the operators) effectively works with MOMHLib++

In the second phase experiments on the multiple objective CVRP have been carried out.

As a benchmark for the single objective CVRP optimization a number of well-known problem instances have been used. It includes instances from Augerat Set A (A-n44-k7, A-n54-k7, A-n80-k10), Augerat Set B (B-n31-k5, B-n41-k6, B-n57-k7, B-n63-k10, B-n78-k10) and Christofides and Eilon (E-n13-k4, E-n22-k4, E-n33-k4, E-n51-k5, E-n76-k10, E-n76-k14, E-n76-k7, E-n76-k8). Only few of those are presented in this paper.

For each method and for each instance two kinds of optimizations have been carried out. In the first one AX operator has been used for mutation and local search, while in the other NX operator has been used.

In case of genetic algorithms there is no significant difference in the results, so AX and NX operator are equally good for mutation. This has not come true in case of hybrid genetic algorithms and local search method, where AX and NX operators have been used for searching the neighborhood. In this case AX operator performs much better than NX. It is probably because less disturbances in solutions appear when AX is used, so this operator gives more chance to precisely search the current solution neighborhood.

But more meaningful are differences in performance between algorithms. Best results have been obtained for the hybrid genetic algorithms (especially for MOGLS and PMA). Local search (MOMSLS) has performed well. The worst performance has been reached by the genetic algorithms (especially by SPEA and NSGA, see table 2). All four genetic algorithms use Pareto-ranking for assessing fitness for each chromosome.

The poor performance of these genetic methods is due to the fact that Pareto ranking is not informative enough to carry out the optimization well.

Table 1 Example results of the single objective CVRP optimization (underlined – the best solution given in the repository of these benchmarks, **bold** - values equal to the reference values, *italics* – better than the reference ones)

Test	Op	SPEA	NSGA	NSGA II	NSGA II C	MOGLS	IMMOGLS	PMA	MOMSLS
A-n32-k5 (784)	AX	1364	1399	855	863	796	796	796	820
	NX	1348	1429	862	882	796	803	801	846
A-n44-k7 (937)	AX	1631	1778	1089	995	<b>937</b>	942	942	961
	NX	1802	1812	1042	1009	<b>937</b>	948	945	996
B-n31-k5 (672)	AX	791	854	684	677	<b>672</b>	<b>672</b>	<b>672</b>	678
	NX	840	849	713	685	677	677	677	682
B-n57-k7 (1153)	AX	2289	2565	1342	1489	<i>1145</i>	<i>1147</i>	<i>1148</i>	1177
	NX	2462	2602	1185	1345	<b>1153</b>	1155	<i>1147</i>	1221
B-n78-k10 (1266)	AX	2979	3195	1591	1442	<i>1229</i>	<i>1242</i>	<i>1224</i>	1302
	NX	3132	3234	1494	1435	1279	1303	<i>1257</i>	1377

Table 2 Computation times (in seconds) of the single objective CVRP optimization

Test	Op	SPEA	NSGA	NSGA II	NSGA II C	MOGLS	IMMOGLS	PMA	MOMSLS
A-n32-k5	AX	1	6	7	7	14	18	9	2
	NX	1	6	7	8	15	21	10	3
A-n44-k7	AX	1	6	8	8	43	58	41	7
	NX	1	7	7	8	44	73	38	7
B-n31-k5	AX	1	6	7	7	10	12	8	1
	NX	1	6	7	8	12	18	12	1
B-n57-k7	AX	2	7	7	8	147	179	119	23
	NX	1	7	8	8	180	196	126	23
B-n78-k10	AX	2	8	8	9	427	634	370	76
	NX	2	7	9	8	569	765	451	91

Table 3 Computation times (in seconds) of the Multiple Objective CVRP Optimization

Test	S	SPEA	NSGA	NSGA II	NSGA II C	MOGLS	IMMOGLS	PMA	MOMSLS
A-n32-k5	C	2	6	8	8	18	24	14	1
	L					14	24	12	2
A-n44-k7	C	1	7	8	8	57	78	53	6
	L					50	82	46	7
B-n31-k5	C	1	7	7	8	15	21	13	2
	L					13	21	12	2
B-n57-k7	C	2	7	8	9	93	136	90	10
	L					212	297	199	24
B-n78-k10	C	2	7	9	9	611	988	606	73
	L					579	987	582	80

The experiments have shown that methods based on Pareto ranking perform much poorer than those based on scalarizing functions. Computation times are presented in table 2. Algorithms using local search method works much slower, because this method is extremely time consuming. Since parameters of all methods were constant, they do not depend on n, where n is the size of the problem, what determines the size of solution representation. Time results only depend on n.

In case of the genetic algorithms computation times are rather constant (they minimally increase with increasing n). This is due to the fact  $O(n)$  is the complexity of operations on chromosome such as reproduction or calculation of objective values. But these operations are less time consuming in attitude to the rest, thus so minimal impact of n on computation times.

In case of the hybrid genetic algorithms and the multiple start local search algorithm things change. Local search is time consuming to such a large extend, that computation times are very high. The size of neighborhood in the local search is  $O(n^2)$  for both AX and NX neighborhood operators. Exactly neighborhood

sizes are  $n(n-3)/2$  and  $n(n-1)/2$  respectively. A single search of the neighborhood is  $O(n^3)$  because of additional complexity related with operations on chromosomes. But the results show that the total complexity is  $O(n^4)$ . This fact can be explained in the following way. The number of all possible solutions (assuming our solution representation) is  $n!$ , but during a single iteration of local search less than  $n^2/2$  are analyzed. Thus when n increases it takes more iteration to find the local optimum, because the neighborhood becomes smaller in relation to the whole solution space. The second phase experiments have been carried out only for AX operator (because of better performance). On the other hand computations for methods incorporating local search have been conducted for two variants of scalarizing functions (linear [L] and Chebyshev [C] functions). Instead of presenting the results in table appropriate diagrams illustrate the results.

Next three diagrams (figures 5, 6 and 7) present the results obtained for A-n32-k5 instance. All solutions in the set have the same value on the third objective – in

all of them 4 vehicles are required. For greater clarity all points are connected and create an approximation of the Pareto front.

In the multiple objective optimization again the hybrid genetic algorithms perform the best (especially MOGLS and PMA), while the genetic algorithms give the poorest results. Another conclusion after detailed analysis is that linear scalarizing function (L) is better than Chebyshev (C) one.

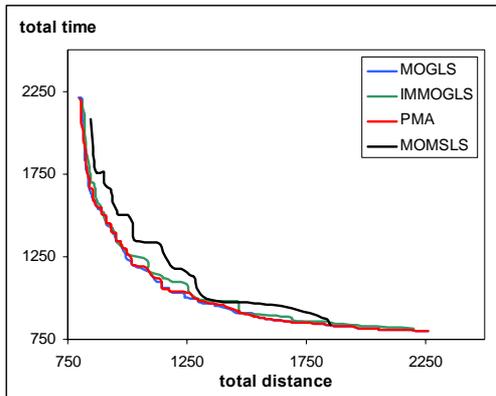


Fig. 5 Results for A-n32-k5 using linear scalarizing function (Hybrid Genetic Algorithms)

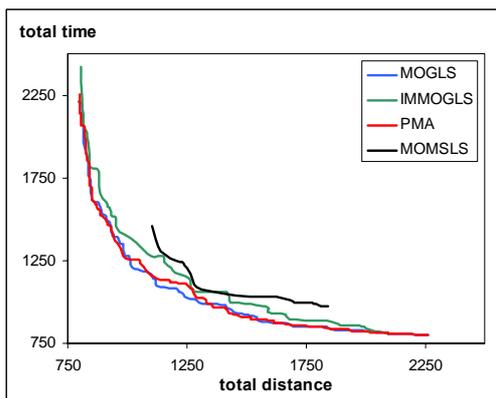


Fig. 6 Results for A-n32-k5 using Chebyshev scalarizing function (Hybrid Genetic Algorithms)

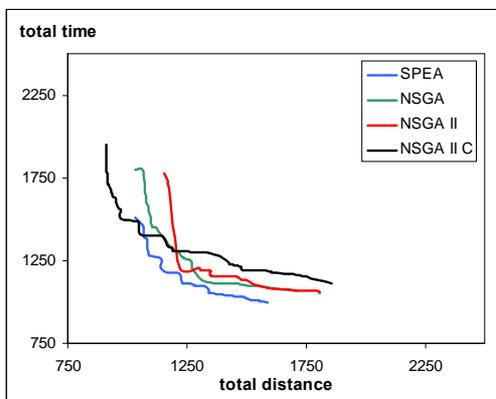


Fig. 7 Results for A-n32-k5 (Genetic Algorithms)

The second phase experiments also proved that PMA and MOGLS give best results, IMMOGL is quite worse

while the genetic algorithms using Pareto-ranking perform worst.

A second conclusion is that computation times in case of the multiple objective CVRP are longer. It is due to the fact that in the single objective version approximation of the nondominated set consists only of one solution, while in the multiple objective version – lots of solutions create the approximation.

Results achieved for test problems authorize us to evolve algorithms into system proposed in this paper and make experiments on real life examples. Currently extensive work is done towards integration of presented algorithms with designed DSS for city logistics and TransCAD to gain access to real life data (with proposed Data Center technology), visualization engine and localized user-friendly interface.

## CONCLUSIONS

The key aim of the exploratory research was to provide practical tools for use by public sector organizations, freight transport operators and industrial and commercial firms who are working towards establishing a more sustainable urban environment.

In this paper the multiple objective approach to the Vehicle Routing Problem has been presented. Multiple objective metaheuristics are quite a new, but fast evolving research domain. There have been many successful attempts to apply these metaheuristic methods to such problems as the Traveling Salesperson Problem or the 0/1 Knapsack Problem. In case of the VRP an algorithm for bi-objective optimization was proposed and then applied it to the bi-objective VRP. The approach presented in this paper allows using of more than only two objectives and with other elements offers efficient and cost acceptable solution for supporting processes in city logistics

Although this paper concerns preliminary approach to creation of presented system – in fact currently extensive work is done to fully implement ideas presented and successfully verify on benchmarking sets.

## REFERENCES

- Jaszkiewicz A. 2001. “Multiple objective metaheuristic algorithms for combinatorial optimization”. Wydawnictwo Politechniki Poznańskiej, Poznan.
- Landeghem van H. R. G. 1988. A bi-criteria heuristic for the vehicle routing problem with time windows, *European Journal of Operational Research*, vol. 36, pp. 217 - 226.
- Maciejewski M. and Walerajańczyk W. 2003. „Solving Vehicle Routing Problem using multiple objective metaheuristic algorithms” MASS 2003, Genova, Italy
- Narbuntowicz E. and Walerajańczyk W. 2003. „Simulation method of public logistics center localization in Poznan city” ISC 2003, Valencia, Spain, pp. 276-280.
- Sung Chul Hong and Yang Byung Park. 1999. A heuristic for bi-objective vehicle routing with time window constraints, *International Journal of Production Economics*, vol. 62, pp. 249 - 258.
- MOMHLib++ code and documentation site: <http://www-idss.cs.put.poznan.pl/~jaszkiewicz/MOHMLib>

# DECISION SUPPORT SYSTEM AND REGULATION SYSTEM FOR ROAD TRAFFIC MANAGEMENT

Sylvain Lerebourg, Antoine Dutot, Cyrille Bertelle and Damien Olivier  
Laboratoire d'Informatique du Havre, 25 rue Philippe Lebon, 76600 Le Havre

E-mail: {Sylvain.Lerebourg, Antoine.Dutot, Cyrille.Bertelle, Damien.Olivier}@univ-lehavre.fr

## KEYWORDS

Road traffic, ant algorithm, multiagent systems, neural networks

## ABSTRACT

Decision support system for road traffic management can be used for freight transport, people transport but also for site evacuation. We deal with two aspects of the decision support system in a same global architecture: one for traffic regulation to avoid jam and the other for road users to choose the shortest path in time between two points. These two aspects interact. The cartography is represented by a weighted digraph. The weights evolve according to the traffic and the graph is therefore dynamic. The regulation system is based on a neural network. The shortest path is based on an ant algorithm well suited for dynamic environments.

## GLOBAL ARCHITECTURE FOR ROAD TRAFFIC MANAGEMENT

The transport development must face up to many constraints like: substructures realization and expansion limitation due to the available space and the costs, reduction of the loud and atmospheric pollution, deregulation and concurrency between the mode of transport and so on. So, it is necessary to find solutions to manage road traffic. Two aspects can be considered. The first one is about Decision Support System (DSS) to help and inform users. The second one is about regulation system based on control aménagement (Virtual Message Signs (VMS), traffic lights, ...).

We propose a global architecture based on two main parts (see figure 1):

- The real world which is split in three elements:
  - the traffic which contains, in one hand, all mobile elements (cars, pedestrians, ...) described with different levels of autonomous behaviour and, in the other hand, spatio-temporal organizations which are predictable (school outs, ...) or not (jam, accident, ...);

- the environment which contains all the road infrastructure and logistic planning;
- the control system which contains sensors (webcams, data traffic magnetic sensors, ...) and effectors (VMS, traffic lights, ...).

- The model which is split in the following elements:
  - information collection and processing in order to use them on the solving level;
  - a dynamic weighted digraph representing these informations and the traffic flow;
  - a regulation system based on this graph and managing the control system;
  - a DSS which use the dynamic graph and the regulation control. A multimodal interface informs and helps different users with respect to their profiles.

The information update and its adaptive treatment give the dynamic aspect of the global architecture as described in figure 1. So, it is typically a complex system model including retro-action phenomena. In this paper we develop two points of this architecture: the regulation system based on multi-layer perceptron with backpropagation algorithm and the decision support system which suggests shortest paths obtained from a dynamic graph.

## REGULATION MODELLING

The model for road traffic regulation uses an agent-based representation for road traffic and a neuronal model for the regulation. This study (Foote, 2002) presented in the following will look at traffic flow on a Manhattan-style road grid. At each crossroad, there is a traffic lights system deciding which cars are going to cross. Cars enter the grid from the outside and decide which direction they wish to use at each crossroad. We use the Madkit package (Gutknecht and Ferber, 1997) to manage the agent world. The neural network is a multi-layer perceptron implementing a backpropagation algorithm.

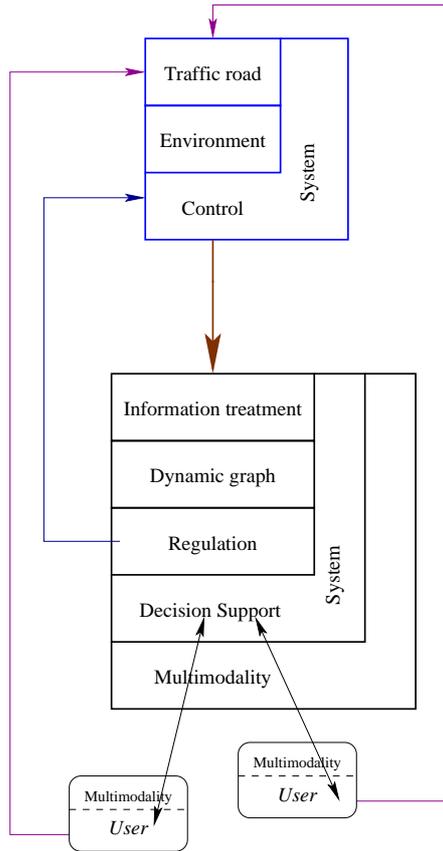


Figure 1: Global architecture

### Controlling Multi-Agent System

The generic agent organization used in this work is based on a theoretical study of A. Cardon (Lesage et al., 1999). He describes how a multi-agent system can be divided into three types of agents:

- *Aspectual Agents* are the basic agents that represent the target population. In our case, they represent cars and traffic lights in a town;
- *Morphological Agents* deal with aspectual agents measurements. They collect only some informations which lead to describe evolutive and adaptive organizational aspects. It is a kind of projection of all agent characteristics onto a smaller dimensional space. In our case, a morphological agent plays a statistic collection service, taking into account for example, cars position and information about their displacements.
- *Analytical Agents* are some rulers of our agent population, looking at the statistics provided by the morphology agents, and then acting on them to control the global behaviour of the system. The analytical agents do not directly modify the behaviour of any particular agent, but rather, indirectly shape the evolution of the aspectual agents as a whole.

Here, we present an application of this theoretical model, based on a neural network. Our problem is as follows: how can we maximise the flow of traffic through a road network? The input layer of our neural network will process information from the morphological space of the aspectual agents and then give an output figure which represents the global state of the network. This figure can then be used to decide on the action to be taken to increase traffic flow.

### Neuronal Approaches Based Models

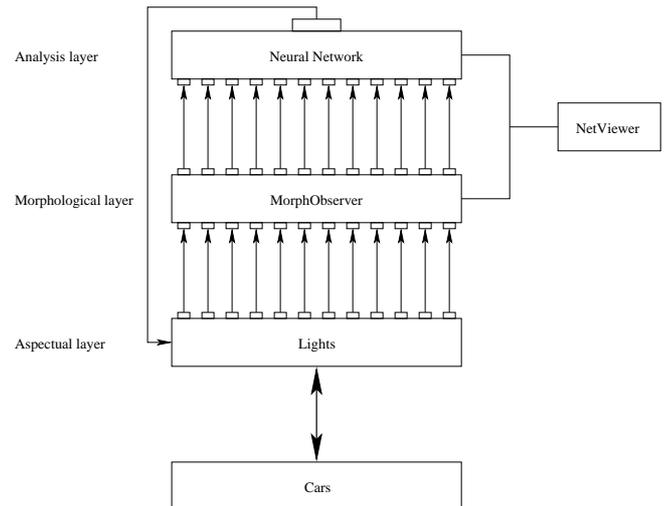


Figure 2: Neural network based regulation

The regulation model uses an agent-based description which is analysed by a neural network based on multi-layer perceptron.

### Agent-Based Description

The simulator is decomposed in three main parts:

- The *environment* is a bidimensional grid composed of set of roads which have their own length and width and where cars evolve;
- The *traffic lights* manage the cars circulation at each crossroad. Each one finds out the identities of its neighbours, it looks for cars which arrive at its crossroad and knows the direction that each car wants to go. A *cooperative light* mode is defined and proceed sorting car queues. The longest queue is first managed and the associated light lets cars go to their chosen direction if space is available, else the second-longest queue acts and so on ...
- The *cars* can be in one the three following states. They are in the state *moving* when they have to go to one crossroad (graph node) to another if there is no car in front of it. When a car reaches its chosen crossroad

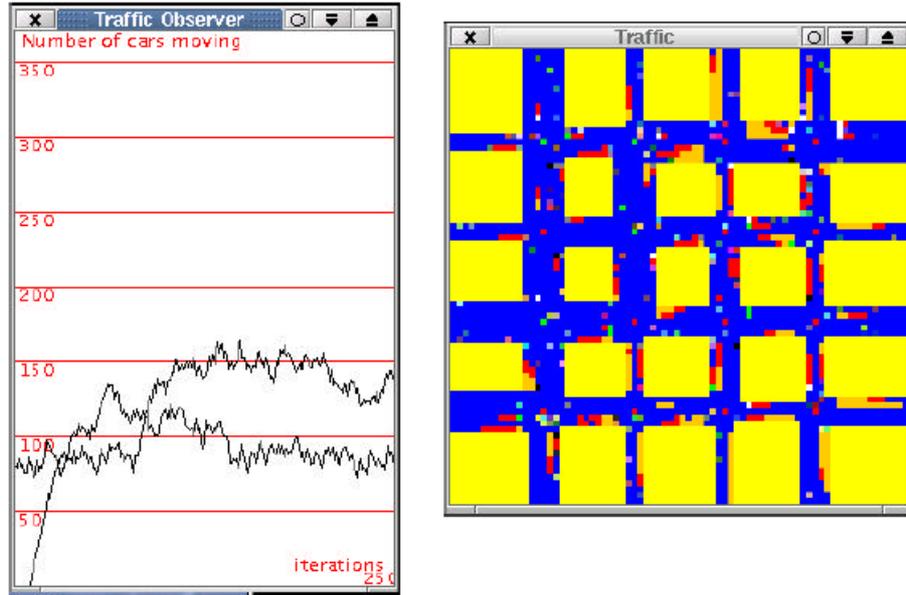


Figure 3: Regulation experimentation

without having other cars in front of it, it changes its state to *atLight* one. In this state, it sends a message to the light telling which way it wants to go. It then waits until the light gives it permission to move. If a car has other ones in front of it, during its move, it changes its state to the *waiting* one.

Moreover, the simulation manage input and output fluxes between the simulated town (as Manhattan-style road grid) and the exterior.

### Multi-Layer Perceptron Regulation

The neural network used for the regulation is a multi-layer perceptron. It is the analysis layer of the generic agent organisation described previously (see figure 2).

The network computes global variables to reduce traffic jams. The three states of its output are: clear, busy and getting blocked corresponding to no danger of gridlock, slight danger and danger of gridlock. So the retro-action of this analysis layer on the aspectual layer consists in altering the following variables:

- *waitTime* corresponds to the delay between sending batches of cars through the lights;
- *carDispersion* corresponds to the authorized cars number able to come into the town from the exterior.

### Experimentations

We show in figure 3, two windows of the visual interface desktop of the simulator which represent respectively a

schematic view of the traffic and a traffic observer curve. This last information gives the number of cars which are moving at each step of the simulation. In this example, the regulation leads to the preservation of the fluidity, the global number of moving cars is preserved between 100 and 150 units.

### CONSTRAINED PATHS COMPUTATION BASED ON ANT ALGORITHM

Ant algorithms are a class of meta-heuristics that can yield near-optimal solutions to hard optimization problems. They maintain a population of agents that exhibit a cooperative behaviour (Langton, 1987). For example, ants deposit *pheromones* in the environment that influence others which tend to follow it. Such an approach is robust and well supports parameter changes in the problem. Ant algorithms has been applied successfully to various combinatorial optimization problems like the Travelling Salesman Problem (Dorigo and Gambardella, 1997), routing in networks (Caro and Dorigo, 1997), (White, 1997), for distributed simulation (Bertelle et al., 2002b) but also to DNA sequencing (Bertelle et al., 2002a), graph partitioning (Kuntz et al., 1997) and clustering (Faieta and Lumer, 1994).

### Dynamic Graph and Regulation Feed-back

The cartography is represented by a weighted digraph  $G = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  is a set of vertices representing crossroads or any other significant information (school, town hall ...) and  $\mathcal{E} = \mathcal{V} \times \mathcal{V}$  is a set of directed edges  $e = (v_i, v_j)$ . Thus each segment of a street that is between

two adjacent vertices as defined previously is represented by either one or two directed edges. Two directed edges, one in either direction, are used if the street is two-way, and a single directed edge is used if it is a one-way segment. The edge weight  $w_{ij}$  between the vertices  $v_i$  and  $v_j$  is a dynamic factor which represents the time to cross the edge  $(v_i, v_j)$  and the traffic load which is computed by the regulation system, as described in the following.

The regulation system which acts on *waitTime* variable, is able to give  $N_{ij}$  the cars number on each edge of the graph modelling the road traffic. Taking into account some physical characteristics of each road modeled with edge, a characteristic fluidity-based time, expressed as:

$$\delta_{ij} = \left( \frac{F_{ij}}{N_{ij}} \right)^{-1}$$

This characteristic time contributes to the regulation feedback of the regulation system on the dynamic graph modelling the road traffic. In fact, each edge weight is the sum of an observed time for crossing the road, noted  $r_{ij}$  and the characteristic fluidity-based time defined above:

$$w_{ij} = r_{ij} + \delta_{ij}$$

Weights evolve according to the traffic and the graph is therefore dynamic and we have to find paths in this graph. These changes are one of the major motivations for using ant algorithms. Ant algorithms which are well suited for that kind of dynamic task. This approach is implicitly distributed. This would not create many communications since the algorithm only uses local informations and stores results directly in the graph (that is, directly in the computing resources local memory).

## Algorithm

We search in the graph some paths between two vertices  $v_0$  and  $v_n$ . The resolution method is distributed and based on auto-organization mechanisms. We continually release numerical ants on the dynamic graph, and allow them to find routes between pairs of vertices. The ants deposit numerical pheromones on edges. The amounts of pheromone deposited is a function of the length and congestion of paths. Ants are attracted by weights of edges and pheromones. The evaporation allows to forget bad paths. The ants tend to converge on paths which are the fastest.

To be able to distribute the computation, we have divided the algorithm in two parts and for each we have a specific time.

- The environment. It is represented by the dynamic graph. Its major role is to manage the ant population, evaporation phenomenon and simulation of weights on the edges. We store also in the vertex  $v_n$  the shortest path which comes from  $v_0$ , the minimal global cost  $W_{0n}$  of the path from  $v_0$  to  $v_n$ . Due to the dynamic

change of weights the duration of the shortest path may change when another ant covers the path crossing the same vertices and we note  $t_{0n}$  the instant where the ant has found the same path. For a given step, we have:

```

t_env = discrete time of the environment
BEGIN
  birth of ants on the vertex v_0
  pheromone evaporation (see (2))
  weights update
  IF t_0n << t_env THEN W_0n = +∞ ENDIF
  // No ants on the path since a long time
  t_env = t_env + 1
END

```

- The ants. Ants try to go from the vertex  $v_0$  to the another vertex  $v_n$ . Ants manage their displacements according to times and pheromones. They also drop pheromones on edges. Three states are possible for an ant looking for food, reaching the final vertex  $v_n$ , and coming back to the source. For one ant located on  $i$  we have :

```

t_ant = discrete time for the ant
vertex = i
ant_state ∈ {search, arrived, go_back}
BEGIN
  IF ant_state == search
    THEN
      //The ant must choose an adjacent vertex to i
      V_i = set of the adjacent vertices of i which
            have not been traversed yet by the ant
      FORALL j ∈ V_i DO
        Compute the probability p_ij (see (1))
        that the ant chooses to hop from
        the vertex i to j
      ENDFOR
      Select the next vertex v_k
      according to the probability p_ij
      Wait during the time w_ik - 1
      vertex = v_k // Move to k
      IF v_k == v_n
        THEN ant_state = arrived
      ENDIF
    ENDIF
  IF ant_state == arrived
    THEN
      update if necessary shortest path
      and times t_0n
      ant_state = go_back
    ENDIF
  IF ant_state == go_back
    THEN
      pheromone deposit on path
      used by the ant (see (4))
      death of the ant
    ENDIF
  t_ant = t_ant + 1
END

```

Let  $\tau_{ij}$  be the amount of pheromone trail deposited on the edge connecting  $i$  and  $j$ ,  $w_{ij}$  the weight of the edges which

depends on the time of the traffic flow to connect the location  $i$  and  $j$ , it is a dynamic variable. The probability that an ant when it is located on  $i$  choose  $j$  is:

$$p(i, j) = \frac{(\tau_{ij})^\alpha \times \left(\frac{1}{w_{ij}}\right)^\beta}{\sum_{k \in \mathcal{V}_i} (\tau_{ik})^\alpha \left(\frac{1}{w_{ik}}\right)^\beta} \quad (1)$$

Where  $\mathcal{V}_i$  is the set of adjacent vertices of  $i$  which have not been traversed yet by the ant. The amount of pheromone  $\tau_{ij}$  on the edge  $(i, j)$  is modified by the environment and by the ants. The environment regularly updates this pheromone quantity using an evaporation rate, noted  $(1 - \rho)$ :

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} \quad (2)$$

where  $0 \leq \rho \leq 1$  and  $\tau_{ij}^{old}$  and  $\tau_{ij}^{new}$  are respectively the pheromone quantity before and after the update. An ant which has found a path between the two vertices  $v_0$  and  $v_n$  and so come back to start vertex, modify the pheromone quantity by reinforcement rate, noted  $\Delta\tau$ :

$$\Delta\tau = \frac{K}{W_{ij}} \quad (3)$$

where  $K$  is a constant and  $W_{ij}$  the global cost of the path between  $i$  and  $j$ .

$$\tau_{ij}^{new} = \tau_{ij}^{old} + \Delta\tau \quad (4)$$

## Results

We show an example of the algorithm execution, based on a simple urban representation with a ring road (see figure 4). The first graph shows the initial situation, the ring road is the fastest way then we jam it, so a new path is detected by the ants. The second graph shows the shortest path obtained which takes the ring road, the last one is the solution when the ring road is jamed. In this example, at each environment time step 10 ants are released,  $\alpha = 3$ ,  $\beta = 1$ ,  $\rho = 0.9$  and  $K = 2$ .

## CONCLUSION

We are developing an architecture of both a regulation system and a decision support system based on a dynamic graph. Ant algorithms are used and well suited for adaptive aspects and anytime approaches of dynamic traffic flow. Neural networks are used and well suited for traffic flow regulation. We actually work on future development concerning management of heterogeneous informations flows from any kind of sources (satellites, webcams, sensors) and multimodal interfaces for the different users. We are searching to extract the most important and urgent informations using organizations of cooperatives/antagonists agents. Multi-agent systems are adapted to find emergent evolutionary solutions in dynamic problems.

## REFERENCES

- Bertelle, C., Dutot, A., Guinand, F., and Olivier, D. (2002a). Dimants: a distributed multi-castes ant system for dna sequencing by hybridization. In *NETTABS 2002*, AAMAS 2002 Conf, Bologna (Italy).
- Bertelle, C., Dutot, A., Guinand, F., and Olivier, D. (2002b). Distribution of agent based simulation with colored ant algorithm. In *ESS2002*, pages 39–43, Dresden (Germany).
- Caro, G. D. and Dorigo, M. (1997). Antnet: A mobile agents approach to adaptive routing. Technical report, IRIDIA, Université libre de Bruxelles, Belgium.
- Dorigo, M. and Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.
- Faieta, B. and Lumer, E. (1994). Diversity and adaptation in populations of clustering ants. In *Conference on Simulation of Adaptive Behaviour*, Brighton.
- Foote, A. (2002). Controlling the global behaviour of a massive multi-agent landscape. a practical example using neural network. Technical report, LIH.
- Gutknecht, O. and Ferber, J. (1997). Madkit: Organizing heterogeneity with groups in a platform for multiple multi-agents systems. Technical report, LIRMM, Montpellier University, <http://www.madkit.org>.
- Kuntz, P., Layzell, P., and Snyers, D. (1997). A colony of ant-like agents for partitioning in vlsi technology. In *Fourth European Conference on Artificial Life*, pages 417–424, Cambridge, MA:MIT Press.
- Langton, C., editor (1987). *Artificial Life*. Addison Wesley.
- Lesage, F., Tranouez, P., and Cardon, A. (16-21 October 1999). A multiagent prediction of the evolution of knowledge with multiple points of view. In *KAW'99*, Banff, Canada.
- White, T. (1997). Routing with swarm intelligence. Technical Report SCE-97-15.

## AUTHOR BIOGRAPHIES

The authors belong to computer sciences research laboratory of Le Havre university (LIH). **C. Bertelle** and **D. Olivier** are associate professors and work on the thematic of natural complex systems modelling. **A. Dutot** and **S. Lerebourg** are PhD students working respectively in swarm intelligence models for dynamic repartition of distributed simulations and organizational flux modelling for complex systems applied in hydrodynamic and road traffic.

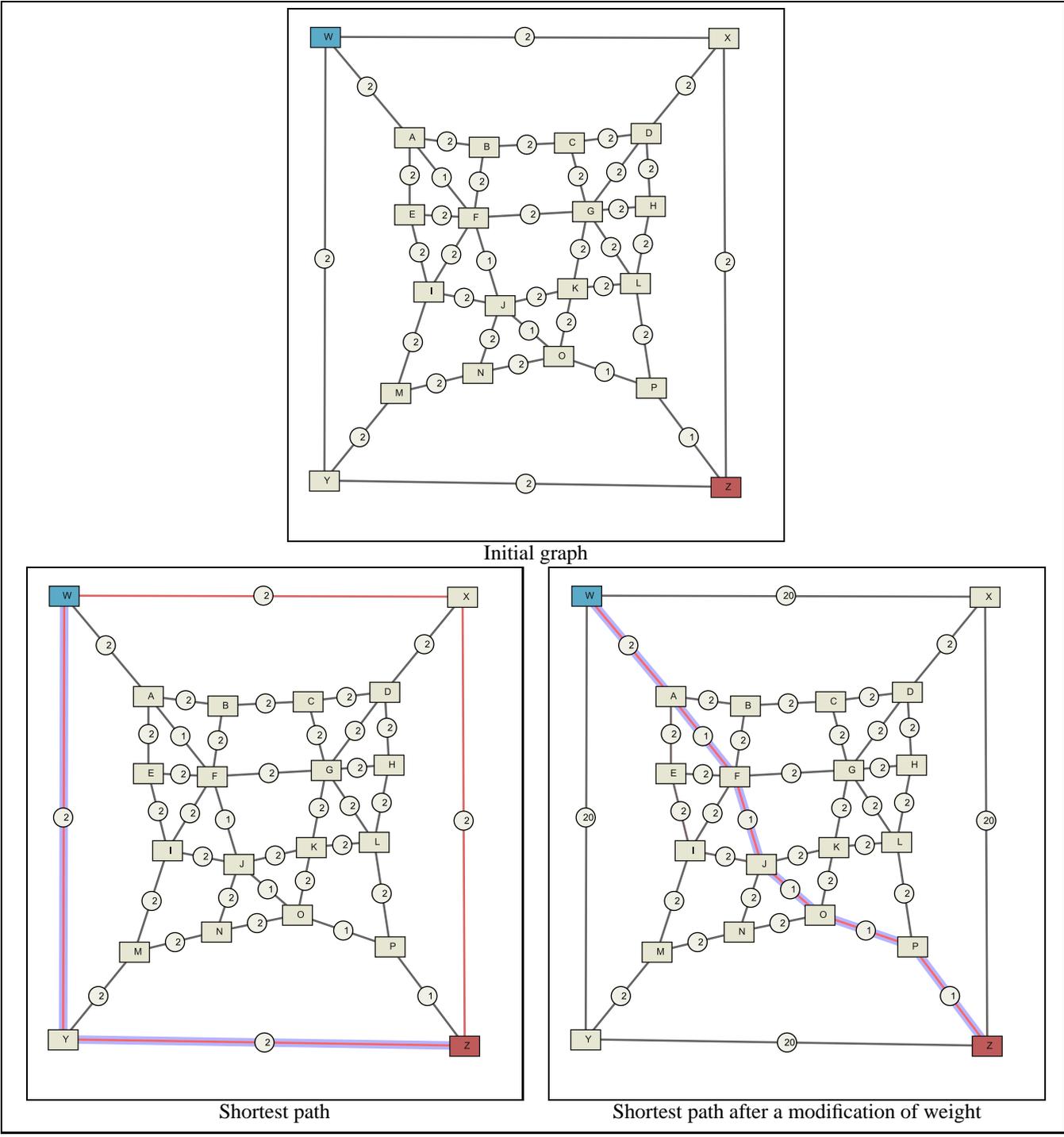


Figure 4: Search of a shortest path on a simple dynamic urban configuration

# DEVELOPMENT OF CAR DRIVE CYCLE FOR SIMULATION OF EMISSIONS AND FUEL ECONOMY

M. Montazeri-Gh and M. Naghizadeh  
Systems Simulation and Control Laboratory  
Department of Mechanical Engineering  
Iran University of Science and Technology, Tehran, Iran  
[montazeri@iust.ac.ir](mailto:montazeri@iust.ac.ir)

## KEYWORDS

Simulation, Drive Cycle, Emissions, Fuel Economy

## ABSTRACT

This paper describes the development of car drive cycle for simulation of vehicles exhaust gas emissions and fuel economy. In this research work, the development of light vehicle drive cycle for city of Tehran has been performed. First, the approach to develop a drive cycle has been described. The necessary measuring tools and software have then been provided, and the car speed has been recorded in the city traffic condition. Based on the analysis of the recorded data, a drive cycle has been developed for city of Tehran. Applying the statistical analysis, the cycle has then been modified, and a real world smoothed cycle has been presented. Finally, characteristics of this cycle have been compared with some of light vehicles cycles provided for other countries.

## INTRODUCTION

Air pollution is a major problem for big cities, and vehicles are the most important air pollution source. In addition, the fuel consumption of city cars is a great share of energy usage. Therefore, it is necessary to assess the vehicles exhaust gas emissions and fuel economy using both simulation studies and laboratory tests. Both simulation studies and test procedures are performed based on drive cycles [Bata and Yacoub 2000]. Figure 1 shows the vehicle gas emissions and fuel economy simulation model using advanced vehicle simulator (ADVISOR) where the first block is drive cycle.

A drive cycle is a speed-time sequence developed for a certain type of vehicles in a particular environment to represent the driving pattern with the purpose of measuring and regulating exhaust gas emissions and monitoring fuel consumption, as shown in figure 2.

As driving patterns vary from city to city and from area to area, the available drive cycles obtained for certain cities or countries are not usually applicable for other cities. Therefore, many research works are targeted to develop drive cycles using recorded real world driving tests (complex transient) as well as steady state (cruise) conditions encountered in road driving. Such drive cycles have not been developed for city of Tehran yet,

and at the moment, the European ECE standard cycles are used for simulation and test. However, it has been shown [Kuhler and Karstens 1978] that the European "ECE" driving cycle has lack of ability to represent the actual driving conditions.

In this paper, the development of vehicle drive cycles are described, and based on the data collected from in use cars, a car drive cycle has been developed for city of Tehran. Development of emissions standard that can be the next step of this research is very useful for Tehran that has serious problem of air quality in many days of a year.

## AVAILABLE DRIVE CYCLES

Several drive cycles have been developed in different countries to represent the driving conditions. The most important cycles are developed for the United States, European community and Japan. In the United States, the "FTP-75" is a transient test cycle used for emission certification testing of cars and light duty trucks. The "SC03" Supplemental Federal Test Procedure (SFTP) has been introduced to represent the engine load and emissions associated with the use of air conditioning units in vehicles certified over the FTP-75 test cycle. The "US06" Supplemental Federal Test Procedure was also developed to address the shortcomings with the FTP-75 test cycle in the representation of aggressive, high speed and/or high acceleration driving behavior, rapid speed fluctuations, and driving behavior following startup. The "IM-240" test is a chassis dynamometer schedule used for emission testing of in-use light duty vehicles in inspection & maintenance programs implemented in a number of states. The "LA92" is a dynamometer driving schedule for light-duty vehicles developed by the California Air Resources Board. It is a more aggressive driving cycle than the federal FTP-75 characterizing by higher speed, higher acceleration, fewer stops per mile, and less idle time.

In European countries, the ECE+EUDC test cycle is performed on a chassis dynamometer. The cycle is used for emission certification of light duty vehicles in Europe. It is also known as the MVEG-A cycle.

In Japan the 10-15 mode cycle is currently used for emission certification and fuel economy for light duty vehicles and the 13-mode cycle for the testing of heavy duty engines.

## DEVELOPMENT OF CAR DRIVE CYCLE

To develop a drive cycle, the first step is to measure and record the real driving behaviors. Subsequently, the recorded data should be analyzed in order to obtain a representative cycle from real condition. Due to essentially different driving conditions, data should be classified in different categories based on traffic conditions. Statistical calculations will then be performed on the representative data, and an individual cycle for each traffic condition will be obtained. The final cycle is a sequence of these individual cycles. These steps have been described in the following.

### Speed Measurement and Recording Methods

The primary need for this work is to measure and record vehicles speed. The methods of measurement can be divided into two groups: 1) Using vehicles facilities to measure speed. We know that all vehicles have speed measurement system. The data from this system may be recorded and used for analysis. However, the vehicle owners do not like to change their vehicle system. Therefore, this method was rejected. 2) Using some additional equipment to measure the vehicle speed. For this method, it was required to design and install a device on all vehicles so that it could give a peace of mind to vehicles owners. For this purpose, an auxiliary wheel was designed as shown in figure 3. This wheel rotates together with the vehicle wheel. This wheel has a cam shaped part so that it makes a pin to reciprocate and open and close an electromechanical switch. As the wheel is rotating, a pulse chain is generated. By measuring the frequency of the wheel rotation, the speed of vehicles can be computed.

The electronic part of the device is an electronic network that works together with a notebook computer as a data logger. By this PC Based system, the output of the network is connected to the notebook parallel port where the frequency of pulse signal can be converted to vehicle speed.

### Analysis of The Recorded Data

Development a drive cycle is based on "microtrips". Microtrip is an excursion between two successive time points at which the vehicle is stopped. This part of motion consists of acceleration, cruise and deceleration modes. By convention, a period of rest is at the beginning of a microtrip.

To analyze the data, a computer program has been developed. This program calculates the vehicle speed and acceleration and save them as a text file. The program can also find and separate "microtrips" and the required parameters including the idle time, acceleration time, deceleration time and cruise time. This program also calculates the average and maximum speeds, acceleration and deceleration, and the number of "microtrips" for each car trip. At the beginning of recording, the Id. number of test, path of trip, vehicle brand and vehicle registration number are inserted by

user. When program starts by user, the date and time will also be saved.

### Parameters Used for Data Analysis

To develop a drive cycle, the data recorded from real driving tests should be analyzed. As mentioned before, the analysis is based on microtrips. The parameters used for analyzing microtrips are:

- Average speed (km/hr),
- Idle time percentage (%).

### Classification of Traffic Conditions

Traffic condition varies from region to region in a city, and therefore, classification of traffic situation is necessary. The classification parameters are considered to be the average speed and the percentage of idle time for each "microtrip". In this research work, to categorize the traffic condition, the idle time vs. average speed distribution chart shown in figure 4 is used. The 4 different traffic conditions are then defined as follows:

1. *Congested Urban Condition*: for central business district flow with low driving speed and frequently stops with the average speed less than 10 km/hr and a wide range of low to high idle time.
2. *Urban Condition*: for non-free flows with moderate and low idle time and average speed between 10 to 25 km/hr.
3. *Extra Urban Condition*: for relatively free flows with low idle time and average speed between 25 to 40 km/hr.
4. *Highways*: for completely free flows with very low idle time and average speed more than 40 km/hr.

Based on the classification above, the microtrips that fall outside the homographic range of idle time are omitted.

By applying these thresholds, the classification used for traffic conditions may be illustrated as shown in table 1.

Table 1: Classification of Traffic Conditions

	Congested	Urban	Extra Urban	Highway
Average Speed (km/hr)	≤10	10-25	25-40	>40
Idle Time (%)	0-100	<60	<24	<13

### Duration of Each Traffic Condition in Final Cycle

To obtain the duration of each category of traffic condition in the final cycle, the proportion of each category in the whole recorded data is used. In other words, it is equal to the duration of each category of traffic condition in the final cycle divided by the duration of the overall cycle as follows:

$$t_i = \frac{t_{drivecycle}}{t_{Overall}} \sum_{j=1}^{n_i} t_{i,j} \quad (1)$$

where:

$t_i$  is duration of category number  $i$  ( $i = 1, 2, 3, 4$ ) in the cycle,

$t_{drivecycle}$  is duration of the final drive cycle,

$t_{overall}$  is duration of all recorded data,

$t_{i,j}$  is the time of microtrip number  $j$  in category number  $i$ ,

$n_i$  is the total number of microtrips within category number  $i$ .

### Total Duration of Cycle

There should be an agreement between representative rate of a cycle and low cost test procedure on dynamometer, while the former implies long duration and the latter requires short duration cycle. It is more appropriate to develop a cycle with adaptation to reference cycle characteristics and with short duration suitable for dynamometer tests. Considering the existing drive cycles, it seems that a cycle with the duration of about 30 minutes can well represent the driving situation on a city as a reference cycle.

### Selection of Representative Microtrips

The final cycle consists of the representative microtrips selected from all existing data. In this study, we have defined the representative microtrip as follows:

"The representative microtrip is the one that minimizes difference between its parameters and those of the whole data in a certain category of traffic condition".

Based on the above definition, the following calculation is used for relative parameters:

$$\bar{v}_{rel,i} = \frac{\bar{v}_{mt,i}}{\bar{v}_{total}} \quad (2)$$

$$\%idle_{rel,i} = \frac{\%idle_{mt,i}}{\%idle_{total}} \quad (3)$$

where  $\bar{v}_{mt,i}$  is the average speed of microtrip number  $i$  and  $\%idle_{mt,i}$  is its idle time percentage. These parameters designated by index "total" for all the data in a certain category of traffic condition as well.

An ideal microtrip is selected so that its relative parameters are equal to 1 [Hann and Keller 2001]. However, such a microtrip is rarely exists. Therefore, the following indicator is defined for each microtrip:

$$N_i = \left| \bar{v}_{rel,i} - 1 \right| + \left| \%idle_{rel,i} - 1 \right| \quad (4)$$

where  $N_i$  is the indicator for microtrip number  $i$ .

Finally, the representative microtrips are selected so as to minimize the indicator  $N_i$ .

### SMOOTHING THE CYCLE

In order to remove the high frequency noise in speed-time series, a filter has been applied. Instead of non-optimal arithmetic mean, the filter of the following form has been used [Hann and Keller 2001]:

$$v_{smoothed}(t) = \frac{1}{h} \sum_{s=-h}^h K\left(\frac{s}{h}\right) v(t+s) \quad (5)$$

The function  $k(x)$  weights the measured speed just before and after the time  $t$  to be smoothed. In this study,  $h=4$  sec together with the so-called "biweight" smoothing kernel has been used [Hann and Keller 2001]:

$$K(x) = \begin{cases} \frac{h^2 - 1}{h^2} (1 - x^2)^2 & (x^2 < 1) \\ 0 & otherwise \end{cases} \quad (6)$$

### TEHRAN CAR DRIVE CYCLE

Based on the recorded data from the cars in the city of Tehran and using the above calculations, the cycles for all traffic conditions are obtained. The primary or non-smoothed and the smoothed cycles consist of congested urban, urban, extra urban and highway traffic conditions are shown in figures 5 and 6, respectively.

The characteristics of these cycles and differences between the non-smoothed and smoothed cycles have illustrated in table 2.

The smoothed cycle is presented as the final drive cycle of cars in Tehran. This cycle is named "TEH\_CAR" cycle.

### COMPARISON BETWEEN "TEH\_CAR" AND OTHER LIGHT VEHICLES CYCLES

In order to compare the driving patterns of TEH\_CAR and other light vehicles cycles in other countries, the characteristics of this cycle and some other important cycles are illustrated in table 3. In addition, the speed distribution charts are shown in figures 7 and 8.

The ECE and J10-15 mode cycles are constructed from only straight lines. They do not have maximum acceleration and deceleration near to real aggressive driving. However, the FTP cycle has higher maximum acceleration and deceleration and is comparable with the TEH\_CAR cycle. It is clear that the maximum acceleration and deceleration of TEH\_CAR cycle are greater than FTP cycle. This fact is due to driving pattern in city of Tehran that can affect the exhaust emissions and fuel consumption levels.

Table 2: Comparison between Non-smoothed and Smoothed Cycles Characteristics

	Time (Sec)	Dist (km)	V <sub>max</sub> (km/hr)	V <sub>avg</sub> (km/hr)	Max accel (m/s <sup>2</sup> )	Max decel (m/s <sup>2</sup> )	Avg accel (m/s <sup>2</sup> )	Avg decel (m/s <sup>2</sup> )	Idle time (%)	Accel time (%)	Decel time (%)
<b>Non-smoothed</b>	1955	15.8	83.6	29.1	3.31	-3.31	.71	-.76	12.8	36.6	34.1
<b>Smoothed</b>	1955	15.9	84.0	29.3	1.93	-2.21	.45	-.51	12.4	38.5	33.7
<b>Difference (%)</b>	0	.6	.5	.7	41.69	33.23	36.62	32.89	3.1	5.2	1.17

Table 3: Comparison of TEH\_CAR and other Light Vehicles Cycles

Idle Time (%)	Avg decel (m/s <sup>2</sup> )	Avg accel (m/s <sup>2</sup> )	Max decel (m/s <sup>2</sup> )	Max accel (m/s <sup>2</sup> )	Vavg (km/hr)	Vmax (km/hr)	Dist (km)	Time (sec)	
12.4	-.51	.45	-2.21	1.93	29.3	84.0	15.9	1955	<b>TEH_CAR</b>
18.92	-.57	.5	-1.48	1.48	31.51	91.25	11.99	1369	<b>FTP-72</b>
32.3	-.75	.63	-.83	1.06	18.35	50	.99	195	<b>ECE</b>
27.38	-.79	.54	-1.39	1.06	32.23	120	10.93	1220	<b>ECE+EUDC</b>
32.58	-.65	.57	-.83	.79	22.68	69.97	4.16	660	<b>J10-15 mode</b>

## SIMULATION RESULTS

The object in the car simulation can be defined as the exhaust gas emissions (g/km) and its fuel consumption (L/100 km). In this study, car simulation performed for different drive cycles to view the effect of changing driving patterns on mentioned parameters. The simulation results obtained for a typical car are shown in table 4. It can be seen as the average acceleration and deceleration of a cycle increase, the emissions and fuel consumption increases.

Table 4: Simulation Parameters for Different Drive Cycles

	HC	CO	NO <sub>x</sub>	Fuel Consumption
<b>TEH_CAR</b>	.937	6.157	1.121	6.1
<b>FTP72</b>	.979	6.785	1.446	6.0
<b>ECE</b>	4.01	15.520	1.780	8.1
<b>EUDC</b>	1.081	6.885	1.668	5.3
<b>ECE+EUDC</b>	1.004	6.752	1.436	6.1
<b>J10-15 mode</b>	1.562	8.585	1.532	7.0

## CONCLUSIONS

In this paper, development of car drive cycle was presented. Drive cycle is necessary for both simulation and test of vehicles gas emissions and fuel economy. Using the designed measurement system, the data were collected from the cars in Tehran. These data were analyzed using a developed computer program, and the Tehran car drive cycle (TEH\_CAR) was presented. Comparing TEH\_CAR with other cycles, it was shown that TEH\_CAR cycle has greater maximum acceleration and deceleration but smaller average acceleration and deceleration similar to FTP cycle, implying lower emissions and fuel consumption.

## REFERENCES

- André, M. 1994. "Statistical elements for the definition of a new European Evaporative Emissions Control Procedure; diurnal test conditions and Urban Driving Cycle." study for the European Commission, DGXI, INRETS report LEN9434, 49 p, Bron, France.
- Bata, R. and Y. Yacoub.2000. "Heavy Duty Testing Cycles: Survey and Comparison." *West Virginia University*.
- Brown, S. and C. Bryett. 2000. "In-service Emissions Performance- Drive Cycles" Volume 1, Final Report(May)
- Haan, P.D. and M. Keller. 2001. "Real-world driving cycles for emission measurement: ARTEMIS and Swiss cycles" Final Report (March).
- Hassel, M.A.D. and F.j. Weber. 1998. "The inspection of in-use cars in order to attain minimum emissions of pollutant and optimum energy efficiency" detailed report No. 2: Development of short driving cycles. INRETS reports (May).
- Kuhler, M. and D. Karstens. 1978. "Improved Driving Cycles for Testing Automotive Exhaust Emissions." Volkswagen AG, Germany. SAEP 780650.

## AUTHOR BIOGRAPHIES

**M. MONTAZERI-GH** has started teaching at IUST in 1986. He has then received his PhD in 1996 from Cranfield University in England. He is now the director of the Systems Simulation and Control Research Laboratory where he has established in 1997 at IUST.

**M. NAGHIZADEH** was born in 1980 in Fasa, Iran and graduated from IUST with B.S. degree in mechanical engineering. His e-mail address is: [moj\\_n\\_z@yahoo.com](mailto:moj_n_z@yahoo.com).

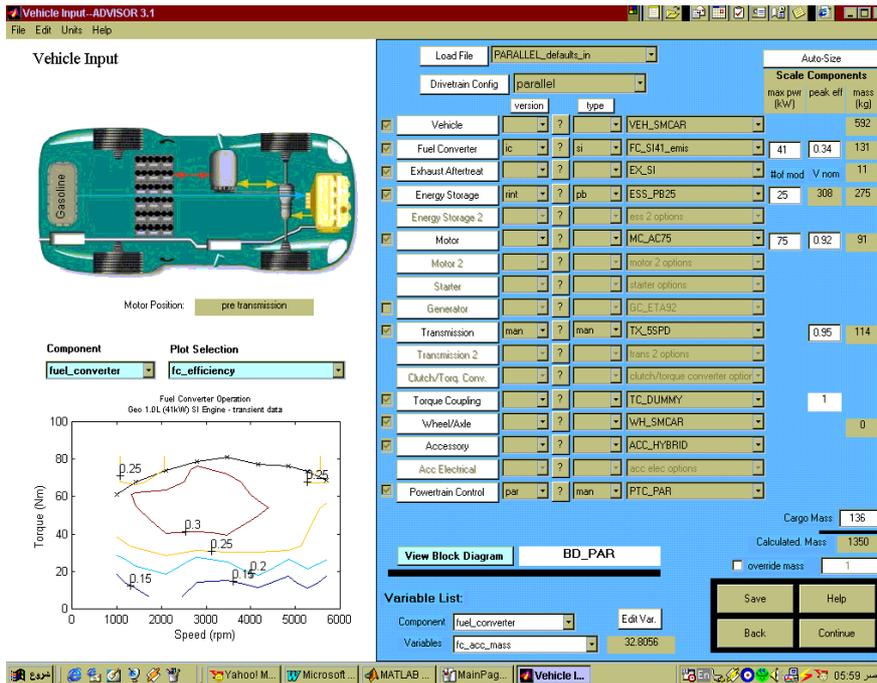


Figure 1: Vehicle Emissions and Fuel Economy Simulation Model

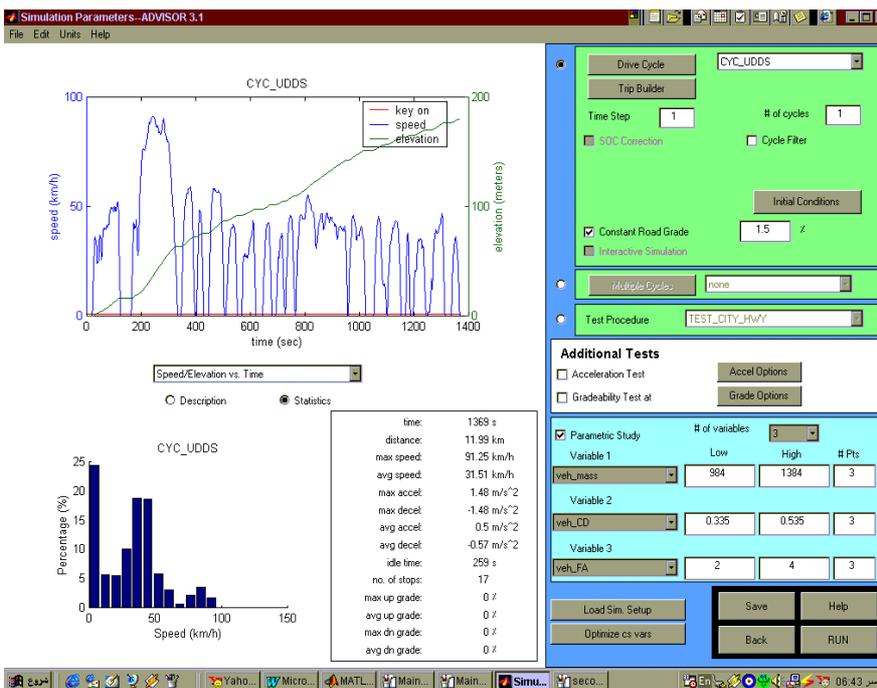


Figure 2: Drive Cycle Setup in Simulation Program



Figure 3: Auxiliary Wheel

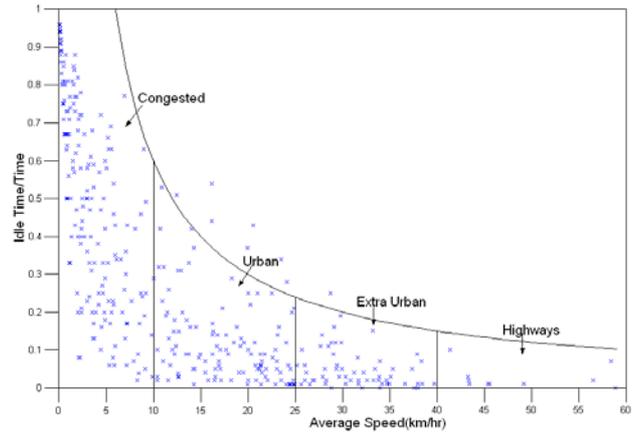


Figure 4: Idle Time vs. Average Speed Distribution for all Microtrips

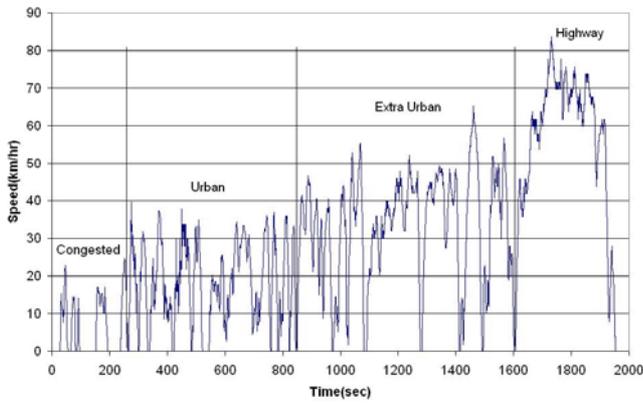


Figure 5: Tehran Cars Primary or Non-smoothed Cycle

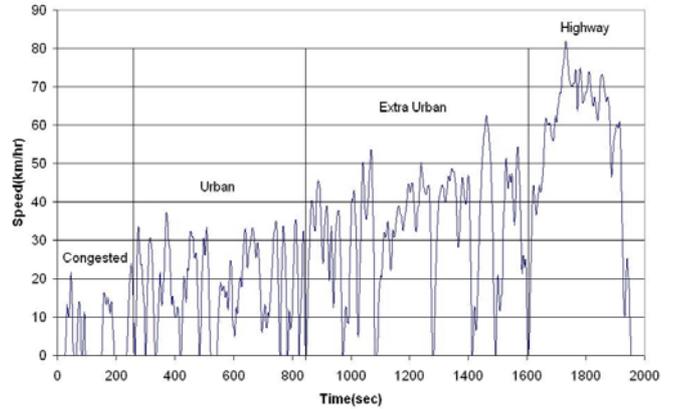


Figure 6: Cars Smoothed Cycle (TEH\_CAR Cycle)

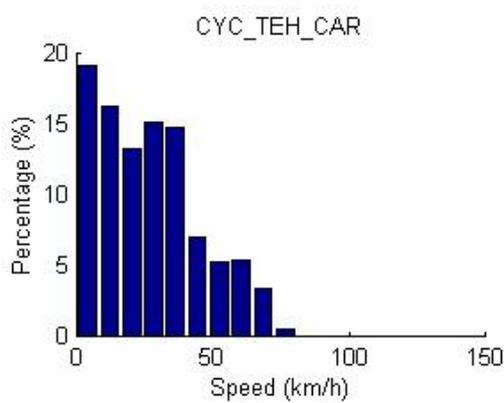


Figure 7: Speed Distribution Chart of TEH\_CAR Cycle

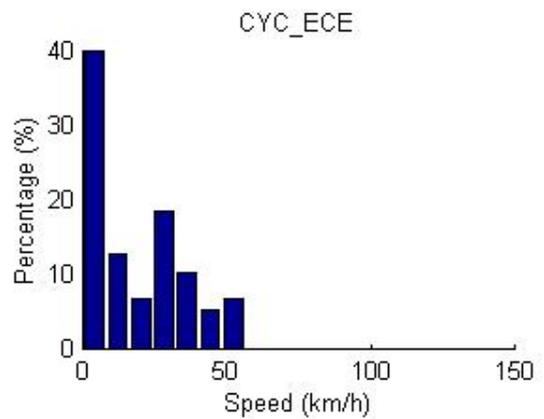


Figure 8: Speed Distribution Chart of ECE Cycle

# DISTRIBUTED E-SERVICES FOR ROAD CONTAINER TRANSPORT SIMULATION

Csaba Attila Boer  
Erasmus University Rotterdam  
Faculty of Economics  
Department of Computer Science  
P.O. Box 1738, 3000 DR Rotterdam  
The Netherlands  
acboer@few.eur.nl

Arjen de Waal  
TBA Nederland  
Vulcanusweg 259a, 2624 AV Delft  
The Netherlands  
arjen@tba.nl

Alexander Verbraeck  
Delft University of Technology  
Faculty of Technology, Policy and Management  
System Engineering Department  
P.O. Box 5105, 2600 GA Delft  
The Netherlands  
a.verbraeck@tbm.tudelft.nl

Bas van Eck and Jerry Seager  
ILLYAN  
Sarphatistraat 642, 1018 AV Amsterdam  
The Netherlands  
info@illyan.nl

## KEYWORDS

Planning and Scheduling, Distributed Simulation, Architecture, Container Handling, Port, e-Services.

## ABSTRACT

This paper describes a recently carried out project that aims to improve the handling process of trucks at new container terminals in ports. The simulation models of the truck companies and the ones of the container terminals might be separately developed by different parties applying different simulation packages. In this paper we give an approach for creating a distributed environment that supports the interoperability between these different models. Further, we introduce a planning and scheduling system that carries out the negotiation between the port terminal and truck companies to negotiate time slots for arriving at the terminal. This planning and scheduling system is included in the distributed environment as well and thereby becomes part of the simulated port and transport system.

## 1 INTRODUCTION

The FAMAS research program (de Hartog et al., 2001) intends to conceptualise and design new container terminals for the future port of Rotterdam. The involved organisations intend to apply the recent technical innovations and attempt to avoid the occurring problems on the present container terminals. One of the difficulties that terminals are faced with is the handling of the truck arrivals. The current situation sometimes results in large number of trucks waiting in excessively long queues, as they arrive more frequently than they are served. This situation especially arises in peak hours and in particular days when the number of trucks drastically in-

creases. Due to the limited number of serving cranes and the limited capacity of parking places these trucks are confronted with delays and a costly situation. This paper introduces a real planning system for scheduling the arriving time of the trucks at the terminal. The new planning system requires the truck companies to register the trucks at the terminal administration before delivering or picking up a container. The terminal administration together with the administration of the truck companies negotiate an arrival time that is acceptable for both sides. We deal with two main problems: *negotiation* from both sides and the *interoperability* of the systems of the negotiating parties. We approach the negotiation problem with an agent-based planning and scheduling system, which intends to negotiate based on some business rules provided by the truck and port authorities. To solve interoperability we use web-services architecture based on Extensible Markup Language (XML). Testing the effectiveness of the proposed scheduling system and its algorithms takes place in a distributed simulation environment where the trucking companies and the detailed port handling system are built as separate simulation models, and the planning system is available as a separate application that has to interface with the simulation models.

The paper is structured as follows. Section 2 gives a short introduction of the trucks scheduling problem and proposes a distributed modelling approach. Section 3 describes the conceptual distributed simulation model and additionally covers the interoperability between the simulation models. Section 4 introduces the participant models (federates) of the distributed system. Furthermore, the implementation aspects are given in this section as well. Section 5 contains some results of different experiments. Concluding remarks and directions for future research are provided in Section 6.

## 2 DETAILED PROGRAM DESCRIPTION

In order to deliver or pick up a container from the terminal in the proposed system, we distinguish different processes. First of all the truck companies that intend to deliver or pick up their containers to/from the terminal have to contact the terminal operator in order to make an appointment. The truck companies usually have a request, which is a desired arrival time at the terminal. Because of the fact that the container terminal has limited capacity (limited cranes, limited parking places, etc.) it might happen that many truck companies request the same time slot. To be able to guarantee truck companies a reasonable maximum turn around time (e.g. 30 minutes 95% reliable), the terminal will set a maximum value for the amount of trucks it grants a timeslot during the booking process. The reservation of a timeslot takes place in a multi-agent system. The agents negotiate on reservations on behalf of the truck companies and the terminal operator. There are some constraints, such as the currently available trucks, the currently available drivers, the time limits for the trucks and drivers (e.g. not driving at night), which the negotiator agents need to take into account. After obtaining a time slot (that describes the proposed arrival time at the container terminal) which is adequate for both parties the truck can drive to the container terminal. Unfortunately, the driving time cannot be precisely predicted due to traffic delays that might occur. Therefore the drivers sometimes have to count on some delays. Finally, the truck arrives at the terminal where it delivers or picks up the containers.

Based on the previous description we distinguish four separate processes (Figure 1). The truck requests can be either generated automatically by the truck simulation models or they can be generated in real time by a user. The request includes the generation of the desired arrival time, the type of container, etc. The planning and scheduling system attempts to provide the best time slot for each truck. Furthermore, the driving process might generate some expected delays due to traffic jams. The container handling operation is carried out in a detailed container terminal model.

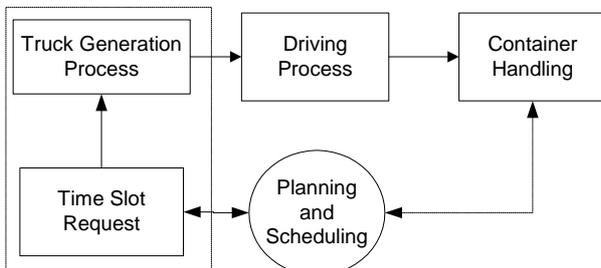


Figure 1. The main interoperability processes

The design and development of the whole complex model can be done monolithically (one big model using one package) or in a distributed way (well-distinguished models designed and developed in one or more packages). In the final stage of the project we would like to test the simulation models with the real planning and

scheduling system that uses high level negotiation mechanisms with recent technologies, such as intelligent agent based technology, web services, XML, etc.

The simulation model of the container terminal is highly detailed, which is essential in order to give a realistic handling time prediction, as the efficiency of the truck handling is very dependent on the other processes in the terminal, and there are many shared resources. This complex model already exists and can run without any external planning and scheduling system.

The second ingredient for the simulation is the driving model for the trucks that go from the trucking company to the port, with or without container(s). Different types of driving models exist (micro and macro traffic simulations) depending on the level of detail in which they were developed. Choosing a highly detailed model will lead to a more accurate result, but also requires very detailed input. Several simulation environments for road traffic were developed in the past years by several groups from different institutes.

Finally, a model component for the trucking company is needed. This model part generates a request, negotiates with the intermediary, and generates a truck to be sent to the container terminal using the road system. This generator is quite simple compared to the previous models.

Choosing a monolithical approach for designing and developing this complex system in one simulation model would result in a lot of work. Either the detailed port model, or the traffic models should be rewritten in order to integrate it with the other model parts. Furthermore, it would be extremely difficult to implement the agent based negotiation system in a simulation language – and actually again a waste of resources, as this system has already been built in the case we are studying. The monolithical approach that most simulation environments support as the only choice, always pose such problems in complex modelling tasks where different model parts from different background disciplines need to be integrated. In cases where other types of systems have to be included as well, the problem is even more aggravated.

In contrast with the monolithical approach we have used a distributed approach in which the possibility to interface with real planning and scheduling software is possible, and the existing developed simulation models and algorithms can be preserved.

## 3 THE CONCEPTUAL DISTRIBUTED MODEL

By, applying the distributed modelling approach the whole complex system can be designed and developed quite realistically. Different participants (organisations) can develop their own model without sharing their business logic. For example the truck companies concentrate on the generation process while the terminal operators focus on the container handling. They share only the relevant information between them, namely that which is necessary for the negotiation process. The planning and scheduling system is also unique, and is not consid-

ered as a part of a certain simulation model. It keeps the contact with the whole participant models and requires only internal information that is necessary for the negotiation process. A big advantage of this approach is thus, information hiding and furthermore, it increases the individual work because different modellers (from different organisations) can work in parallel (Taylor, et al. 2003).

The challenge we are faced with by applying distributed simulation modelling is that the individually designed and developed simulation models need to be coupled in order to form a consistent simulation federation. During the simulation run the different models interoperate with each other. For this reason *interoperability* must be achieved between the different simulation models (Fujimoto, 2000).

In order to achieve interoperability between the simulation models we apply a *distributed simulation architecture*. Several different distributed simulation architectures exist. One of them is the High Level Architecture (HLA), which is a standard architecture for modelling and simulation activities in the Department of Defense in the United States (Defense Modeling and Simulation Office 1996). Another distributed simulation architecture is the FAMAS Backbone (FAMAS MV2 Backbone Project, 2001), (Boer et al., 2002b). A comprehensive comparison between these two distributed simulation architectures can be found in (Boer et al., 2002c). For our purpose we chose the FAMAS Backbone Architecture because it is lightweight and it can be more easily interfaced with the simulation languages in which the already developed simulation models were written than HLA.

The FAMAS Simulation Backbone Architecture is represented by technical and functional components. Whereas the functional components represent the simulation models themselves, the technical components provide common tasks used by the functional components. The functional components can be simulation models, control programs or even real equipments.

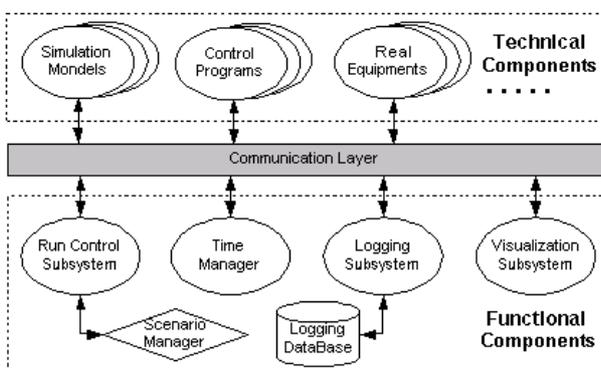


Figure 2: FAMAS Simulation Backbone Architecture

Figure 2 gives a clear picture of the separately defined functional and technical components. There are five well-defined subsystems, namely the Run Control Subsystem, the Backbone Time Manager Subsystem,

the Logging Subsystem and the Visualization Subsystem (Boer et al. 2002a), (Veeke et al. 2002). The functionalities of the technical subsystems are the followings: overall control of experiments, starting, stopping and periodically monitoring the simulation process (Run Control), synchronizing the simulation time among different simulation subsystems (Backbone Time Manager), collecting logging information from the distributed functional and technical components into a central database (Logging subsystem), providing separate or combined visualization views for the subsystems or the entire simulation (Visualization subsystem), completely defining a simulation run of a distributed model (Scenario Management).

Although most of the models use the FAMAS communication protocol for time synchronization and data exchange, other communication protocols are applied as well. The planning and scheduling tool communicates with the generator model through SOAP (Simple Object Access Protocol) by using the World Wide Web's Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchange (Schmelzer et al., 2002). Figure 3 depicts the conceptual distributed model of the whole system including the interoperability between different models.

## 4 FEDERATES OF THE DISTRIBUTED MODEL

### 4.1 Truck Generator Model

The truck generation process is based on requests for picking up and/or delivering containers. The generation of the requests and trucks is accomplished by the Truck Generator model. The Truck Generator can be either a simulation model that automatically generates requests or can be a real time model controlled by users (Figure 3). Requests are generated in the form of timeslots that refer either to a *desired departure* time (when the truck starts driving to the port) or a *desired arrival* time (when the truck should arrive at the port). It is not guaranteed that the truck can start the driving process at the desired departure time. Due to the limited capacity of the container terminal (limited parking places, limited cranes, etc.) it can accept only a certain amount of trucks at a given time. As there can be more truck organisations that reserve the same timeslot for arrival time, the container terminal might not be able to handle some of them at the desired time as it cannot accept more trucks at the same time. In order to avoid congestion and time waste caused by waiting at the terminal for handling, a negotiation process is taking place between the truck companies and the port authorities. The negotiation process provides a *scheduled* time slot regarding the departure and arrival time of the truck at the container terminal.

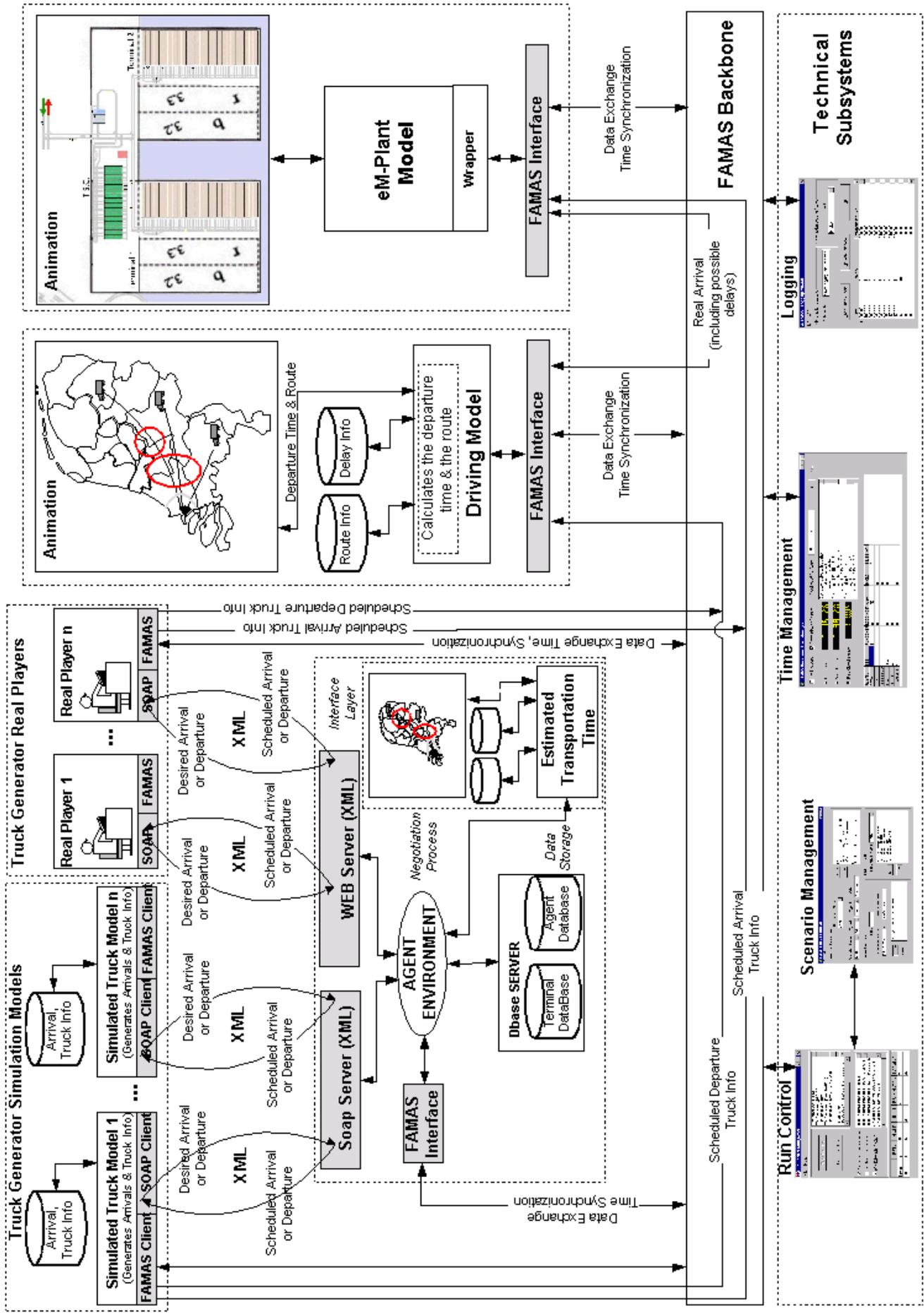


Figure 3. The Conceptual Distributed Model

The planning and scheduling model, which performs the negotiation, is using a model of the road traffic system in order to estimate the transportation time. This is needed in order to find out the approximate arriving time if the departure time is given or vice versa. Having the scheduled arrival and departure time, the scheduled departure time is provided to the road traffic model and the (approximated) scheduled arrival is provided to the container terminal model.

The simulation model of the truck generator stochastically generates the requests and creates the trucks based on earlier observed historical data (Figure 4). The Truck Generator uses special mechanisms in order to simulate the reservation of desired arrival or departure requested by truck companies. It allows the truck companies to make reservation for a timeslot for a certain time period in advance (in our case one week). The trucks that make a reservation for a time slot earlier get the desired time slot more easily compared to the companies that register late.

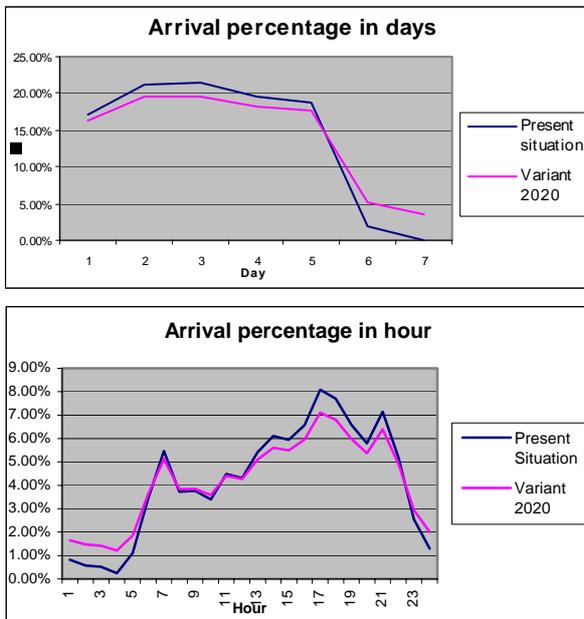


Figure 4. Arrival percentage in days and hour

We implement a general Truck Generator that can be instantiated in different ways for different truck companies. The company models use a configuration file with company specific data, such as nr. of trucks available, driving time, etc. Using several instances at the final test we can run several scenarios for different truck companies. We can analyse for example the different outcomes for earlier and later registered companies.

The truck generator simulation model is designed and developed in Java. The interoperability between the Truck Generator and the other models is achieved by the FAMAS Backbone and Simple Object Access Protocol (SOAP). The negotiation process between the Truck Generator and the Planning and Scheduling model is achieved through XML. Therefore the Truck Generator model is developed as a SOAP client for the planning and scheduling model.

## 4.2 Road Traffic Model

The aim of the road traffic model is to simulate the driving phase of the trucks from the companies to the port based on a given starting point and time provided by the truck generator, and external route and delay information provided by input data. Due to the fact that everything is modular in the distributed environment, we can use existing preserved models of the road traffic systems, although we need to solve the interfacing with other models. As we mentioned before we distinguish between micro and macro traffic simulation models depending on the level of detail at which they were developed. For a more accurate result, it is advisable to choose a highly detailed model, although it needs more input. We intend to carry out experiments with both micro and macro simulation models.

We use a road traffic model for two purposes. On the one hand one of its instances is used by the planning and scheduling model for estimation of the transportation time (here the animation is irrelevant), on the other hand another instance of this general model is used to simulate the driving of the trucks to the port. In this case the animation plays a crucial role, as it is indispensable for demonstration purposes. Both instances define the driving time stochastically, which highly depends on the exact day, hour and routes driven.

The input of the Road Traffic model contains the route and delay information stored in a database. Depending on the level of detail the external data might provide further information regarding the distance between two points (e.g. two intersections, two cities, etc.), the name of the road (e.g. A1), the maximum speed on that road, the earlier measured delays on this distance considering different days and hours, etc.

The Road Traffic model includes a demonstration by means of animation. The animation represents a map (e.g. map of Netherlands or even map of Europe) and visualizes the driving process of the trucks.

Data exchange and time synchronization with other models is solved through the FAMAS Backbone. The Truck Generator model provides to the road traffic model a scheduled departure time through this backbone. At that time a truck starts driving to the port. As we mentioned earlier this process is not deterministic, we can count on unexpected delays, or accidents, therefore the earlier provided scheduled arrival time to the port might be different to the real arrival. Although, the container terminal is informed about an approximately scheduled time when the truck arrives to the port, the driving model is responsible for sending a pre-arrival notice, which informs the real arrival of the truck. In this way the Automated Stacking Cranes (ASC) are able to start the preparing of the requested container for the truck (if it intends to pick up containers).

## 4.3 Container Terminal Model

The complex model of the container terminal already exists and can run without any planning and scheduling system. Furthermore, it has its own simple truck genera-

tor. However, we aim to improve the handling process of trucks at new container terminals by introducing a planning and scheduling system. Currently there are two independent container terminal models designed and developed for the Maasvlakte. The only aspect that they have in common is the simulation of the truck generator, which generates trucks according to the density per day and per hour as indicated in the description of section 4.1. Both of these concepts are designed and developed on a detailed level, which is essential to provide a realistic prediction.

### 4.3.1 Detailed Truck Handling Model at the Container Terminal

Regarding the truck processes at the container terminal we distinguish two concepts (van Til, 2003):

1. Compact Terminals (named as Concept 1)
2. Compact Terminals with a Central Gate and Truck Service Centre (named as Concept 5)

#### Concept 1: Compact Terminals

The schematic overview of the truck processes using this concept is depicted in figure 5.

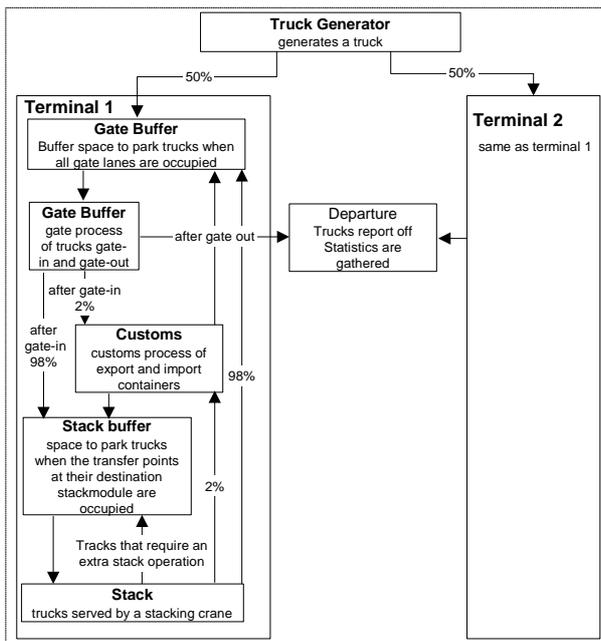


Figure 5. Schematic overview of concept 1

When a truck arrive at the terminal, the truck is assigned to the normal gate or to the Info Lack gate. The truck is positioned in the buffer in front of the correct gate. If one of the gate lanes is available, the truck continues immediately. After the treatment at the gate, the truck leaves towards the terminal. At customs, 2% of the trucks are being checked. These trucks are scanned after which they go to the substack of their destination. If no transfer points are available at the substack, the truck is positioned in the stack buffer, where it waits until one of the transfer points is available.

Trucks that are not checked at customs have passed the same routine before arriving at the substack. At the substack, the trucks are loaded or unloaded by the ASC. If the truck still has orders after the stack treatment, it will preferably stay in the same substack for the remaining orders. When this is not possible (e.g., because the truck has to pick up a container from another substack) the truck will go to another substack, if necessary, via the buffer. A favourable substack may be chosen for containers that are delivered by a truck. This is a substack where few orders are planned on the landside or a substack where the truck has to pass anyway to pick up a container. If all orders of the truck are carried out, the truck drives back to the gate to sign out. If necessary, the truck first takes place in the buffer prior to drive through the lane. After signing out the truck is removed from the model. All measured handling times are registered.



Figure 6. Layout terminals for concept 1

The simulation model of this concept is implemented in eM-Plant (eM-Plant official website, 2003), a commercially available simulation package. Figure 6 depicts the design of the compact terminal concept. The grey lines represent the roads and the grey blocks the buffers, placed in front of the trucks. The gate lanes are represented by the light blue colour, customs by the red and the sub stacks by the brown colour.

#### Concept 5: Compact Terminals with a Central Gate and a Truck Service Centre

The schematic overview of the truck processes using this concept is depicted in figure 7.



agent, this agent will implement enterprise specific strategies and will handle the message streams much faster than human operators will be able to (Leenaarts et al., 2003). Therefore an agent-based system has been selected for the timeslot negotiations at the container terminal. Requirements for this system were: Automated multi-channel communications, support for planning and scheduling, facilitating automated negotiations and eventually personalisable strategies.

In the agent-based system, every party (truck company or terminal operator) is represented by its own agent. A request with a desired arrival time from a truck company arrives at the system and is picked up by the representing truck operator agent. The request is passed on to the terminal operator agent, who will check availability of resources in the requested time slot. If there are sufficient resources in the requested slot, the terminal agent confirms the booking to the truck agent and updates the terminal database. If the terminal is fully booked at the requested time slot, the terminal agent will propose another slot, taking into account the bandwidth for negotiation that was sent with the request, the estimated transportation time based on the truck's departure information and information about the deep-sea vessel on which the container has to be loaded or from which the container has to be unloaded.

Because the generation of trucks with arrival times is completely automated during the simulation runs, there is no room for feedback from the truck company in case a requested timeslot is rejected. To show the functioning in a simulated "real" environment, a real player interface has been developed to submit time slot requests manually. Figure 10 below illustrates the real player applet interface.

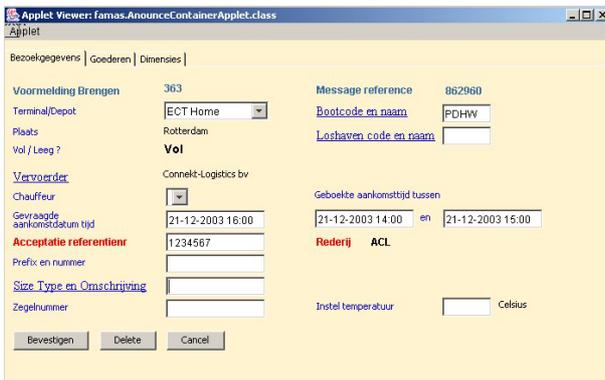


Figure 10. Real player applet interface

When a request for a specific timeslot has been rejected and a new timeslot has been proposed by the terminal, the truck company can accept the proposed time slot or try to get another slot which suits him better.

The system is based on the ILLYAN Agent Framework which is built in Java. It uses a SOAP Server and a Web Server for the communication with the other models and the interfaces. The messages are based on Extensible Markup Language (XML) for flexibility and interoperability.

The selected database management system (DBMS) is from the open source Firebird project. Because the standard JDBC 2.0 protocol is used to communicate with the databases, the system is effectively independent of the chosen DBMS. For visualisation purposes, a Java Swing GUI has been developed to display an overview of the requested and assigned timeslots. An example is shown below in Figure 11.

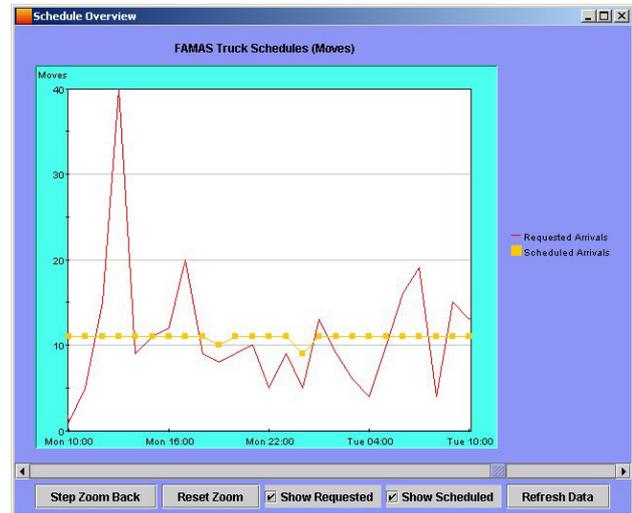


Figure 11. Visualization of assigned timeslots

## 5 EXPERIMENTS

### Evaluation of the distributed models

Several tests have been carried out to test the proper working of the individual models and of their interfaces to the other federates. Using the DLL that was described in section 4.3.2, it was very easy to connect the detailed eM-Plant terminal models to the FAMAS backbone. The truck generator models and road traffic models have been written in Java. As the FAMAS backbone architecture is based on plain TCP socket communication, which is well supported in Java, interfacing these models to the backbone also took place without problems. The ILLYAN Agent framework for planning and scheduling the timeslots is also a Java application, and could therefore be easily included in the federation as well. All the technical subsystems of the FAMAS backbone (section 3 / Figure 2) were already available from earlier projects. Interfacing tests showed that all information exchange and synchronisation took place as indicated. Efficiency tests still have to be carried out.

### Evaluation of the two container terminal concepts

Both terminal concepts discussed in section 4.3.1 are designed to serve 95% of the visiting trucks in 30 minutes. In concept 5 one standard stack module more is required to meet the performance requirements. On the other hand, an extra off-standard stack module is needed in concept 1. The fact that in concept 1 an extra stack module for off-standard containers is required, results in

a considerably lower efficiency of the present off-standard stack modules than in concept 5. Concept 1 requires a few extra gate lanes: two for the standard trucks and one for the Info Lack trucks.

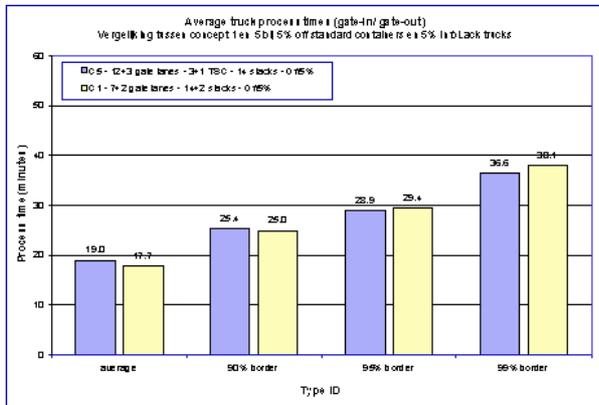


Figure 12. Comparison of handling times

Figure 12 shows a comparison between the handling times in concept 1 and 5. The average handling time of trucks in concept 1 is well over a minute shorter than in concept 5. The variation in handling times however, is smaller in concept 5. 99% of the trucks in concept 5 are treated within 36,6 minutes, whereas in concept 1, 99% is treated within 38,1 minutes. Concept 5 is more reliable concerning the handling times.

## 6 CONCLUSIONS

In this paper we introduce a distributed model that aims to improve the handling process of trucks at container terminals. In contrast to the already developed simulation models of the container terminal, which are monolithic, this approach integrates these models with other models to consider several additional aspects. In this sense it harmonizes the arrival time so as to be acceptable both for the container terminal and for the truck companies, and takes into account delays that might occur during the driving process of the truck to the terminal. Further it guarantees truck companies a reasonable maximum turn around time (e.g. 30 minutes 95% reliable).

This approach combines several independent models using a distributed modelling techniques. This entails a truck generator, a simulation model of the road system and the application of an agent based planning and scheduling model. The models of these systems are from different background disciplines and some of them already exist. In order to use these models together with the model of the container terminal we need to integrate them. In this paper we present a distributed environment that allows for the integration of these models. Although there is some collaborative work of the participants, this approach allows individual and parallel work, and the internal information in the models can remain hidden

for the other partners as they only share the relevant information for the interoperability process. This distributed environment can be extended and used by other participants as well.

Some final tests will be carried out to research the overall working of the federation, measuring the number of messages, the delays as a result of the distribution, and the efficiency of the model execution of the overall federated model.

## REFERENCES

- Boer, C.A., Y.A. Saanen, H.P.M. Veeke, and A. Verbraeck. 2002a. "Final Report Simulation Backbone FAMAS MV2. Project 0.2 Technical Design." Research report to Connekt, 34 pages, Delft, The Netherlands. (March).
- Boer, C. A., A. Verbraeck, and H.P.M. Veeke. 2002b. "Distributed Simulation of Complex Systems: Application in Container Handling." In *Proceedings of the 2002 European Simulation Interoperability Workshop* (Harrow, Middlesex, UK, June 24-26). SISO, pp. 134-142.
- Boer, C. A., and A. Verbraeck. 2002c. "Connecting High Level Distributed Simulation Architectures: An Approach for FAMAS-HLA Bridge." In *Proceedings of the 14<sup>th</sup> European Simulation Symposium* (Dresden, Germany, October 23-26), SCS, pp. 398-405.
- Defense Modeling and Simulation Office. 1996. *HLA Specification*. Washington DC, USA. Available online via <https://www.dmsomil/public/transition/hla/>, [accessed June 27, 2002].
- de Hartog, A. H., et al. 2001. *International state-of-the-art in container logistics and performance requirements for mega hubs. A vision for container logistics in the port of Rotterdam*. Connekt, Delft, The Netherlands.
- eM-Plant official website. 2003. Technomatics Technology Ltd. Available online via [www.emplant.de/simulation.html](http://www.emplant.de/simulation.html) [accessed June 27, 2003].
- FAMAS MV2 Backbone Project. 2001. Research Program FAMAS Maasvlakte II Project 0.2 - Simulation Backbone. Delft, The Netherlands. Available online via <http://www.famas.tudelft.nl> [accessed September 24, 2002].
- Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*. John Wiley & Sons, Inc., New York.
- Law A. M. and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 3d. ed. McGraw-Hill, New York.
- Leenaarts M., and M. Kentrop. 2003. "Distributed Planning of container terminal resources with agent technology". Application for BNAIC 2003, Amsterdam, The Netherlands.
- Schmelzer, R., T. Vandersypen, J. Bloomberg, M. Siddalingaiah, S. Hunting and M. Qualls. 2002. *XML and Web Services Unleashed*. Sams.
- Straßburger, S. 2001. *Distributed Simulation Based on the High Level Architecture in Civilian Application Domains*. Ghent : Society for Computer Simulation International, Magdeburg, Germany.
- Taylor, J. E. S., S. Robinson, and J. Ladbrook. 2003. "Towards Collaborative Simulation Modelling: Improving Human-to-Human Interaction through Groupware" In *European Simulation Multiconference* (Nottingham, UK, June 9-10), SCS.
- Van Til, K. P. 2003. "FAMAS MV2: Simulatie Logistieke Performance", TBA Nederland (Februari).

Veeke H.P.M., Y.A. Saanen, W. Rengelink, A. Verbraeck. 2002. "Final Report Simulation Backbone FAMAS MV2. Project 0.2 Functional Design." Research report to Connekt, 20 pages, Delft, The Netherlands. (April).

## AUTHOR BIOGRAPHIES

**CSABA ATTILA BOER** is a Ph.D. student at the Department of Computer Science of the Faculty of Economics at Erasmus University Rotterdam, The Netherlands. He received his M.Sc. degree in Computer Science at the Babes Bolyai University, Cluj Napoca, Romania. Since April 2001 he has been involved in the FAMAS MV2 Simulation Backbone project and since January 2003 in FAMAS MV2 Road Container Handling project. His research focuses on multi-level distributed simulation of complex systems. His email address is [<acboer@few.eur.nl>](mailto:acboer@few.eur.nl).

**ALEXANDER VERBRAECK** is an associate professor in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology, and part-time full research professor in supply chain management at the R.H. Smith School of Business of the University of Maryland. He is a specialist in discrete event simulation, both for real-time analysis and control of complex transportation systems and for modelling business systems. His current research focus is on the development of generic libraries

of distributed object oriented simulation building blocks. His email address is: [<a.verbraeck@tbm.tudelft.nl>](mailto:a.verbraeck@tbm.tudelft.nl).

**ARJEN DE WAAL** is a consultant at TBA Nederland, a company that specializes in simulation of complex logistic systems. Arjen graduated his study Operations Research and Management at the University of Amsterdam in 2000. He is an expert in discrete event simulation. His email address is [<arjen@tba.nl>](mailto:arjen@tba.nl).

**BAS VAN ECK** is a consultant for ILLYAN in Amsterdam, The Netherlands. He received his MSc in Chemical Engineering at the University of Amsterdam. After a career in process automation he co-founded ILLYAN and now focuses on e-business solutions in logistics. Since September 2001 he has been involved in the FAMAS MV2 Road Container Handling project. His email address is [<bvaneck@illyan.nl>](mailto:bvaneck@illyan.nl)

**JERRY SEAGER** is a consultant for ILLYAN in Amsterdam, The Netherlands. He received his BSc Upper Second Honours in Mathematics at Heriot-Watt University, Edinburgh, Scotland and his MSc in Advanced Software Engineering at Sheffield University, England. He joined ILLYAN in 2000 after a career in aircraft simulation in the UK. He has been involved in the FAMAS MV2 Road Container Handling project since January 2003. His email address is [<jseager@illyan.nl>](mailto:jseager@illyan.nl).

# ALLOCATION OF SHIPS IN A PORT SIMULATION

Eelco van Asperen  
Rommert Dekker  
Mark Polman  
Henk de Swaan Arons  
PO Box 1738  
Erasmus University Rotterdam  
Faculty of Economics and Business  
3000 DR Rotterdam, THE NETHERLANDS

## KEYWORDS

Port Simulation, Discrete-Event Simulation, Arrival Processes, Port Logistics.

## ABSTRACT

The limited jetty capacity of ports causes costly ship delays. This is a particular concern for large ocean-going vessels. Terminal operators attempt to reduce ship delays both in number and duration but have to take the number of jetties and their functionality and layout as a given. In such a setting, the arrival process of ships determines the delays in the loading and unloading process. Ships can arrive according to a schedule, for example based on stock levels or regular intervals, unscheduled, or even uncontrolled which is the case in a Poisson process. Priority rules in the processing of ships further impact the efficiency, both for stock-controlled and equidistant arrivals. Based on data from a real case study, this paper describes a number of simulation experiments to assess the impact of the arrival process on ship delays and, and to show the beneficial effect of the application of priority rules on the efficiency of loading and unloading.

## 1. INTRODUCTION

Little has been published on the simulation of port facilities, apart from some very scattered material. There is a nice book edited by Van Nunen and Verspui (Nunen and Verspui 1999) on simulation and logistics in the port, but it is in Dutch only. We briefly recapitulate the literature review on jetty design from Dekker (Dekker 1999) in that volume. Well-known to insiders are the reports from (UNCTAD 1978) on the design of jetties. They report results from both queuing theory and simulation applied to the capacity of jetties. The reports are however difficult to obtain and they give yardsticks for simple cases only. The other papers more or less describe that they have done a simulation study, without trying to generalize their results. We like to mention (Philips 1976) and (Andrews et al. 1996) who describe the planning of a crude-oil terminal, (Baunach et al. 1985), who deal with a coal terminal, (Heyden and Ottjes 1985), (Ottjes et al. 1992), and (Ottjes et al. 1994), who deal with the set-up of the simulation programs for terminals. None of these papers, however, deals explicitly with the arrival process.

In this paper, we focus on the analysis of ship waiting statistics and stock fluctuations under different arrival processes using a simulation model which is fed with data (types and number of ships handled per year) from a confidential case study in the Port of Rotterdam. The case study concerns a jetty and accompanying tank farm facilities belonging to a new chemical plant in the Port of Rotterdam. Both the supply of raw materials and the export of finished products occur through ships loading and unloading at the jetty. Since disruptions in the plant's production process are very expensive, buffer stock is needed to allow for variations in ship arrivals and overseas exports through large ships. We consider three types of arrival processes. The first type are the so-called stock-controlled arrivals, i.e., ship arrivals are scheduled in such a way, that a base stock level is maintained in the tanks. The second type of arrival process is based on equidistant arrivals in time of ships carrying the same product type. The last type of arrival process is an uncontrolled process, derived from a Poisson process.

Within each arrival process type a further distinction can be made between prioritized and non-prioritized queues in front of the jetty's mooring points. In this paper the various arrival processes will be compared with and without the application of priority rules. In the simulation model, some details concerning the diversity of ships and their numbers have been omitted. Also, details concerning tank operation, tank farm layout, and inland transport have been abstracted from. Still, the resulting model is general enough to draw conclusions applicable to many jetty simulation studies.

Section 2 briefly describes the model of the loading and unloading process. The various arrival processes are discussed in more detail in Section 3 with a focus on the application of priority rules in processing ships. The implementation model is the subject of Section 4 and the experiments carried out with it and their results are discussed in Section 5. The conclusions are presented in Section 6.

## 2. THE MODEL

A detailed description of the model can be found in (Asperen et al. 2003b). The model comprises the arrivals in time of ships, a jetty with a number of mooring points, storage tanks and a factory.

*The Jetty.* This is the loading and unloading facility with a number of mooring points. In this case there are four mooring points (mooring point 1 to 4) in a T-shaped layout. They differ in a number of aspects such as the length of the ships they can handle and the materials (raw materials A or B, and finished products C or D) they can load and/or unload.

*Raw Materials, Finished Products, Tanks and Stocks.* After being unloaded, raw materials are stored in tanks A and B, from where they are withdrawn by the factory. Finished products are transferred to tanks C and D, to be loaded into ships. Tanks can be used for only one type of raw material or finished product. In reality, there are several restrictions that affect actual tank operations, e.g. no simultaneous pumping and running into and out of a tank. We ignore these restrictions, because they do not affect the comparison between the arrival processes. The same holds for stocks; for simplicity we allow the stocks to take on any value (including negative values), and neglect ship delays because of stock outs or lack of ullage (available tank space).

*Ships.* There are ocean-going vessels, short-sea shipping vessels and inland barges which unload raw materials or load finished products. Each ship has properties relevant for the model such as size (tonnage), length (a distinction between long or short suffices), product (each ship handles just one specific type of cargo) and the (un)loading time. When a ship has arrived in the port, a suitable mooring point is selected according to specified rules, which are discussed below.

### 3. THE ARRIVAL PROCESS

In many simulation studies, the assumption is made that arrivals in client-oriented processes cannot be controlled. Consequently, simulation languages and environments tend to offer Poisson as a first-choice option for the specification of arrival processes. As mentioned above, this paper considers three scenarios to capture the ship arrival process.

#### Types Of Arrivals

*Stock-controlled arrivals.* These types of arrivals aim at maintaining a target base stock level in the tanks. For the loading process, this implies that the arrival time of the next ship is planned to coincide with the moment that, through production, there is sufficient stock in the tank to load the ship without dropping below base stock level. In this calculation, the parameters are the loading time of the present ship, the cargo capacity and loading time of the next ship, and the production capacity of the factory. Setting the appropriate base stock level for a tank involves an estimation of the tendency of ships to arrive ahead of schedule, this being the only threat to maintaining base stock level. For the unloading process, maintaining base stock levels in the raw materials tanks is achieved by planning the next ship's arrival to coincide with the moment that, through extraction of raw material during production, base stock level will be reached. In this calculation, the parameters are the cargo

capacity of the present ship, and the rate at which the factory extracts material from the tank. Here, the danger of stock dropping below base stock level comes from late arrivals (or from ships unable to instantly find an unoccupied mooring point).

*Equidistant arrivals.* With equidistant arrivals, arrivals of ships within the same ship type are assumed to be evenly spread over the year. For example, per year, 12 vessels carrying 6000 ton of product B arrive (see Table 1). With equidistant arrivals, this means a 1-month inter-arrival period between such ships.

*Uncontrolled arrivals.* The third arrival process considered in this paper is an uncontrolled process: within each cargo type, ships arrive uniformly distributed over the year. This process is obtained by specifying the number of arrivals in a Poisson process. See (Asperen et al. 2003b) for details.

The stock-controlled and equidistant arrival processes actually yield a series of *expected times of arrival* (ETAs). However, in reality ships seldom meet this schedule. For this reason disturbances to the ETAs are generated, modeling early and late arrivals resulting in the actual time of arrival (ATA) of each ship. See (Asperen et al. 2003b) for more details on these disturbances.

#### Ship Types And Arrival Rates

In order to be able to compare model outcomes over multiple years and among multiple arrival processes, the annual total number of arriving ships of each type is fixed, and identical for stock-controlled, equidistant, and uncontrolled arrivals. Table 1 shows which ship types are distinguished, and how many arrive per year. For example, every year, a total of 14 short vessels arrive each carrying 4000 tons of product B, with a loading time of 26 hours (for the meaning of the priority column, see below).

For each product/cargo type, the number of ships carrying it is chosen so that the total amount of cargo transported matches the factory's capacity. For instance, per year, the factory processes 1,070,000 tons of raw material A. Therefore, the total cargo capacity of ships carrying product A into the port needs to be 1,070,000 tons, which can be verified from the table.

This implies that among simulation runs, only the mutual order of arriving ships and their interarrival times are variable. Thus comparisons regarding port efficiency among arrival processes are kept clean (i.e. devoid of other circumstantial factors such as random fluctuations in production.)

#### Priorities

In reality, the arrival time of a ship is known, sometimes days beforehand, to the plant. This information can be used in a mooring point allocation system based on priorities. The general idea is to incorporate all ships within an n-hour horizon into the choice of mooring point for an incoming ship, in order to reduce costs induced by waiting for available mooring points, given

Table 1: Ship Types, Properties, and Arrival Rates

Ship type	barge/vessel	Size (tons)	Length	Product	Loading time (hours)	Ships per year	Priority	Tons per year
1	barge	1,500	short	A	8	196	low	294,000
2	vessel	2,000	short	A	8	48	low	96,000
3	vessel	4,000	short	A	20	80	low	320,000
4	vessel	6,000	long	A	26	60	high	360,000
								1,070,000
5	barge	1,000	short	B	10	38	low	38,000
6	vessel	2,000	short	B	11	161	low	322,000
7	vessel	4,000	short	B	26	14	low	56,000
8	vessel	6,000	short	B	26	12	low	72,000
								488,000
9	barge	1,000	short	C	10	180	low	180,000
10	vessel	2,000	long	C	14	126	high	252,000
								432,000
11	barge	1,500	short	D	8	134	low	201,000
12	vessel	2,000	short	D	8	300	low	600,000
13	vessel	10,000	long	D	44	14	high	140,000
14	vessel	20,000	long	D	56	8	high	160,000
								1,101,000

the fact that for some ship types, waiting is more expensive than for others (e.g. dependent on the type of cargo, the capacity, or the crew size).

This general idea can be implemented in many ways.

In this paper, we use a simple priority scheme, with two priority classes (high and low), in which long ships get high priority, and short ones get low priority. The allocation of a mooring point to a ship can now proceed as follows. A high-priority ship entering the port is in principle assigned to a free mooring point suitable for its cargo type and length. If all suitable mooring points are occupied, the ship is placed in a queue in front of the mooring point with the smallest workload, or, in case of equal workloads, the shortest queue so far. Here, the workload of a mooring point at instant  $t$  is defined as the total time from  $t$  that the mooring point will be occupied by the ship currently using it, and the ships currently in the queue in front of it.

For low-priority ships, the situation is similar, apart from an additional condition. To explain this, let  $s$  be a low-priority ship, let  $t$  be the current time, let  $Wi(t)$  be the workload of mooring point  $i$  at time  $t$ , and let  $Di(s)$  be the time that ship  $s$  needs if serviced at mooring point  $i$ . Then mooring point  $i$  is considered reserved if a high-priority ship arriving within a 48-hour horizon will need mooring point  $i$  between  $t$  and  $t + Wi(t) + Di(s)$ . If this is the case,  $s$  is not assigned to  $i$ , or enqueued in front of  $i$ . Note, that the shorter mooring points at the jetty are never reserved by high-priority ships, since all high-priority ships are too long for these mooring points. Hence, a low-priority ship will always either be as-

signed to a mooring point directly or placed in a queue in front of one.

In the presentation of the results in Section 6, we will make a distinction between model outcomes with and without priority-based mooring point allocation, so that the impact of incorporating such allocation is clearly visible.

#### 4. THE IMPLEMENTATION MODEL

The implementation model is based on the model outlined in Section 3. A detailed description can be found in (Asperen et al. 2003b). However, for a better understanding of the present paper, a few highlights are presented here.

The simulation model has been implemented in Enterprise Dynamics, a simulation package for discrete-event simulation (Enterprise Dynamics 2003). The implementation model outlined in Figure 1, comprises various types of atoms, the Enterprise Dynamics equivalents of objects. Some of the atoms implement the simulation's logic, others hold the simulation data (tables), define the types of experiments or provide the desired output (e.g., graphs).

The figure shows the number of ships which have entered the port thus far (262). Nine ships are on their way to the jetty. The utilization of mooring points 1 through 4 up to now has been 61.3%, 47.1%, 63.1% and 72.8%, respectively. At present, all four are occupied. Queues 1, 2, and 3 are empty, whereas Queue 4 contains one waiting ship. The actual contents of tanks A through D

# Jetty Simulation 1.3

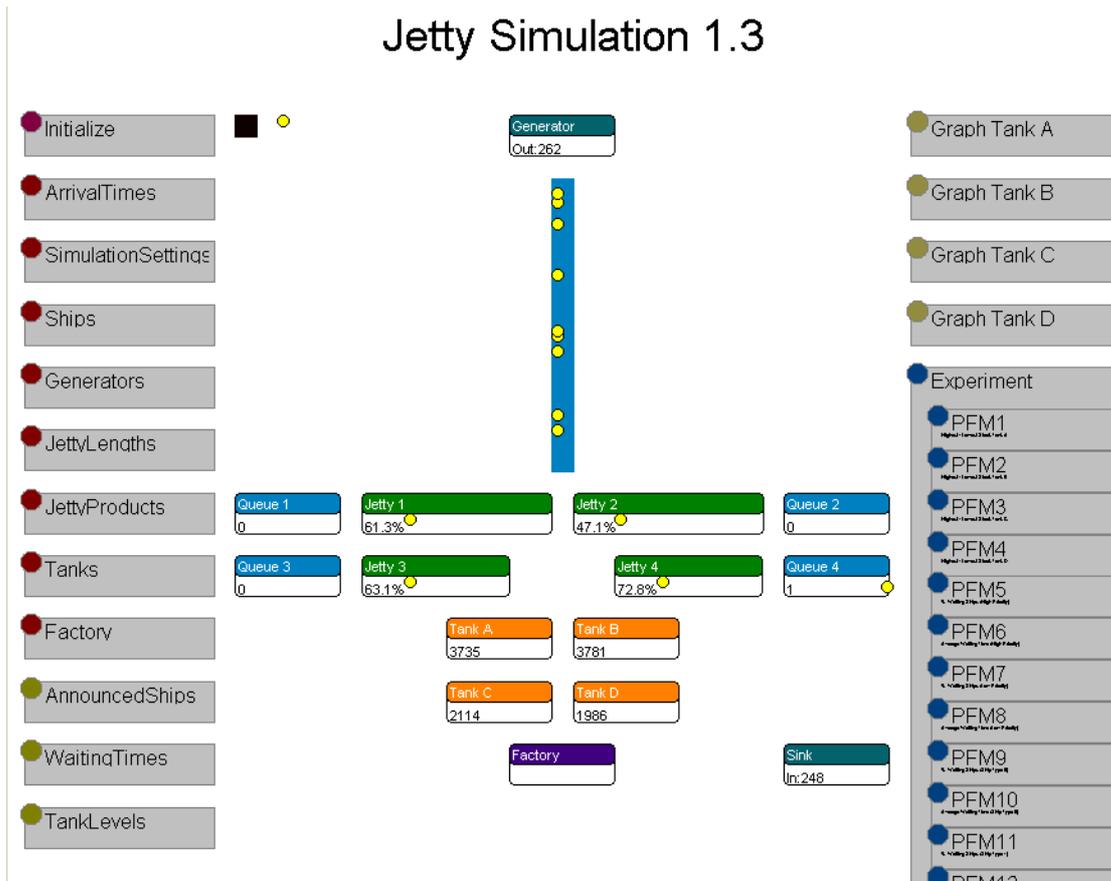


Figure 1: Implementation of the Simulation Model

are 3735, 3781, 2114 and 1986 tons, respectively. The total number of ships that have been processed is 248, which, added to the nine approaching ships and the 5 at the mooring points, matches the number of ships generated thus far.

## Logic

The Generator atom is responsible for generating ship arrivals. After arrival a ship proceeds along the atom ArrivalRoute (the vertical atom in the middle) to one of the four mooring points that suits its length and cargo type (see Section 3.4). If all suitable mooring points are occupied, the ship waits in one of the queues (Queue 1, 2, 3 or 4). Raw materials are unloaded and transferred to either Tank A or B, from which they are withdrawn by the Factory atom. The factory stores finished products in Tank C or D, from which they are withdrawn to be loaded into ships. After loading or unloading the ships leave the system. The tanks are assumed to have unlimited capacity and the possibility to contain negative stock. This simplification does not affect the simulation's objective.

## Data

The atoms on the left side represent tables providing data for the simulation process. All but the Initialize atom, which contains some code to be executed at the

beginning of each run, are actually tables. The top seven of these are filled from text files at the beginning of each run, and contain data concerning the arrival times (both ETA, ATA, including disturbances); some initializing data in the simulation settings; specific ship data such as type and size; the lengths of the mooring points and the products they can handle; the base stock levels of the various tanks; and the annual amounts of raw material processed and finished products produced by the plant.

The bottom three tables on the left are filled with data during simulation runs. They contain the data concerning the allocation of a ship to a mooring point, the waiting times statistics for all ship types and the stock level movements for each tank.

The Graph atoms on the right side (Graph Tank A to B) convert simulation results into the necessary graphs. The other atom (Experiment) on the right allows the user to define general preferences of a simulation experiment. In this case the Experiment atom also contains more than 30 PFM atoms (Performance Measure), each defining one output variable of interest. The atoms PFM1 till PFM4 provide the differences between the highest and lowest stock data of the tanks; PFM5 provides the percentage of the high priority waiting ships and PFM6 their average waiting times; PFM7 and PFM8 do the same for the low-priority ships. The re-

maining PFMs are used to collect similar data per individual ship type.

## 5. EXPERIMENTS AND RESULTS

The implementation of the model outlined in the previous section has been used to carry out experiments. While it is capable of generating results on a variety of topics, and on many levels of detail, we focus on the ones relevant to our objective: assessing the impact of using different arrival processes on stock levels and ships' waiting times. All in all, a total of six ten-year simulation runs are conducted: with stock-controlled arrivals, equidistant arrivals per ship type and uncontrolled arrivals, each with or without the use of priority rules.

Each run starts in a steady-state situation, with the tanks filled to base stock level. This eliminates the need for a warm-up period, which has consequently been omitted. Tables 2 and 3 show the relevant simulation outcomes. Table 2 contains the waiting statistics for ships for the three arrival processes without priority rules, each divided into separate results for high and low-priority ships (this distinction is made to facilitate a comparison with the results of simulation runs that *do* include a priority scheme, as described below.) Table 3 reports on the maximum and minimum stock levels reached for each of these arrival processes, both in raw material and finished product tanks. Table 4 shows the differences for each arrival process between using and not using priority rules for mooring point allocation.

## Waiting Times

From Table 2, it can be observed that the choice for a particular arrival process has significant impact on the number of waiting ships and the number of hours spent waiting by these ships. With uncontrolled arrivals both numbers are higher than those observed with equidistant and stock-controlled arrivals. This holds for both high and low-priority ships. Clearly, the lack of a mechanism to keep ships apart, whether it be equidistant or stock-controlled arrival planning, allows for clusters of ships arriving within a small time frame, causing queues.

Table 2 also reveals a noticeable difference between the outcomes of equidistant arrivals and stock-controlled arrivals. For both low- and high-priority ships, the stock-controlled arrival process 'outperforms' the equidistant arrival process. The explanation for this is manifold. For one, stock-controlled arrivals are more efficient overall since they tend to keep ships of identical cargo types apart, whereas equidistant arrivals keep ships of identical types apart. With multiple ship types per cargo type this is an advantage. However, the arrival rates of the individual ship types (which is something very particular to this simulation) have an impact as well. Consider, for example, the 126 type 10 vessels, and the 14 type 13 vessels from Table 1. If, with equidistant arrival times, the first ships of both types have identical expected times of arrival, every arrival of a ship of the latter type coincides with one of the former. The observed differences in waiting time statistics among arrival processes, and their causative factors, clearly demonstrate the need for careful arrival process modeling, which is this paper's primary objective. Obviously, arrival process modeling requires a careful look

Table 2: Ship statistics for the various arrival processes without priorities (means over a 10-year period)

		Ship Priority			
		High		Low	
		Mean	St. dev.	Mean	St. dev.
Percentage of ships that had to wait (%)					
	Stock-controlled	21.1	3.7	12.0	1.0
	Equidistant	34.7	1.8	23.5	0.8
	Uncontrolled	45.7	2.1	35.2	2.0
Average waiting time of ships that had to wait (hrs)					
	Stock-controlled	7.9	1.1	3.5	0.2
	Equidistant	9.5	0.6	6.2	0.2
	Uncontrolled	12.3	1.8	7.5	0.9

Table 3: Stock level ranges for the various arrival types without priorities (means in tons over a 10-year period)

	Tank							
	A		B		C		D	
	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.
Stock-controlled	6970	468	5890	294	3011	320	15982	578
Equidistant	10756	273	11245	312	3381	283	27474	574
Uncontrolled	74396	18333	48058	11789	32045	9112	89177	15112

Table 4: Ship statistics for the various arrival processes, priority rules vs. no priority rules (means over a 10 year period)

		Ship Priority			
		High		Low	
		No priority rules	Priority rules	No priority rules	Priority rules
Percentage of ships that had to wait (%)					
	Stock-controlled	21.1	8.5	12.0	14.2
	Equidistant	34.7	9.2	23.5	28.7
	Uncontrolled	45.7	18.3	35.2	40.5
Average waiting time of ships that had to wait (hrs)					
	Stock-controlled	7.9	10.0	3.5	3.8
	Equidistant	9.5	9.8	6.2	7.2
	Uncontrolled	12.3	14.6	7.5	9.4

at the real situation, involving expert input on many subjects. Only then are simulation results valid, and can they be used in corporate decision-making. Alternatively stated, providing only the numerical data from Table 1, and simply assuming an uncontrolled process, is insufficient, rendering any subsequent decision (for example on expensive alternative jetty layout to reduce waiting times) ill-founded.

#### Stock Levels

Table 3 shows 10-year stock level statistics in terms of the difference between minimum and maximum levels reached. As could be expected, stock fluctuations are smallest with stock-controlled arrivals, whereas uncontrolled arrivals allow for the largest. Also, with equidistant arrivals, considerable fluctuations are observed. It is clear that the choice of arrival process is an important factor in simulation outcomes. More information about the stock fluctuation patterns over time can be found in (Asperen et al. 2003b).

#### The Effect Of Using Priority Rules

In section 4.6 it was explained that priority rules are expected to reduce the waiting costs of high-priority ships. A simple priority scheme was considered with two priority classes (high and low), where long ships get high priority, and short ones low priority.

Table 4 shows the ship waiting statistics over a ten-year simulation period for each arrival process, both with and without (copied from Table 2) priority rules. Standard deviations have been omitted for brevity.

In all cases, applying priority rules indeed reduces the percentage of high-priority ships, while increasing the percentage of low-priority ships that have to wait. All waiting time means go up, for which there are, again, multiple causing factors. One seemingly obvious mechanism is that high-priority ships are now very rarely blocked from suitable mooring points by low-priority ships. Hence, if a high-priority ship has to wait, it is probably for another high-priority ship, which takes longer to (un)load, causing longer delays.

The question as to whether total waiting costs are reduced by applying priority rules, or to what extent, depends on how much more expensive an idle high-priority ship is over a low-priority ship. The tender of the original case study did not provide a cost function.

#### 6. CONCLUSIONS AND FURTHER RESEARCH

In (Asperen 2003a) it was already concluded that careful arrival process modeling is very important with respect to ship and stock statistics. Model outcomes over various arrival processes vary significantly, e.g. the uncontrolled process has by far the worst performance of the three processes discussed, both in terms of waiting times and in terms of the required storage capacity, whereas the stock-controlled process performs best overall.

In this paper the emphasis was put on how priority rules affect these results. It may be concluded that priority rules have a positive effect on the ship statistics. The percentage of the high-priority ships that had to wait was reduced with a factor of about 3, at the cost of low-priority ships. In the less frequent cases that high-priority ships had to wait, priority rules slightly pushed up average waiting times.

There are various directions in which future research is planned. First, the role of the jetty's layout needs to be explored, specifically the impact of limited length of the individual mooring points, and the restrictions on the availability of piping for specific products.

Also, the effects of using more sophisticated allocation strategies than a two-class priority scheme for assigning ships to mooring points, requires further study.

Finally, we intend to consider another arrival process, a hybrid one, with planned arrivals for the larger vessels and equidistant or uncontrolled arrivals for the smaller barges.

#### ACKNOWLEDGEMENTS

The authors would like to thank the students Nees Jan van Eck, Arthur Oink, Gerard Seedorf and Ludo Waltman for their solid implementation of the simulation model in Enterprise Dynamics and for carrying out

the experiments. They are also grateful to Stef Kurstjens for his work in the original case study.

## WEB

More information on this study can be found on the website: <http://www.few.eur.nl/few/research/eurfew21/m&s/article/jetty/>.

The website contains graphs showing the levels of all tanks over a one year period and a video that shows a simulation run.

## REFERENCES

- Andrews, S., F.H. Murphy, X.P. Wang, and S. Welch. 1996. "Modeling crude oil lightering in Delaware Bay." *Interfaces* 26, No. 6, 68-78.
- Asperen, E. van, R. Dekker, M. Polman and H. de Swaan Arons. 2003a. "Modeling Ship Arrivals in Ports." In *Proceedings of the Winter Simulation Conference 2003*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice (Eds.). New Orleans. Accepted for publication.
- Asperen, E. van, R. Dekker, M. Polman, H. de Swaan Arons, and L. Waltman. 2003b. "Arrival Processes for Vessels in a Port Simulation." ERIM Report Series Research in Management, ERS-2003-067-LIS.
- Baunach, G.R., E.S. Wibberley, and B.R. Wood. 1985. "Simulation of a coal transshipment terminal: Batam Island, Indonesia." *Math. Comp. Simul.* 27, 115-120.
- Dekker R. 1999. "Simulation of jetty and storage activities for oil and chemicals." In *SimLog, Simulation and logistics in the harbor*, J. Van Nunen and L. Verspui (Eds.). Eburon Delft, Netherlands, 105-116 (in Dutch).
- Enterprise Dynamics. 2003. *Documentation material*. Incontrol Enterprise Dynamics. Contact <http://www.enterprisedynamics.com>.
- Heyden, W.P.A. van der, and J.A. Ottjes. 1985. "A decision support system for the planning of the workload on a grain terminal." *Decision Support Systems* 1, 293-297.
- Nunen, J. van, and L. Verspui, 1999. *SimLog, Simulatie en logistiek rond de haven*, (translation: *SimLog, Simulation and logistics in the harbor*). Eburon Delft, Netherlands (in Dutch).
- Philips, O.O. 1976. *Optimization models for a crude oil storage export system*. Ph.D. thesis. Penn. State University.
- Ottjes, J.A. 1992. *Modelvorming en simulatie van logistieke systemen*. Manual i76C, Delft University (in Dutch).
- Ottjes, J.A., S. Hengst and W.H. Tutuarima 1994. "A simulation model of a sailing container terminal service in the port of Rotterdam." In *Proceedings of the 1994 Conference on Modelling and Simulation*, Guasch and Huber (Eds.), 876-880.
- UNCTAD. 1978. *Port development: a handbook for planners in developing countries*. New York UN, TD/B/C.U/175, chapter II: the break-bulk berth group, 108-128.

## AUTHOR BIOGRAPHIES

**EELCO VAN ASPEREN** graduated in Business Computer Science at Erasmus University Rotterdam in 1993. From 1991 to 2000 he was a member of the IT support group at the Erasmus University Rotterdam, for the department of Computer Science and from 1995 for the Faculty of Economics, focusing on the design and implementation of large scale computer facilities. Since January 2000 he is an assistant professor at the Depart-

ment of Computer Science of the Faculty of Economics at Erasmus University Rotterdam. His research focuses on simulation with applications in logistics. You can reach him by e-mail at [vanasperen@few.eur.nl](mailto:vanasperen@few.eur.nl) and his web address is <http://www.few.eur.nl/few/people/vanasperen/>.

**ROMMERT DEKKER** is a full professor in operations research at the Econometric Institute of Erasmus University Rotterdam. He obtained his Ph.D. in operations research at the State University of Leiden, and his M.Sc. degree in industrial engineering from Twente University of Technology. He worked with Shell for seven years on reliability and refinery logistics. His current research interests are: maintenance and logistics (inventory control, spare parts, ports, containers and reverse logistics). He has applied simulation models in various logistical problems. His e-mail address is [rdekker@few.eur.nl](mailto:rdekker@few.eur.nl); his web address is <http://www.few.eur.nl/few/people/rdekker>.

**MARK POLMAN** graduated in Business Computer Science at Erasmus University Rotterdam in 1993. Since January 2000 he is an assistant professor at the Department of Computer Science of the Faculty of Economics at Erasmus University Rotterdam. Previous research areas include machine learning and communication modeling in distributed systems. His research focuses on discrete-event simulation. His e-mail address is [polman@few.eur.nl](mailto:polman@few.eur.nl); his web address is <http://www.few.eur.nl/few/people/polman>.

**HENK DE SWAAN ARONS** is an associate professor at the Department of Computer Science of the Faculty of Economics at Erasmus University Rotterdam. He graduated in Applied Mathematics at Delft University of Technology in 1972. In 1991 he obtained his Ph.D. degree in computer science at Delft University of Technology. The thesis was mainly concerned with the design, applicability and applications of expert system tools. His research focuses on discrete-event and continuous simulation, with the emphasis on economical applications. His e-mail address is [deswaanarons@few.eur.nl](mailto:deswaanarons@few.eur.nl); his web address is <http://www.few.eur.nl/few/people/deswaanarons/>.

# PLANNING THE RECONSTRUCTION OF A SHIPLIFT BY SIMULATION OF A STOCHASTIC PETRI NET MODEL

Matthias Becker

Thomas Bessey

Institute of Systems Engineering, University of Hannover

Welfengarten 1, 30167 Hannover, Germany

{xmb,tby}@sim.uni-hannover.de

## KEYWORDS

Shiplift, Case Study, Stochastic Petri Net

## ABSTRACT

In this case study, two alternatives for reconstruction of an existing shiplift are evaluated. At the moment, the shiplift consists of two long chambers. One chamber is to be rebuilt. Instead of rebuilding it in its original length, a shorter and cheaper chamber could also be built.

In this paper, a stochastic Petri net model of the shiplift is used to simulate the shiplift and to evaluate the two alternatives, taking performance, load and customer satisfaction into consideration.

The Petri net model has been chosen because Petri nets are a universal modeling language that allows a quick creation, validation and evaluation of models of arbitrary systems. Petri nets furthermore offer a graphical illustration/animation that is useful for the communication with the non-simulationists that are involved.

## INTRODUCTION

At the moment, the shiplift located in Lower Saxony, Germany consists of two parallel chambers, where each chamber has a length of 220 meters. One chamber will be too old for safe operation in approximately twelve years. The alternatives that have to be considered then are to renovate or completely rebuild this chamber. A complete rebuild offers more alternatives, either to rebuild it in its original size, or to build a shorter and thus cheaper chamber of 110 meters length. These three alternatives have to be considered under financial, environmental and political aspects which are out of scope here.

This work concentrates on the question whether the building of a shorter chamber will be able to cope with current and projected traffic.

In the next section, we give details about the shiplift. Then its stochastic Petri net model is explained. After that we describe the simulation experiments and

their results. We conclude by discussing advantages and drawbacks of our approach.

## THE SHIPLIFT

The shiplift consists of two parallel chambers that are operated independently. If a ship arrives at the shiplift and finds at least one chamber open, it enters the chamber and is brought to the other side. In case the ship finds both chambers closed, it has to wait. The operator of the shiplift decides whether to assign a chamber for the waiting ship or to wait for a ship on the other side that is known to arrive soon because of a radio announcement.

The number of ships fitting into one chamber depends on their length. There are eight classes ranging from 40 to 110 meters. We rearrange these classes into three classes:

- Small ships up to 50 meters make up 10.0 percent of the traffic.
- Very large ships with a length of 110 meters make up 24.6 percent.
- The rest of the traffic are middle class ships (65.4 percent).

The reason for this abstraction is that it is crucial to have the very large ships in one class, because only two of them fit into one long chamber, and only one would fit into a short chamber. The small ships are only a small share of the overall traffic and in most cases they still fit into a partly filled chamber. Two ships of medium size fit into a long chamber and only one into a short chamber. We will come back to this when describing the model.

The mean interarrival time between ships is 29.6 minutes from downstream as well as from upstream. This mean was calculated from the total number of ships of the last year and the sum of the periods that the shiplift has been operational. We had no data of exact arrival times. We only had the times of the ships as they entered a chamber, from three

days. Note that these entrance times are not the arrival times, since, while one ship waits for the chamber, another ship may arrive and enter the chamber concurrently. Thus the entrance times show a more 'batchy' pattern than the actual arrival times. Distribution fitting of the entrance times showed that a Poisson arrival process can be assumed.

The overall time needed for a ship to enter one chamber, close the gates, raise or lower the water level, open gates and leave the chamber is 28.0 minutes. The action of operating the gates and adjusting the water level can assumed to be deterministic.

From these numbers it can easily be concluded that utilization of the shiplift is relatively low for two long chambers. But the arrival process is not deterministic, thus queuing occurs despite the low utilization. And especially when substituting one of the long chambers by a shorter one, the question is what the average waiting time is and also what the probability for a non-acceptable waiting time/queue length is.

## THE PETRI NET MODEL

Since there is no special simulation software for this problem, we decided to use Generalized Stochastic Petri Nets (GSPN) as described e.g. in [1, 4]. GSPN are a universal modeling language that allows a quick creation, validation and performance evaluation of models of arbitrary systems. In our case, communication with non-simulationists has been necessary, thus a graphical representation and animation of the model was desirable. GSPN provide this graphical representation, that is easier to understand for non-specialists than e.g. the simulation code written in some programming language.

In GSPN, the system state is modeled by tokens in places (small filled dots inside circles), i.e. the marking. State changes are modeled by transitions (bars). If the state change needs some time, then a timed transition is used (unfilled bar), while for timeless state changes immediate transitions are used (filled bars). When an action occurs, transitions move as many tokens from and to places as indicated by the arcs connecting the places and the transitions. See e.g. [1, 4] for details of the dynamics of GSPN.

Figure 1 shows the GSPN model of the shiplift as constructed with the tool TimeNET [2], which enables simulation of the GSPN as well as performance analysis (based on its Markov chain) and qualitative analysis.

The transitions `arrival_ds` and `arrival_us` model the Poisson arrival process of ships from upstream and downstream, with a mean interarrival time of

29.6 minutes. Ships arrive at harbours denoted by the places `harbour_ds` and `harbour_us`. The marking of places `chamber_us`, `chamber_ds`, `chamber_ds2` and `chamber_us2` indicate whether the current water level in chamber one/two is adjusted either to the upstream or downstream level. The transitions `enterxy` test via inhibitor arcs, whether a chamber is in the correct position and also whether the chamber is empty. If there is only one ship to enter a chamber, then this ship enters. If more than one ship is waiting, then the transitions `enterxy` decide based on specific probabilities which are derived from the given distribution of shiplengths (as discussed earlier), how many ships enter the chamber. Note that in the Petri net used here, all ships are uniformly modeled as tokens without any information, so ships cannot be distinguished with respect to their lengths.

This approach introduces a certain level of abstraction of the model, which was not easy to understand for the engineers concerned with the reconstruction of the shiplift. Thus we also built a more detailed model with colored Petri nets [3] of the Renew type [5], where each ship has been modeled as a colored token carrying more detailed information like the length of the ship. The loading of the chambers has also been modeled in detail, that means that it was tested for each ship at the front of a queue, if it still fits into a chamber or not. This detailed model showed nearly the same results, but it is much more complicated and less intuitive. However it gave the engineers more confidence in Petri net models. It is out of the scope of this paper also to explain the colored Petri net model.

Once ships have entered a chamber, the filling or emptying of the chamber begins. This is modeled by the four deterministic transitions `fill_chamber`, `empty_chamber`, `fill_chamber2`, `empty_chamber2`, each taking 28.0 minutes. Chambers may only be operated if a token is present in place `trigger_chamber` / `trigger_chamber2`. Such token is generated when either ships entered the chamber, or waiting ships request a chamber if all two chambers have the wrong water level.

After the chamber's operation time, the ships are released into places `ds1`, `ds2`, `us1` and `us2`. The marking dependent arc weights ensure that always the proper number of ships is moved.

## RESULTS

- First we validated the model by simulating the current configuration with two long chambers. The mean number of ships in one chamber in our model is then 0.91 for chamber one and 0.88 for chamber two. If a ship needs to request

a chamber, then it will request only chamber one; this explains the slight asymmetry of these two values. The mean number of waiting ships (on both sides) sums to 1.09.

Both the mean number of ships in the chambers and the mean number of waiting ships correspond very well to the measured data at the real shiplift.

- Then we simulated the design alternative with one shorter chamber. The mean number of ships in the shorter chamber is then 0.86 and that in the longer chamber is 0.93.

The overall number of waiting ships has increased to 1.32.

(All simulations have been done with 98% confidence level and a maximal relative error of 5%.)

## CONCLUSION

In this case study, design alternatives for the layout of a shiplift have been evaluated. Despite the shiplift having a low utilization, queueing occurs due to the stochastic arrival process of ships. It showed that also the alternative layout with less capacity (i.e. one shorter chamber) would suffice and only minimally raise queue length and waiting time. Thus the decision probably will be determined by financial and political aspects.

The conclusion we draw from this case study is that it has been very convenient to use GSPN for this purpose. GSPN offer a universal formal modeling language making creation and debugging of the model faster and easier than e.g. using C-code. Furthermore, GSPN have a graphical representation and animation at no extra cost. In our case, these benefits proved to be very useful for communication with the non-simulationists.

## REFERENCES

- [1] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93 – 122, 1984.
- [2] R. German, C. Kelling, A. Zimmermann, and G. Hommel. TimeNET — A toolkit for evaluating non-Markovian stochastic Petri nets. *Performance Evaluation*, 24:69–87, 1995.
- [3] K. Jensen. *Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use, Vol. 1: Basic Concepts*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1992.
- [4] M. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalized Stochastic Petri Nets*. John Wiley and Sons, Chichester England, 1995.
- [5] Renew – the reference net workshop. Website at [www.renew.de](http://www.renew.de).

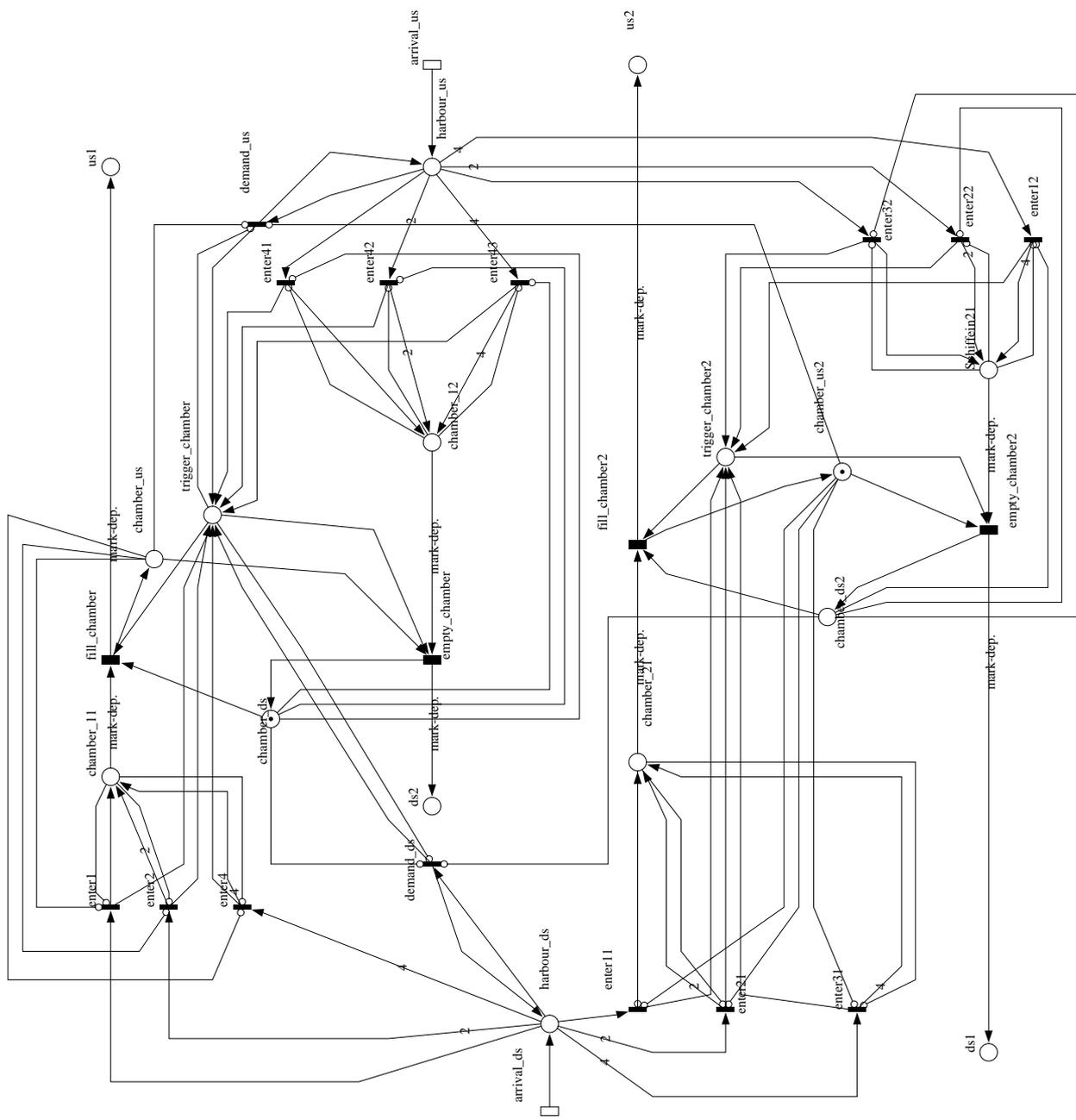


Figure 1: The Petri Net Model of the Shiplift (TimeNET)

# INTERACTION CONTROL IN A COMBINED LOGISTICS AND CHEMICAL PROCESS SIMULATION

Alexander Lavrov, Dietmar Hietel, Stefan Nickel

Fraunhofer-Institute for Industrial Mathematics

D-67663 Kaiserslautern, Germany

E-mail: lavrov@itwm.fraunhofer.de

## KEYWORDS

Parallel & distributed simulation, hybrid systems, logistics simulation, chemical process simulation

## ABSTRACT

This paper describes a synchronisation component of a framework intended at a rapid and transparent integration of (sub)models of logistics components (usually discrete event based) and of chemical processes (continuous) into an overall simulation model of an industrial system. Such kind of model enables an analysis of those behavioral features of the complete system which result from the interplay of its components and are not directly observable under their separate investigation. Domain-specific properties of subsystem interaction allow to implement a flexible, powerful yet simple synchronisation scheme. It operates with standard communication interfaces and does not rely upon arranging a complex specialised platform for distributed simulation. A detailed description of the synchronisation algorithm is given, and some causality related issues are discussed. The prototype framework works with models implemented in eM-Plant (for logistics) and WinZPR (for chemical processes).

## INTRODUCTION

Simulation is a standard technique in the design and analysis of complex industrial systems, in particular those involving interacting discrete and continuous components, e.g. chemical enterprises. Simulation of hybrid systems (Barton 2002) has for a long time attracted attention of researchers and practitioners. Usual practice however still demonstrates a strong separation between the process simulation (primarily continuous) (Turton 2002), and the logistics simulation (mostly discrete) (Banks 2001).

In the models used at one of these two sides, the influence of the other side is usually represented by the imitation of the latter via input/output flows. For the common purposes such a separated scheme

usually does satisfy its needs. In addition, separate consideration is often justified by the disparity between both sides: The process side plays the superior one, and the aim of its simulation is to find the optimal operation modes of the equipment, ensuring high quality of the end product, safety and reliability. The logistics side, on the contrary, is required to fulfil the prerequisites imposed by the process side (e.g. transport service, personnel, etc.), and its simulation serves for determining the necessary capacities and developing corresponding operation modes.

Yet there are applications (e.g. control system testing) where an integrated view of the system, with an explicit interaction between its heterogeneous parts, is especially important. For example, complex causal chains may occur in the operation of the whole system, which can lead to blocking, failures, etc. Such chains often cannot be detected via analysing separate subsystems under restricting assumptions about their interaction.

The main barrier to use industry relevant hybrid simulation is the fact that each of the parties uses completely different simulation tools that are best appropriate for its needs. The most widespread tools are either of discrete (e.g. AutoMod, eM-Plant) or of continuous (e.g. ASPEN, ChemCAD) nature, with sometimes available restricted hybrid features. At the same time, existing hybrid simulation languages and tools do not provide enough application-oriented functionality and do not enjoy a wide popularity in industry. Possibilities to combine specialised simulation models in a distributed framework are rather restricted, since the corresponding tools rarely possess interfaces to special tools and environments of parallel and/or distributed simulation based, e.g. based on HLA (DMSO 2003). In addition, the use of the latter requires special knowledge, experience, essential amount of additional programming including, possibly, an intervention into the source code.

The goal of the work presented in this paper was to develop an easy-to-use open framework with the following properties. Firstly, it should be based on special domain features, allow a fast and transpar-

ent integration of different discrete event and continuous submodels into an overall combined (hybrid) simulation model, avoiding the time- and specific knowledge-consuming application of specialised frameworks. Secondly, it should support the model development process and the usage of optimisation components both for guiding the experiments and for representing the decision-making activities inside a model. The primary reason was to develop simple-to-use facilities for integrating available models, a higher parallelisation etc. being a secondary aspect.

This paper is focused on the development of a synchronisation unit which would, on one hand, follow the main causality-related principles of distributed simulation, and, on the other hand, avoid the technicalities and programming complexity associated with an implementation or usage of a universal distributed simulation environment.

First we give a general characteristic of the components. The main part describes the implemented synchronisation algorithm. After that, a weak conservative time advance scheme, used in the case of multiple models without rollback, is described. Finally, some implementation issues related to causality and communication are discussed.

## INTENDED FRAMEWORK AND CHARACTERISTICS OF SIMULATION COMPONENTS

The synchronisation scheme presented in this paper, is used in the control unit (CU) of a simulation-based decision-support framework (Fraunhofer 2003) consisting of discrete event simulation models, continuous models, an optimisation component, and a groupware component with an associated database (fig. 1). We focus on the part of the framework highlighted in fig. 1 by a dotted line.

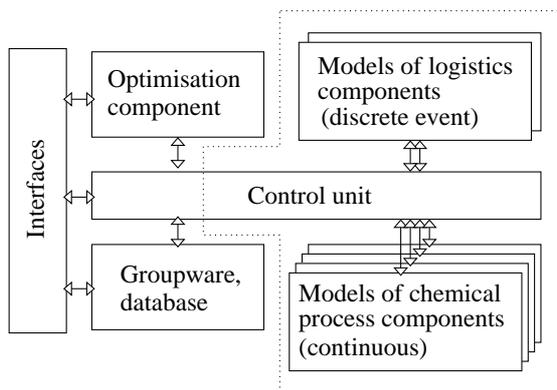


Figure 1: General structure of the combined simulation-based decision-support framework

The modeled system (chemical enterprise) has special properties which can be used while developing a framework for combined simulation:

- P1** Interactions between the logistics components and the process components typically relate to transport operations (arrivals, departures, loading/unloading, pumping, etc.).
- P2** The tolerance of time representation of such operations is usually higher than that of the duration values inside a submodel (especially chemical).
- P3** Interactions take place rarely compared to the internal events of individual models.

From the view of distributed simulation, the to-be-combined models (tools) must be examined concerning the availability/implementability of the following features: (1) rollback, (2) lookahead, (3) monitoring and detection of certain (types of) events and conditions, (4) interruption/resumption of simulation in predetermined points of model (virtual) time, (5) communication and control interfaces, (6) event list observation and external control. Table 1 contains corresponding characteristics of logistics- and chemical process-oriented simulation tools.

Table 1: Characteristics of logistics- and chemical process-oriented simulation tools

	Chemical process simulation tools	Logistics simulation tools
(1)	Operation consists in solving differential equations and representing the results as system's evolution over time. Technically, the main functionality of the rollback can thus be implemented via recalculation ( <b>rerun</b> ) from a given point of time.	Operation consists in processing of event lists. Rollback features and the possibility to rerun the simulation from an intermediate point are usually not available.
(2)	For some modeled reactions the shortest duration may be available, thus giving a basis for a lookahead. Embedded programming features are usually enough for its implementation.	Some operations allow forecasting (e.g. the shortest transportation time) and hence a certain lookahead is possible. It is easily implementable via internal programming features (e.g. SimTalk language in eM-Plant).
(3)	Availability of continuous monitoring and detection of special events and/or conditions varies essentially depending on tools.	Continuous monitoring of conditions and events is implementable via <b>waituntil</b> -like features, while the scope of the functionality is determined by the restrictions on the triggering condition.
(4)	The possibility of recalculation (see feature (1) above) always allows to implement detection via a combination of "model run with trace logging + retrospective detection and evaluation + rerun up to the desired point".	Interruption at a given time point is usually realisable, however for implementation of possible additional requirements, such as "after processing all the events with the given timestamp", additional programming efforts are needed.

Table 1 (continued)

(5)	Communication capabilities of most of the available tools are sufficient for general purpose exchange (at least: file read/write features, data base interfaces, etc.).	Sufficient general-purpose communication features are usually available.
(6)	Event list is not available. Indirect observability is essentially associated with the feature (3) above.	The event list is often open for complete or partial observation. The possibilities to directly influence it are either not available or very restricted.

Analysis of features P1-P3 of the domain and features (1)-(6) (table 1) of simulation tools allows to determine the following main principles of the intended integration framework design:

- simulation proceeds via subsequent advance steps in the two groups of models: those with and without rollback,
- a limited set of interaction events ("I-events") is completely specified,
- simulation steps inside a group are synchronised using a simple barrier-based technique, the barriers being associated with the I-events,
- parallel model runs are possible under availability of the lookahead,
- two strictly distinct message types are used: control and domain-related,
- a wide spectrum of communication options is available: via file read/write, database, sockets, etc.

These principles are implemented in the algorithms and approaches presented in the following sections.

## MAIN SYNCHRONISATION SCHEME

Information on individual models, or logical processes (LPs), and corresponding communication types is available to the CU as specified in table 2:

Table 2: General features of the participating submodels as represented in the CU

path	log. name	var. type	synchr. type	comm. medium
D:\trsp\t.exe	transport	discr.event	cons.	socket
C:\ref\ref.exe	refinery	contin.	optim.	file
...	...	...	...	...

For simplicity, we will call "optimistic" those LPs that are able to perform rollback, and "conservative" the other LPs. The sets of conservative and optimistic LPs will be denoted  $\mathcal{CLP}$  and  $\mathcal{OLP}$ , respectively.

The implemented synchronisation scheme represents a combination of standard techniques of conservative and optimistic synchronisation, involving, in one or another form, the functionality of barrier algorithms, null messages, rollback (fig. 2).

### Control unit actions

```

Set starting time  $t_0$ ; start all LPs
while not termination-condition do
  determine minimal guarantees:
     $t_{guar}^o = \min\{t_{guar_i} \mid LP_i \in \mathcal{CLP}\};$ 
     $t_{guar}^c = \min\{t_{guar_i} \mid LP_i \in \mathcal{OLP}\};$ 
  if  $t_{guar}^o > 0$  and  $t_{guar}^c > 0$  then
    module A, fig. 3
  elseif  $t_{guar}^o = 0$  and  $t_{guar}^c = 0$  then
    module B, fig. 4
  elseif  $t_{guar}^o = 0$  and  $t_{guar}^c > 0$  then
    analogous to module B (fig. 4),
    with value  $t_{guar}^c$  instead of  $t_{lim}^o$ 
  else /* i.e. if  $t_{guar}^o > 0$  and  $t_{guar}^c = 0$  */
    module C-1, fig. 6, or module C-2, fig. 7
  endif
  read the domain-related messages from LPs;
  run the supervisory decision component;
  (LP of the central control system);
  forward messages and commands;
  update  $t_0 = t^*$  (see below)
endwhile

```

Figure 2: CU actions (synchronisation algorithm)

The algorithm consists of cyclic repetition of a model time advance step, starting in the current point  $t_0$  (common for all submodels). Such a step consists of a sequence of actions (denoted as "module" in fig. 2) resulting in one of four possible cases depending on the relation between the lookahead values (guarantees) obtained from the different groups (optimistic and conservative) of LPs. The termination condition corresponds to reaching a given upper bound on the simulation time or completing the to-be-simulated production period.

The case with nonzero guarantees from both, conservative and optimistic, LP groups (fig. 3) is the most efficient: it allows parallel runs of all the models over a common predetermined model time interval and does not require a rollback.

#### Module A

```

Run all models (optimistic and conservative)
from  $t_0$  to  $t^* = \min\{t_{guar}^o, t_{guar}^c\}$ .

```

Figure 3: Module A (both guarantees are available)

In the case when both minimal guarantees are zero, parallel runs are only possible inside a group of LPs (conservative or optimistic), whereas between the

two groups a purely sequential form is used (fig. 4): first, the optimistic LPs are ordered to proceed to the given point  $t_{lim}^o$ .

*Module B; actions illustrated in fig.5(a)*

```

run each  $LP_i \in \mathcal{OLP}$  from  $t_0$  to its first
  I-event at  $t_i^e$ , but not further than  $t_{lim}^o$ ;
let  $t_i^*$  be the stop time of  $LP_i$ th run;
set  $t_{min}^o := \min\{t_i^* \mid LP_i \in \mathcal{OLP}\}$ ;
run "weak conservatively" all  $LP_i \in \mathcal{CLP}$ 
  from  $t_0$  to the first I-event at  $t^*$ ,
  but not further than  $t_{min}^o$ ;
rerun each model  $LP_i \in \mathcal{OLP}$ 
  such that  $t_i^* > t^*$  from  $t_0$  to  $t^*$ .

```

Figure 4: Module B (no lookahead guarantees available)

If an I-event  $e_i$  occurs in an LP, this LP stops just after processing all internal events with the timestamp equal to that of  $e_i$  (step 1 in fig. 5 (a), see next page).

The minimum stopping time  $t_{min}^o$  of all optimistic LPs yields the upper bound for the next advance of the conservative LPs (step 2 in fig. 5 (a)). If there are more than one conservative LPs, then a conservative synchronisation must be applied inside group  $\mathcal{CLP}$  (symbolically represented in fig. 5 by small steps on the "conservative" side).

Taking into account property P2 of the domain, a special "weak conservative" approach is used here which is implemented using three elementary operations on event list (*check the next event time*, *scheduled method call*, and *cancel a scheduled method call*) and thus does not require complicated programming intervention. This approach is explained in a separate section below.

The conservative LPs (as a weak conservatively synchronised group) can either reach the point  $t_{min}^o$  (diagram Ⓐ in step 2 in fig. 5 (a)) or stop earlier at  $t_{min}^c$  if an internal interaction event has occurred (diagram Ⓑ in step 2 in fig. 5 (a)). Let  $t^* = \{t_{min}^o, t_{min}^c\}$ . The third, optional step, performs rollbacks (via rerun) of those optimistic LPs whose end time point at step 1 was greater than  $t^*$ : these are either all optimistic LPs (diagram Ⓒ in step 3 in fig. 5 (a)) or all optimistic LPs except the one(s) stopped at  $t_{min}^o$  (diagram Ⓓ in the same step).

When a nonzero minimal guarantee  $t_{guar}^c$  is available only from the group of conservative LPs, the action sequence is the same as in module B, with the only difference that  $t_{guar}^c$  is used as the upper bound for the optimistic runs at step 1.

If, on the contrary, only the optimistic LPs delivered

a nonzero minimal guarantee, then possible actions can take on two main forms: sequential (module C-1 in fig. 6) or parallel (module C-2 in fig. 7).

*Module C-1; actions illustrated in fig.5(b)*

```

run "weak conservatively" all  $LP_i \in \mathcal{CLP}$ 
  from  $t_0$  to the first I-event at  $t^e$ ,
  but not further than  $t_{guar}^o$ ;
let  $t^*$  be the stop time of the conservative LPs;
rerun the optimistic models to  $t^*$ .

```

Figure 6: Module C-1 (optimistic guarantees available; alternative 1: sequential)

Module C-1 is rollback free, though this is reached via purely sequential runs of the two groups of LPs and thus may be time consuming. Module C-2 allows a partial parallelism and is appropriate when the probability of earlier interaction events during the next runs of the conservative LPs is low.

*Module C-2; actions illustrated in fig. 5(c)*

```

run optimistic models from  $t_0$  to  $t_{guar}^o$  and
  simultaneously run "weak conservatively"
  all  $LP_i \in \mathcal{CLP}$  to the first I-event
  at  $t^e$ , but not further than  $t_{guar}^o$ ;
if stopping time of the conserv. group  $t^* < t_{guar}^o$ 
then rerun the optimistic models from  $t_0$  to  $t^*$ 
endif

```

Figure 7: Module C-2 (optimistic guarantees available; alternative 2: parallel)

## WEAK CONSERVATIVE ADVANCE

Weak conservative advance (fig. 8) consists of repetitive small parallel steps (equal to the tolerance interval) of all conservative LPs; the advance stops as soon as an interaction related event has been detected in any participating LP during the last step, or if the upper bound of the interval has been reached (fig. 9).

*Weak conservative advance; actions illustrated in fig.9*

```

 $t_1 = t_0$ ;  $t_2 = t_0$ ;  $int.event = FALSE$ ;
while  $t_2 < \hat{t}$  and not  $int.event$  do
   $t_2 = t_1 + \min\{\Delta t, t_{max} - t\}$ ;
  run conservative LPs over  $[t_1, t_2]$ ;
  if interaction event(s) registered in some LPs
    then  $int.event = TRUE$ 
  endif
   $t_1 = t_2$ ;
endwhile

```

Figure 8: Weak conservative advance

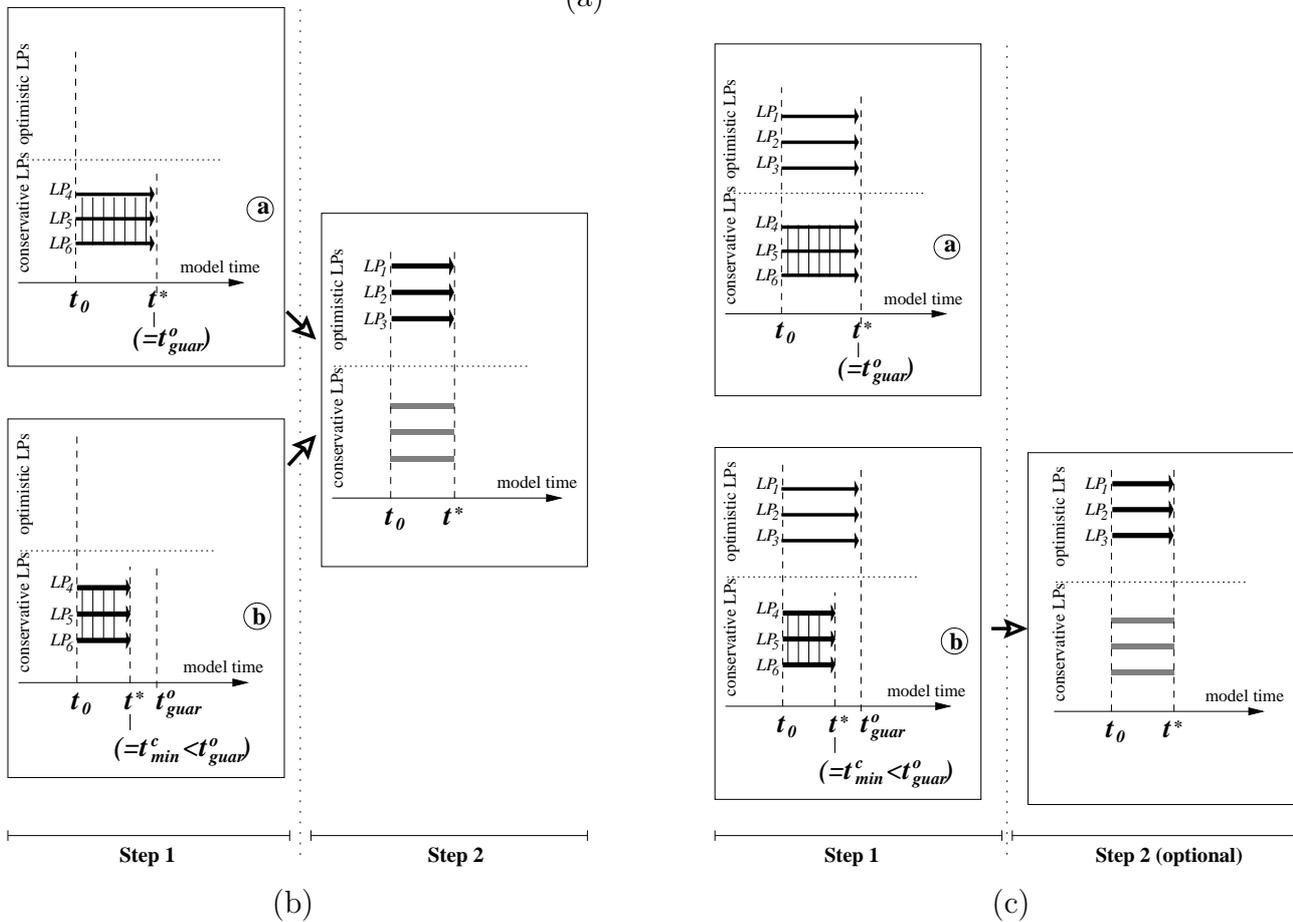
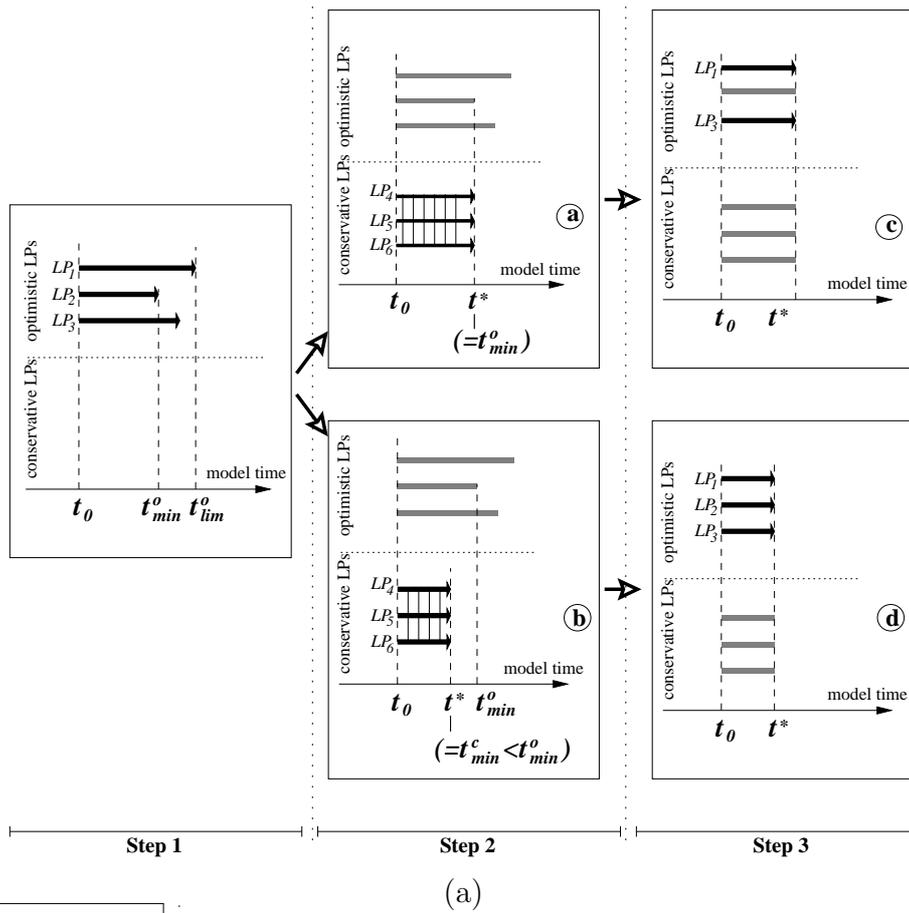


Figure 5: Illustrating module actions: (a) action sequence of module B from fig. 4, (b) action sequence of module C-1 from fig. 6, (c) action sequence of module C-2 from fig. 7

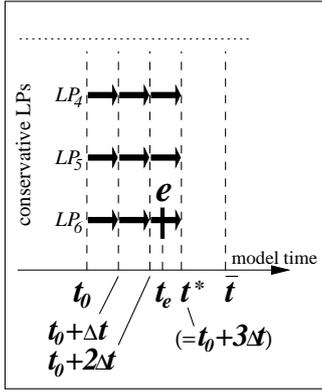


Figure 9: Weak conservative advance

The underlying idea is based on property P2 of the domain and on an assumption about the existence of some time tolerance interval which allows to extend certain duration values (and consequently the timestamps of corresponding events events). The I-events which occurred in an LP inside this interval will be considered by the other (recipient) LPs as having the timestamp of the end of the interval. This is a plausible assumption since the tolerance (precision) of several seconds (sometimes even dozens of second) in the logistics processes (transport arrivals and departures) is usually acceptable. On the other hand, this allows to avoid synchronisation on every time increment related only to internal events in every LP.

### CAUSALITY AND COMMUNICATION: IMPLEMENTATION ISSUES

The CU communicates individually with each LP. Two types of messages are used: control messages (domain independent), and domain-related messages, each type having its own fixed format. LPs exchange control messages with the CU at the interaction (barrier) points. Part of the synchronisation-relevant information maintained by the control unit is shown in table 3.

A control message, issued by the CU, contains, among other items, its number (each pair "CU - LP" has an independent counter), timestamp (issuing time), command ("forward", "rerun" or "stop"), the start point and the end point of the time interval for the corresponding model run. For the replies, issued by an LP, additional positions are reserved for the real end point of the run and for the indicator of the interaction events occurred during the last run.

A domain-related message contains information on the event type (arrival, departure, etc.), order number, substance, aggregate state, volume, components and their mol percentage. Domain-related messages from other LPs are processed by an LP only at the beginning of the "forward" run. Domain-related message exchange can only be initiated by the CU,

which issues a special "read impulse" in the control message.

A strict sequence of internal and external information processing is used (fig. 10): first, own events up to the barrier point are completely processed. Afterwards, the information on the (simultaneous) events in other LPs becomes available. Thus, it can impact only the future evolution of the process. This ensures that only unconditional external information is used. (Note that all the domain-related messages distributed after each iteration in algorithm as in fig. 2 have the same timestamp corresponding to the barrier point.) The messages generated by an optimistic LP during its run, which have become invalid due to a subsequent rerun, are destroyed without having been read by the recipients.

Distribution of the domain-related messages can be organised in two ways: either in a centralised manner, when the CU reads the output buffers of the LPs and then forwards the messages to their recipients, or directly by the LPs, which write their messages into the input buffers of the recipient LPs. For the prototype implementation, the first way was chosen, since the centralised distribution allows also to easily prepare an integrated data set on the complete system, which can be sent to the optimisation component making some strategic decisions.

The participating LPs are assumed to be deadlock-free. The obligatory control messages at the barrier points, together with the strict pre-defined sequencing of LP group advances depending of the guarantees availability, as well as the information processing order as in fig. 10 ensures the deadlock prevention and a causality respecting synchronisation.

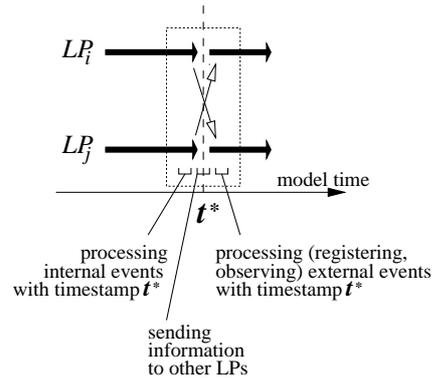


Figure 10: Information processing at the barrier point

### CONCLUSIONS

A framework for integrating discrete-event and continuous simulation models implemented by different tools into a combined model is presented. The framework uses a simply-to-realize and transparent

Table 3: Fragment of a synchronisation management table

LP	request No.	request time	command	$t_{start}$	$t_{end}$	...	reply No.	reply time	int. events	...
transport	3	5.0	forward	7.0	10.0	...	4	9.0	true	...
refinery	7	9.0	rerun	7.0	9.0	...	6	10.0	false	...
...	...	...	...	...	...	...	...	...	...	...

synchronisation scheme. This scheme is based on accounting only the predefined sets of interaction-relevant events and on special rules for preventing causality violation. The synchronisation control performs the runs of groups of logical processes depending on their rollback capabilities. The framework can be used both for a rapid prototyping of complex models of hybrid systems, and for in-depth simulation analysis of a complete system whose individual components' models are available.

## ACKNOWLEDGEMENTS

This research was supported by grant UMTS 126 ("SILVER") of German Federal Ministry for Education and Research.

## REFERENCES

- Alur, R.; T. Dang, ; J. Esposito; Y. Hur; F. Ivancic; V. Kumar; I. Lee; P. Mishra; G. Pappas; and O. Sokolsky. 2003. "Hierarchical modeling and analysis of embedded systems." *Proceedings of the IEEE* 91, No. 1, 11-28.
- AspenTech. 2003. *Aspen Engineering Suite*. <http://www.aspentech.com/products>
- AutoMod. User's Manual v 10.0*. Vol. 1,2. Brooks Automation, Inc. - AutoSimulations Division. 2001.
- Banks, J. (Ed.) 1998. *Handbook of Simulation*. John Wiley & Sons, Inc. Prentice-Hall, N.J.
- Banks, J.; J.S. Carson, II; B.L. Nelson; and D.M. Nicol. 2001. *Discrete-Event System Simulation*. Prentice-Hall, N.J.
- Barton, P.J. and C.K. Lee. 2002. "Modeling, Simulation, Sensitivity Analysis, and Optimization of Hybrid Systems." *ACM Transactions on Modeling and Computer Simulation* 12, No. 4, 256-289.
- van Beek, D.A.; J.E. Rooda; and M. van den Muyzenberg. 1996. "Specification of Combined Continuous-Time/Discrete-Event Models." In *Proc. 1996 European Simulation Multiconference*. Budapest, June 1996, 219-224.
- van Beek, D.A.; J.E. Rooda. 2000. *Multi-domain Modelling, Simulation, and Control*. In *Proceedings of 4th International Conference on Mixed Processes: Hybrid Dynamical Systems (ADPM2000)*, Dortmund, 139-146.
- ChemCAD. 2003. *CHEMCAD Process Simulation Software*. <http://www.chemcad.fr/en/index.html>.
- eM-Plant 6.0: Objects Manual and Reference Manual*. Tecnomatix. 2001.
- DMSO (Defence Modeling and Simulation Office of U.S. Department of Defense). 2003. *High Level Architecture*. <https://www.dmsomil/public/transition/hla>
- Esposito, J.M.; G. Pappas; and V. Kumar. 2001. "Accurate event detection for hybrid systems." In *Proceedings of HSCC2001 (Hybrid Systems: Computation and Control)*, Rome, Italy, March 2001, 204-217.
- Fraunhofer UMSICHT. 2003. *Anlagensimulation mit WinZPR*. <http://www.umsicht.fhg.de/WWW/UMSICHT/Produkte/software/zpr/index.html>
- Fujimoto, R.M. 2000. *Parallel and Distributed Simulation Systems*, John Wiley & Sons.
- Kim, Y.J. and T.G. Kim. 1998. "A Heterogeneous Simulation Framework Based on The DEVS Bus and the High Level Architecture". In *Proceedings of the 1998 Winter Simulation Conference*. Washington, D.C., 421-428.
- Lampert, L. 1978. "Time, clocks, and the ordering of events in a distributed system." *Communications of the ACM* 21, No. 7, 558-565.
- Fraunhofer (ITWM, IML, UMSICHT, FIT). 2003. *SILVER: Simulationsbasierte Systeme zur Integration logistischer und verfahrenstechnischer Entscheidungsprozesse*. Berichte zu Arbeitsbereichen 1 ("Leit-szenario"), 2 ("Gesamtkonzepte"), 3 ("Gekoppelte Modelle zur hybriden Simulation").
- Turton, R.; R.C. Bailie; W.B. Whiting; and J.A. Shaeiwitz. 2002. *Analysis, Synthesis, and Design of Chemical Processes*. Prentice Hall.
- Zeigler, B.; H. Praehofer; and T.G. Kim. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, London.

## AUTHORS

Dr. Alexander Lavrov is a Research Scientist at the Optimisation Department of Fraunhofer-Institute for Industrial Mathematics (ITWM) in Kaiserslautern (Germany).

Dr. Dietmar Hietel is a Research Scientist at the Department of Transport Processes of Fraunhofer ITWM.

Dr. Stefan Nickel is the Head of the Optimisation Department of Fraunhofer ITWM and Professor at the University of Saarbrücken (Germany).

# COMPARISON OF PREDICTION METHODS FOR URBAN NETWORK LINK TRAVEL TIMES

Joanna K. Hartley  
School of Computing and Mathematics, The Nottingham Trent University  
Burton Street, Nottingham, NG1 4BU, U.K.  
Tel. +44 (0) 115 848 6172, Fax. +44 (0) 115 848 6518  
Email: [Joanna.Hartley@ntu.ac.uk](mailto:Joanna.Hartley@ntu.ac.uk)

## KEYWORDS

Transportation, Optimisation, Efficiency, Prediction methods.

## ABSTRACT

Traffic congestion is becoming a serious environmental threat that must be resolved quickly. The mobile travel information system developed at The Nottingham Trent University enables the integration of data concerning traffic flows and individual journey plans thus making it possible to perform optimisation of travel. This paper focuses on the issue of provision of real-time information about urban travel and assistance with planning travel. Nottingham's SCOOT (Split Cycle Offset Optimisation Technique) traffic-light control system provides real-time information about the link travel times within certain areas of the city. However, rather than using link travel times at the time of the request, it is more effective to predict the link travel times for the time of travel along the particular links. The future link travel times depend upon the historical travel time of the link (for the specific time step in the day) as well as the current link travel time. Consequently, the link weights are a combination of real-time data, historical data and static data. Three prediction methods have been implemented and tested in the context of Nottingham's urban road network. The preliminary results suggest that the information discounting technique gives the best results.

## BACKGROUND

Traffic congestion is becoming a serious environmental threat that must be resolved quickly. Great Britain has become a role model in the battle against global pollution. The Prime Minister, Tony Blair, has acknowledged that a 20% reduction in carbon dioxide emissions in Great Britain is a credible target for the year 2010 (Brown, 1997). However, significant measures are necessary to attain this target. Road vehicles and industry are the main sources of pollutant emissions. In the United Kingdom, road vehicles are responsible for over 50% of the emissions of nitrogen oxides and over 75% of carbon monoxide emissions (DETR, 1998). Congestion is already a major problem

in many areas and traffic volume is set to grow by 30% in the next 20 years (MacAskill, 1999).

This paper describes the infrastructure that is currently being developed at The Nottingham Trent University to facilitate multi-modal travel throughout the city of Nottingham. This paper focuses on the issue of provision of real-time information about urban travel and assistance with planning travel. This includes consideration of uncertainty about traffic delays, inconvenience of parking and the variability of travel time along urban links.

## TRAVEL INFORMATION SYSTEMS

There are a number of ways of informing travellers about the location of congestion areas – such as, radio or television broadcast and variable message signs. The growing body of opinion, that the traditional forms of supervisory control are both too expensive and inaccurate, prompts new development. The traditional forms require full involvement of a human operator. However they do not take into account the specific requirements of individual journeys. In particular, because of the protection of privacy, the crucial information about the intended destinations of individual vehicles is not normally available to these controllers and, even if it was, it could not be processed efficiently. On the other hand, an attempt to delegate the responsibility for journey optimisation to road users by informing them (through radio broadcasts or variable message signs) about the best routes, that are relevant to various journeys, is bound to be counterproductive because of the resulting information overload.

To eliminate these constraints, this research project takes a fundamentally different approach and rather than aiming at maximising the efficiencies of the use of individual modes of transport taken in isolation, it considers a broader multi-modal travel framework. Travel requirements are defined in terms of journeys and the mode of travel is just one of the decision variables. An important feature of our approach is that it recognises the individual nature of journeys. The enquirers are able to select their journey according to their individual preferences, such as the importance of

a short journey time or the importance of timely arrival at the destination. Although the answers may be highly subjective it must be remembered that it is precisely these preferences that make people opt for one mode of transport or another. In this sense, multi-modal travel optimisation offers good mapping onto human decision-making.

Travel information systems are now being developed which incorporate route guidance systems to divert drivers away from the congested areas either by change of travel mode or travel route. Dynamic guidance is of the greatest benefit to travellers, as new routes and travel modes can be suggested as conditions change (McDonald and Montgomery, 1996). The system will aid the road users by providing access to information that is not readily observable from the current location of the traveller, yet is relevant because of the planned journey. The wide use of such a system will reduce the amount of congestion within the city, by the choice of departure time and shorter routes as suggested by the route guidance system. This will lead to expected reductions of pollutant emissions in currently congested and critical areas.

The system must be capable of simultaneous data acquisition, processing and dissemination of the traffic/travel advice in real-time to a full spectrum of end users. Preston et al. (1993) proposed the use of in-vehicle telephones to communicate with a remote computer. The increasing use of mobile phones by the general public takes their suggestion one step further, opening access to the decision support system to many more users. The integration of data concerning traffic flows, public transport and individual journey plans thus makes it possible to perform multi-modal optimisation of travel.

The system enables progression from a passive mode of interaction between traffic control systems and road-users (one-way flow of information) to an active mode. Within the active mode, the road users supply the information about their intended destination (without disclosing their identity) and, in response, receive customised traffic information that optimises their journeys.

So, there is a need for a hierarchical urban sustainability structure that would be specifically responsible for providing a global optimisation layer while relying on the local optimisations affected by the individual organisations responsible for urban transport (Peytchev and Bargiela, 1998; Pursula, 1998). The development of such a structure should clearly rely to a maximum extent on the standard computer and public communication systems. However, the feasibility of such an undertaking has to be proven by detailed consideration of the technological constraints of the sub-systems that are to be integrated.

The developed structure is a Distributed Memory Environment (DIME) (Peytchev and Bargiela, 1998) that manages the data from a number of sources (Traffic Control Centre, public transport company and the user) (Figure 1). Nottingham Traffic Control Centre continues its kind agreement of allowing the Intelligent Simulation and Modelling group to have access to its Traffic Control System (SCOOT and congestion data), providing the necessary current traffic information. Nottingham City Transport has obligingly approved the use of the necessary information concerning their bus timetables.

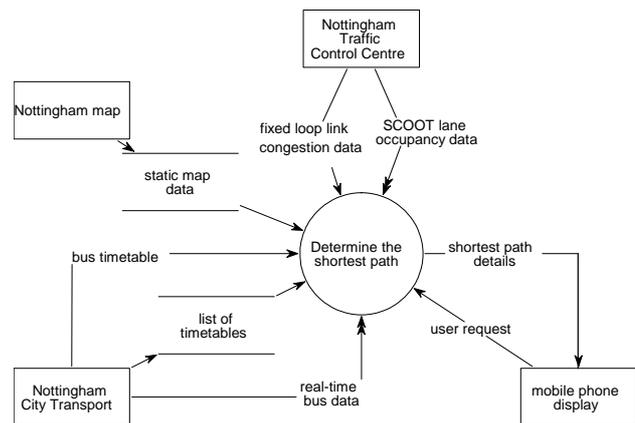


Figure 1: Determining the Shortest Path

The user communicates with the DIME system via a mobile phone. The advantage of a mobile phone is that there do not exist the constraints of being part of a private vehicle's equipment or being deployed at a particular location. Along with the increasing use of mobile phones by the general public, this means that the system has a much broader user base. By implication, this will result in a much greater impact on travel mode switching decisions (Bargiela and Berry, 1999). Also, the system is easy to use and not prohibitively expensive.

## ROUTE GUIDANCE

There are many methods of path finding that are appropriate for use within the spectrum of route guidance. Some of these methods have been considered and evaluated in the context of multi-modal travel in Nottingham's urban network. The results are published in (Hartley and Bargiela, 2001; Hartley, 2003a). This paper concentrates on the delivery of timely route guidance given the available real-time traffic information. This involves the prediction of traffic. Three prediction methods will be presented, and tested using private vehicle real-time traffic data. However, any of these prediction methods may also be used in the context of public transport and multi-modal travel.

## Urban Network Information

In the context of private vehicle travel, the pre-requisite to determining any shortest path in an urban network is having information about the road link weights (Figure 2).

The necessary input data, mostly provided by the different organisations managing the urban network, comprises SCOOT (Split Cycle Offset Optimisation Technique) link data, congestion data for fixed loop links and static map data (Figure 1).

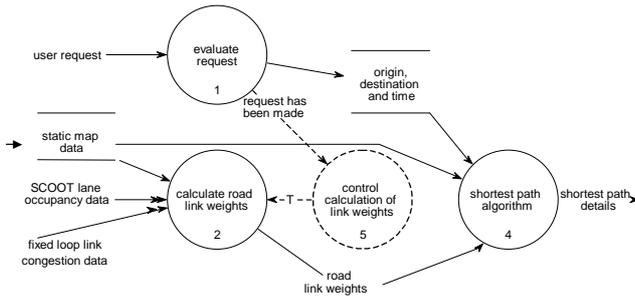


Figure 2: Calculation of Travel along Links

SCOOT is an intrinsic part of Nottingham's traffic-light control system comprising induction loops that detect the presence of vehicles in real-time. The SCOOT link data provide real-time information about the link travel times within certain areas of the city. Fixed loop congestion data again provide real-time information – these data are specific to certain junctions or roads (distinct from the SCOOT-managed areas). The static map data include information about the topology of the urban network and the length of roads. The integration of SCOOT lane occupancy data (leading to link times in SCOOT-managed areas), fixed loop congestion data (leading to link times in some non-SCOOT areas) and static map data (providing estimated static data of the link times for the remainder of the network) are manipulated into up-to-date, reliable information of alternative paths and adverse traffic conditions on appropriate links. This enables the derivation of the optimal route for travel by private vehicle.

## Travel by Private Vehicle

Dijkstra's algorithm (1959) has been used to determine the optimal route by private vehicle. Dijkstra's algorithm builds an expanding list of examined vertices and looks at paths through vertices on the list. The path with the smallest total of link weights is incrementally found.

## Dijkstra's algorithm

```

pathlength(all links) = ∞
marked(all links) = .false.
marked(origin) = .true.
pathlength(origin) = 0
do for all links until marked(destination) = .true.
{
  search for all pairs of nodes s.t.
  marked(node1) = .true. & marked(node2) = .false.
  then
  pathlength(node2) = min(pathlength(node2),
    pathlength(node1)+length(node1,node2));
  from this set determine which node2 has minimum
  pathlength then
  marked(node2) = .true.
}
  
```

## PREDICTION OF LINK TRAVEL TIMES

Rather than using link travel times at the time of the request, it is more effective to predict the link travel times for the time of travel along the particular links. The method used was developed as part of the Ali-Scout project (Kotsopoulos and Xu, 1993). The future link travel times depend upon the historical travel time of the link (for the specific time step in the day) as well as the current link travel time. The process is as follows:

$$D = \frac{T_h(l,n)}{T_{cur}(l,n)} \quad (1)$$

where  $T_h(l,n)$  is the historical mean travel time of link  $l$  at time step  $n$

and  $T_{cur}(l,n)$  is the current (time step  $n$ ) travel time at link  $l$ .

$$T_p(l,m) = \frac{T_h(l,m)}{D} \quad (2)$$

where  $T_p(l,m)$  is the predicted travel time of link  $l$  at future time step  $m$ .

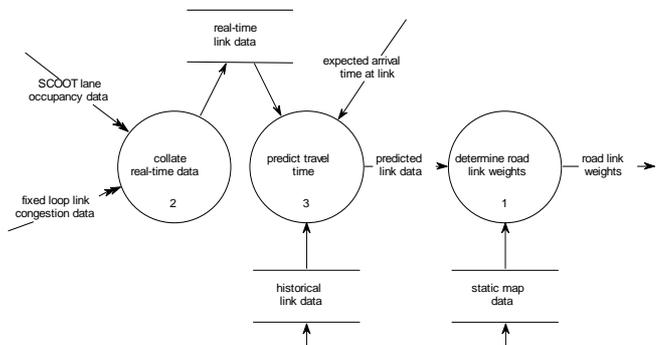


Figure 3: Combination of Historical, Current and Static Data

Consequently, the link weights are a combination of real-time data, historical data and static data (Figure 3).

### Information Discounting

As the state of the network can change extensively in a short period of time, a combination of real-time data and historical data should be used, with the proportions dependant on the expected time taken to arrive at the measured link (Kotsopoulos and Xu, 1993). Kotsopoulos and Haiping (1993) propose information discounting:

$$T_{dis.}(l,m) = a * \frac{T_h(l,m)}{D} + (1-a) * T_h(l,m) \quad (3)$$

where  $T_{dis.}(l,m)$  is the predicted travel time of link  $l$  at future time step  $m$  using the information discounting method, and where  $a$  is a decreasing function of  $(m-n)$ , say  $e^{-k(m-n)}$ . An appropriate choice of  $k$  will lead to accurate predictions. For  $10 < k < 90$ , the point at which historical data has more influence than current data ranges from ten minutes to one hour into the future.

### Resolution of Data

Abdulhai et al. (1999) argue that the level of data aggregation should be comparable to the prediction horizon for best accuracy. The method is an adaptation of the Ali-Scout information discounting method.

$$D_m = \frac{\sum_{i=2n-m-1}^n T_h(l,i)}{\sum_{i=2n-m-1}^n T_{cur}(l,i)} \quad (4)$$

$D_m$  is the ratio between the historical information aggregated over the previous time period of  $(m-n)$  and the most recently collected information aggregated over the same time period.

$$T_{res.}(l,m) = a * \frac{\sum_{i=n+1}^m T_h(l,i)/(m-n)}{D_m} + (1-a) * \frac{\sum_{i=n+1}^m T_h(l,i)}{m-n} \quad (5)$$

$T_{res.}(l,m)$  is the predicted travel time of link  $l$  at future time step  $m$  using a resolution of data that is comparable to the prediction horizon.

### Prediction Updating

The prediction of events in the near future is often considered to be highly related to recent past events. However, when determining predictions in the longer future, this correlation often decreases. The use of

predictions in the near future to determine predictions in the longer future results in an iterative process.

$$T_{iter.}(l,m) = a * \frac{T_h(l,m)}{D_{m-1}^p} + (1-a) * T_h(l,m) \quad (6)$$

$$D_{i:i>n}^p = \frac{T_h(l,i)}{T_{iter.}(l,i)} \quad (7)$$

$$D_n^p = \frac{T_h(l,n)}{T_{cur}(l,n)} \quad (8)$$

$T_{iter.}(l,m)$  is the predicted travel time of link  $l$  at future time step  $m$  using prediction updating.  $D_i^p$  is the ratio between the historical information and the predicted travel time (for a specified time period in the day,  $i$ ).

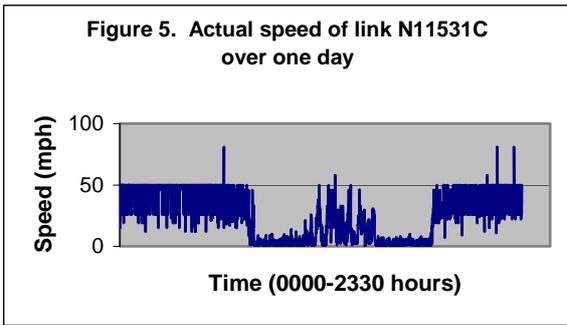
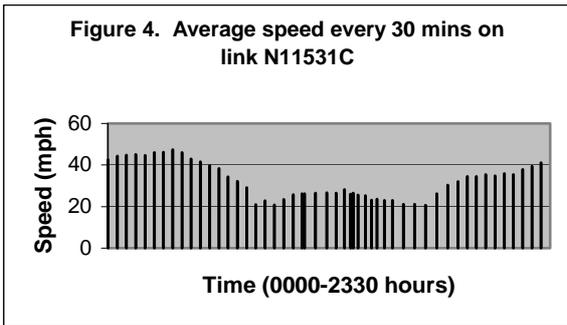
## REAL URBAN TRAFFIC NETWORK APPLICATION

Currently, the available real-time information in the Nottingham urban network is collected from SCOOT detectors, which monitor highly traversed links within the city centre and the arterial routes into the city. 100 links out of 2018 are currently equipped with SCOOT inductive loop detectors. The SCOOT data consists of a large amount of traffic control information relayed in the form of messages (Siemens PLC, 1997) (which include information about flow, occupancy, delay and speed etc.). The U06 message provides information every 30 seconds about the average point-speed of a private vehicle travelling along a link (measured over the last 5 minutes). This speed and knowledge of the link length is used to estimate the current travel time of the link. As some of the routes across Nottingham may take up to one hour to traverse, it is not sufficient to use the current travel time estimations (Hartley, 2003b). So, instead predictions of travel time are used (as described in section 3.2).

## RESULTS

The available historical U06 messages will be used to determine the validity of the prediction method in the context of Nottingham's urban network. The results will also show how the incorporation of real-time information routes traffic away from congested areas. It will also be determined how capable the methods are in dealing with the transition between peak and off-peak conditions.

Figures 4 and 5 show that the historical speed cannot be relied upon, as the speeds fluctuate even within a short period on a single link on a single day.



The efficiency of the algorithm is of paramount importance, so that the information provided to the user is timely and thus relevant. For Nottingham's network of 597 nodes and 2018 links, Dijkstra's algorithm has a system run-time of 0.4 seconds. The execution time of the predictive route finder algorithm should not be greater than a few seconds.

#### Information Discounting

Using  $k=1/60$ , Kotsopoulos' (1993) information discounting method gives the following results (Figure 6), where one step is equivalent to 30 seconds.

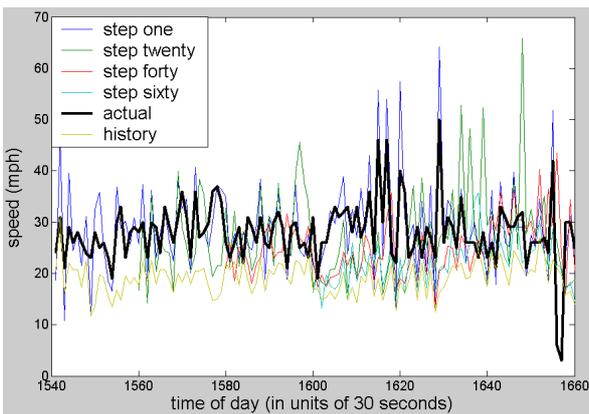


Figure 6: Ali-Scout Prediction with Information Discounting

The mean square error (MSE) between the actual speed and predicted speed is presented in Table 1. The methods have also been tested to determine how they cope with incidents.

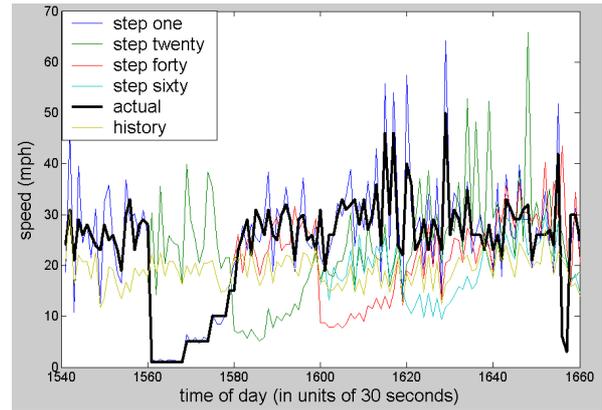


Figure 7: Prediction using Ali-Scout with Information Discounting after an Incident

Figure 7 shows that the information discounting method lessens the effect of the incident as the prediction moves further into the future. The MSE between the actual speed and predicted speed is shown in Table 2.

#### Resolution of Data

Data resolutions that are comparable to the prediction horizons have been used in conjunction with the Ali-Scout (with information discounting) prediction method. Figure 8 shows how the results show much less fluctuation as the step size increases. Figure 9 shows that after an incident, the prediction method is considering data prior to the incident (as part of its averaging process) and consequently poor performance is shown. The MSE between the actual speed and predicted speed in both cases are shown in Tables 1 and 2.

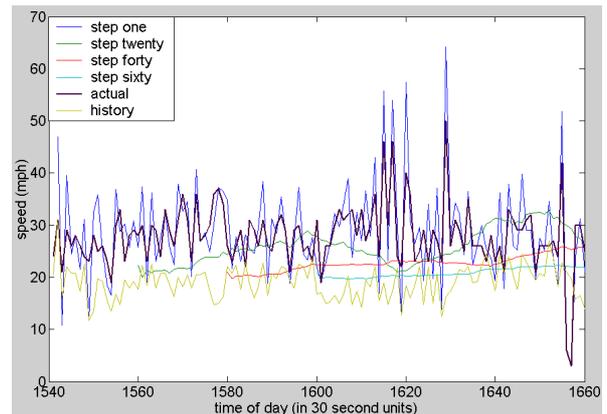


Figure 8: Prediction with Data Resolution

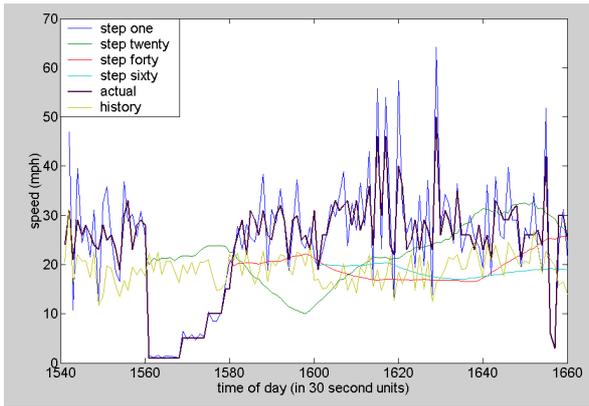


Figure 9: Prediction with Data Resolution after an Incident

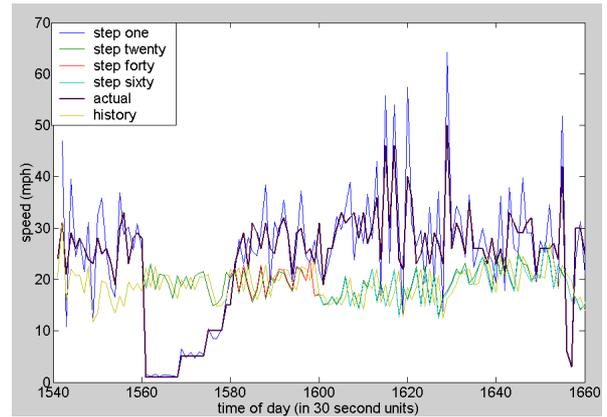


Figure 11: Prediction using an Iterative Process after an Incident

### Prediction Updating

The iterative process using predictions for the near future will be implemented to determine larger prediction horizons. Clearly this method is more time-consuming than the previous two methods as numerous calculations are required as part of the iterative process. Figure 10 shows that the method tends to severely underestimate the actual value.

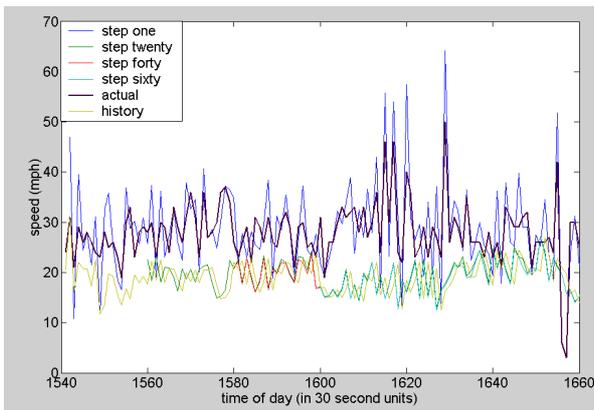


Figure 10: Prediction using an Iterative Process

In a similar manner to the data resolution method, Figure 11 shows that after an incident, the iterative method is considering data prior to the incident (as part of its iterative process) and consequently poor performance is shown. In fact, there is little evidence of the predicted data being influenced at all by the incident. Again, the MSE between the actual speed and predicted speed in both cases are shown in Tables 1 and 2.

Table 1: The Mean Square Error between the Actual Speed and Predicted Speed

Step number	Prediction method			
	Historical	Information Discounting	Resolution of Data	Prediction Updating
One	151.1	42.5	42.5	42.5
Twenty	151.1	51.8	84.5	161.45
Forty	151.1	53.8	84.3	169.72
Sixty	151.1	89.5	113.6	192.72

Table 2. The Mean Square Error between the Actual Speed and Predicted Speed after an Incident

Step number	Prediction method			
	Historical	Information Discounting	Resolution of Data	Prediction Updating
One	160.5	40.0	40.0	40.0
Twenty	160.5	51.6	155.9	174.18
Forty	160.5	53.1	124.3	169.76
Sixty	160.5	89.5	146.4	192.72

### DISCUSSION

The above clearly show that the Ali-Scout prediction method with information discounting alone gives the best results. The resolution of data method gives better results than the historical information in all cases, while the prediction updating method gives worse results. A value of  $k=1/60$ , as part of the information discounting calculation, was used in all cases. Recalibration of this value may give better results for each of the methods.

### CONCLUSIONS

The Ali-Scout information discounting prediction method (Kotsopoulos and Haiping, 1993) has been shown to be more reliable than historical information in the context of link travel times in Nottingham's urban transport network. The difference is significant when predicting travel times up to 40 minutes ahead of

time. This prediction method has been shown to be capable of dealing with incidents.

This paper demonstrates the necessity of real-time information when providing traffic/travel information to the general public. The use of real-time data provides the user with information about the state of the network, not normally foreseeable by the traveller.

It should be noted that minimisation of travel time by private vehicle does not necessarily produce the optimal route from 'door-to-door'. With the increasing ownership of private vehicles, there is more demand on the limited number of parking spaces within any city. Consequently, the inconvenience of parking can make travel by private vehicle be less preferable especially for those travellers who are particularly adverse to travel time uncertainty. So users may ultimately be encouraged to travel by public transport instead. This becomes especially apparent when travel advice includes both private vehicle and public transport as modes of transport, as is the case in the developed real-time travel information system detailed in section 2.

## FUTURE WORK

Studies have shown that the acceptance of route guidance is strongly correlated to any previous experience of the system. Simulation (Peytchev and Bargiela, 1995) will be used to test how the use of the route guidance system will enhance the progression of the traveller (McDonald et al., 1995).

Due to the large amounts of static and real-time data that will be used by the path finding algorithm, there are a number of issues to investigate with regard to storing information. The appropriateness of storing set paths, or calculating paths on demand will be investigated. The pruning of the urban network will be necessary – this may be achieved in a pre-processing mode or as part of the algorithm. Also, further analysis of multiple users (of the order of 100) will need to be considered to continue to provide a viable service. Some possible long-term solutions are the use of more processors, parallel algorithms, or some form of artificial intelligence (such as neural networks).

The inherent fluctuation of both past and future travel times along links means that any predicted travel times are subject to uncertainty. Consequently, an extension of the prediction method to determine confidence intervals will be considered in the future.

## REFERENCES

Abdulhai, B., Porwal, H., Recker, W., 1999. "Short Term Freeway Traffic Flow Prediction Using Genetically-Optimized Time-Delay-Based Neural Networks", *California PATH Working Paper*, ISSN 1055-1417.

Bargiela, A., Berry, R., 1999. "Every BIT counts", *Traffic Technology International*, Feb/Mar, pp 63-66.

Brown, P., 1997. "Britain's Green Lead at UN", *The Guardian*, 24 June 1997.

DETR, 1998. "Air Pollution – What it Means for Your Health", Air and Environmental Protection, <http://www.environment.detr.gov.uk/airq/aqinfo.htm>.

Dijkstra, E.W., 1959. "A Note on Two Problems in Connection with Graphs", *Numerische Mathematic*, 1, pp 269-271.

Hartley, J.K., 2003a. "Efficiency vs. Correctness of a Travel Information System", *Proc. UKSim 2003*, April 2003, ISBN 1-84233-088-8, pp 214-219.

Hartley, J.K., 2003b. "Prediction of Link Travel Times in the Context of Nottingham's Urban Road Network", *Proc. of 17<sup>th</sup> European Multi-Conference*, Nottingham, June 2003, ISBN: 3-936150-25-7, pp 423-428.

Hartley, J.K., Bargiela, A., 2001. "Decision Support for Planning Multi-Modal Urban Travel", *Proc. of 13th European Simulation Symposium*, Marseille, October 2001, ISBN: 90-77039-02-3, pp 387-391.

Kotsopoulos, H.N., Haiping, X., 1993, "An Information Discounting Routing Strategy for Advanced Traveller Information Systems", *Transportation Research Part C*, Vol. 1, No. 3, pp 249-264.

MacAskill, E., 1999. "Promise of a better, faster, more reliable system", *The Guardian*, 14 December 1999.

McDonald, M., Hounsell, N.B., Njoze, S.R., 1995, "Strategies for Route Guidance Systems Taking Account of Driver Response", *Pacific Rim Trans. Tech. Conf., 1995 Vehicle Navigation and Info. Systems Conf. Proc. 6<sup>th</sup> International VNIS*, pp 328-333.

McDonald, M., Montgomery, F.O., 1996. "Urban Traffic Control In Europe", *Proc. Instn. Civ. Engrs. Transp.*, Vol. 117, February, pp 50-56.

Peytchev, E., Bargiela, A., 1995, "Parallel Simulation of City Traffic using PADSIM", *Proceedings of Modelling and Simulation Conference ESM'95*, Prague, Eds. Snorek, Suhansky, Verbraeck.

Peytchev, E., Bargiela, A., 1998. "Traffic Telematics Software Environment", *Proc. European Simulation Symposium*, Oct. 1998, ISBN 1-56555-147-8, pp 378-382.

Polenta, T., Hartley, J.K., 2003. "A comparative Study of Stochastic 'Least-Time' Path Algorithms in the Context of the Nottingham Urban Network", *Proc. UKSim 2003*, April 2003, ISBN 1-84233-088-8, pp 194-200.

Preston, J.M., May, A.D., Aldridge, D.M., 1993, "The Specification of Trip Planning Systems", *IEE Colloquium on 'Electronics in Managing the Demand for Road Capacity'*, pp 2/1-4.

Pursula, M., 1998. "Simulation of Traffic Systems - An Overview". *Proc. 10th European Simulation Symposium*, Nottingham Trent University, October 1998, pp 20-24.

Siemens PLC, 1997. 'SCOOT User Guide', Poole, Issue 17.

#### **AUTHOR BIOGRAPHY:**



**JOANNA HARTLEY** was awarded a BSc (Hons) degree in Mathematics at the University of Durham in 1991. In 1992, she became a research assistant in the Department of Computing at The Nottingham Trent University and was awarded a PhD in 1996. The title of her PhD is "Parallel

Algorithms for Fuzzy Data Processing with Application to Water Systems". She is now a senior lecturer at The Nottingham Trent University and an active member of the Intelligent Simulation and Modelling group. She is a member of the UKSim committee and was an associate editor of UKSim 2003 and a member of the organising committee for ESM'03. Her current research interests include parallel processing, mathematical modeling and probabilistic state estimation relating to urban traffic networks and water distribution systems.

# THE DESIGN AND ASSESSMENT OF NEXT GENERATION AUTOMATED CONTAINER TERMINALS

Yvo Saanen  
TBA Nederland / Delft University of Technology  
Jeroen van Meel  
Port Authority Rotterdam  
Alexander Verbraeck  
Delft University of Technology

## KEYWORDS

Container terminal robotisation, simulation, costs, productivity, new technology.

## ABSTRACT

One of the answers to the ever-increasing transshipment volumes that are required within the same timeframe at the quays of the world's large container ports, is the automation of the processes. However, until now ECT in Rotterdam and HHLA (CTA) in Hamburg are still the only two stevedores that have fully automated the transport from quay to stack as well as the stacking operations. Both operators have chosen to apply RMGs in combination with AGVs. However, a new automated concept is about to be introduced: the automated lifting vehicle (ALV). In this paper a comparison by means of simulation and cost modelling is made between the operational productivity of an AGV-RMG and an ALV-RMG terminal. We will show the pro's and con's of each concept and assess the dynamic behaviour in a detailed simulation model. Furthermore, we will compare the two automated concepts with a manually operated shuttle carrier (a 1 over 1 straddle carrier, in essence). The results show that for cost reasons, the ALV should not be preferred over AGVs, although you less vehicles to achieve the same quay crane productivity. Furthermore, ALVs are not yet to be considered as proven technology, which is for most terminals an important criterion to assess the project risk..

## INTRODUCTION

Since 1991 container terminal robotisation has become the new way of designing terminals with a capacity exceeding 1 mio TEU, providing a cost-efficient alternative for traditional straddle carrier or RTG-TT operations. Although robotisation means that one has to cope with start-up problems – the newly built container terminal in Hamburg (CTA) is a good example – terminal operators cannot longer ignore the benefits of automation, simply because the benefits are making the difference between success and failure.

Until now we have seen two types of robotisation; first the automation of the yard by means of rail mounted gantries (RMGs) and secondly the automation of the horizontal transportation by means of automated guided vehicles (AGVs). Although cost-efficient, the combination of RMGs and AGVs is not considered to

be flexible nor highly productive. However, these considerations are not supported by facts, or simulation results. Much is expected from automated straddle carriers, either 1 over 1 or 1 over 0. These automated lifting vehicles (ALV) have the same characteristics as straddle carriers concerning transportation, but do not fulfil the stacking functionality. The stacking is done by RMGs, which leads to a much denser stack than possible with straddle carriers.

The question that will be answered in this paper is whether high expectations regarding ALVs can be justified or not, and we point out the differences and similarities between its main competitor, the AGV. Hereby, we dare to question the general understanding that AGVs cannot be productive. As a benchmark for automation we take the manned shuttle carrier as a reference.

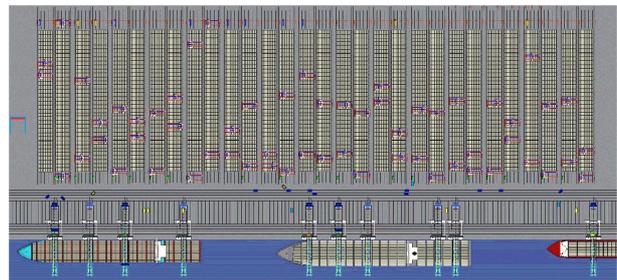


Figure 1: Example layout (true-to-scale) with perpendicular stack modules, operated by dual RMGs and a horizontal transportation performed by either AGVs, ALVs or manned SCs.

The analysis we present in this paper consists of three parts. First we present the results from a comparison of the productivity by means of dynamic simulation. Then, we present the costs of the three alternatives for the horizontal transportation system, and finally we combine the productivity and cost results with a number of other aspects in a multi-criteria analysis.

We take here a fantasy terminal configuration that is, however, representative for the operations in a number of ports in the Le Havre – Hamburg range. The terminal should be capable to handle 2,400,000 containers per year when fully extended. The percentage transshipment is low (<10%) and there is a huge amount of rail moves.

## QUANTITATIVE COMPARISON

### Characteristics of the operation at a robotised terminal

During corridor chat, robotised terminals are labelled “underperforming”. Of course, it is generally accepted that the operational costs are significantly lower, but the perceived lack of waterside productivity, together with a high project risk and an inflexible operation, lead to aversion to automation. Then, there are also the unions, which oppose against the reduction of jobs.

Regarding the current state-of-the-art of robotised terminals, two questions arise: First, what are the key measures that have been taken to get the robotised terminal at a similar productivity level as there manned equivalents? Secondly, we question the perception that robotised terminals are underperforming in comparison with terminals that face similar conditions, such as the calling pattern, the demands from the shipping lines, the labour conditions. We will first deal with the first question, because there are reasons that the productivity at the current robotised terminal is lower than aimed for, or actually attainable.

### Improvement measures to robotised terminals

The terminals at ECT (17 QCs, 77 stacks, 150 AGVs) are the example of a robotised terminal. The recent successor at Altenwerder is in principle a similar concept. Both terminals have carried (CTA during the design phase, ECT during the operation) through a number of productivity improving measures to overcome the initial pitfalls of the ECT concept. What are the pitfalls that should be avoided whenever possible? In the following table, we summarise a number of these improvement measures, as they have been tackled in recent improvement projects at ECT and within the terminal design at CTA.

### Robotisation: mixture of automation and human control

Robotised systems are in the contrary to manually operated systems well predictable in their behaviour. Machines can reproduce tasks with the preciseness of a Swiss watch, whereas humans tend to vary their behaviour. However, the currently realised automated terminals show a mixture of automated tasks and manually operated tasks. The quay crane driver, the reefer man, or the truck driver serving the rail terminal, are examples of interactions between automated and manual operations. This is one of the reasons that even the operation at an automated terminal is less predictable than one would expect.

### Quality of information

A second reason for the occurrence of stochastic behaviour is the lack of quality of the information available. Information about load lists, container weight, PoD, et cetera. This information changes until the container is loaded onto the vessel. Due to these

changes, the processes are not as deterministic as one would like.

Table 1: differences ECT and CTA

Current system	Future system (new terminal)
<b>Operation at QC</b> <ul style="list-style-type: none"> <li>– AGV operation in gauge of QC</li> <li>– Single hoist QC (no buffer)</li> <li>– No dynamic replanning based on actual events</li> <li>– Single cycling of QC</li> <li>– Single lifting of QC</li> <li>– Fixed loading sequence</li> </ul>	<b>Operation at QC</b> <ul style="list-style-type: none"> <li>– AGV operation in backreach of QC</li> <li>– Dual hoist QC (platform in QC)</li> <li>– Continuous replanning</li> <li>– Dual cycling of QC</li> <li>– Twin lifting of QC</li> <li>– Flexible loading sequence</li> </ul>
<b>AGV operation</b> <ul style="list-style-type: none"> <li>– Static topology</li> <li>– No buffering at quay</li> <li>– Limited freedom of movements</li> <li>– Static claim areas</li> <li>– Maximum speed 3 m/s</li> <li>– Single carry</li> <li>– Long start-up times</li> <li>– No flexibility in case of disturbances</li> </ul>	<b>AGV operation</b> <ul style="list-style-type: none"> <li>– Dynamic topology</li> <li>– Buffering at quay</li> <li>– Tuned movements for specific locations</li> <li>– Speed dependent claim areas</li> <li>– Maximum speed 6 m/s</li> <li>– Twin carry</li> <li>– Early messaging to reduce start-up delay</li> <li>– Dynamic rerouting in case of disturbances</li> </ul>
<b>Operation ASC</b> <ul style="list-style-type: none"> <li>– No cycling of ASC</li> <li>– Single ASC on stack → no redundancy nor flexibility</li> <li>– No preplanned move before AGV arrives</li> <li>– No integration between stowage planning and yard planning</li> </ul>	<b>Operation ASC</b> <ul style="list-style-type: none"> <li>– Dynamic job selection and co-operation between ASCs</li> <li>– Two ASCs on stack</li> <li>– Integrated planning of QC – AGV – ASC – landside move</li> <li>– Interaction between stowage planning and yard planning</li> </ul>
<b>Operation at landside</b> <ul style="list-style-type: none"> <li>– Intermediary step with straddle carrier</li> <li>– No pre-notification of truck arrival</li> </ul>	<b>Operation at landside</b> <ul style="list-style-type: none"> <li>– Direct operation under ASC</li> <li>– Pre-notification at gate of truck arrival</li> </ul>

### Equipment failure

The final reason for disturbances is the way fleets of automated vehicles, such as AGVs or ALVs, are controlled is not according to a pre-planning, which means that the real-time interaction affect the reliability of the execution of moves. Also influencing the behaviour of the automated vehicles is the failure rate; not only mechanical failure but software failures as well. These disturbances disrupt the operation and demand for quick intervention.

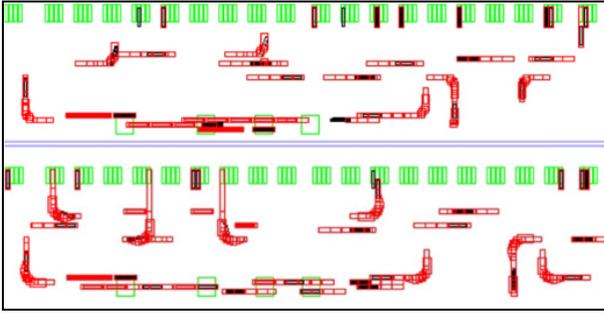


Figure 2: Visualisation from the (simulated) control system of the collision manager. The green rectangles are the transfer locations; above the transfer points at the RMG, below the transfer points at the QC. The red rectangles are the space reserved for each AGV. The black rectangles are the AGVs. The difference between the two pictures is the curve-behaviour of the AGVs; the above picture shows optimised space reservation behaviour, below the standard behaviour that causes more interference between AGVs.

### The (mis)perception of underperformance

Most traditional operators and shipping lines point at the robotised terminals and address them as underperforming. This perception originates from the initial start-up problems that robotised terminal have; more than manually operated terminals. However, it is our opinion – given the actual numbers in the range Le Havre – Hamburg – that after the start-up phase is more perception than fact, that the automated terminal perform less than their manually operated competitors. All numbers of gross quay crane productivity vary between 22 and 30 containers per hour. Of course, the magnificent productivities that are registered in Asia and the US are not met, but they are not met anywhere in Europe! Furthermore, when benchmarking those productivities (gross often above 40 containers per hour) the specific local conditions have to be accounted for. Smaller terminals, lower berth occupancy, no acceptance of late arrivals, gate closing times during the night; all circumstances that enable terminals to prepare for the upcoming operation. In most terminals in Europe, these conditions are not met.

Therefore, one should be careful in writing robotisation off because of the perceived inability to deliver appropriate service. When the conditions are right, the information is at a sufficient level, robotised terminal may even perform better than their manned equivalents, because the cost of preparation (i.e. housekeeping) is so much lower, and therefore, more easily done. Therefore, under similar conditions, three alternative systems will be compared by means of simulation.

### Modelling the operation at a robotised terminal

The first step of our quantitative analysis consists of dynamic simulation of the three waterside handling systems. In order to analyse these systems in detail, we reckon that a simulation model is required that contains a fair depiction of the process control system, since it

affects the performance of the transportation system and the yard handling system. It also deals with the stochastics mentioned earlier; by means of planning and real-time re-planning, taking the latest information into consideration. We state that *any automated system can be made or broken by its process control system*.

The following rules should at least be considered when comparing a system with AGVs and ALVs to express the different characteristics of the vehicles:

- Order planning (to estimate the handshake moments at RMG and QC).
- Job assignment (which vehicle or crane will perform which transportation or stacking job?).
- Transfer point management (assignment interchange points at the QC and RMG to AGVs or ALVs).
- The transfer protocol at the transfer points: interference between RMG and ALV/SC (traffic lights!) and between QC and ALV/SC. Also the landside transfer protocol (managed by remote operators) is relevant for the waterside operation.
- Sequence control (managing the sequence of containers under the QC).
- Collision and deadlock avoidance (making sure that the vehicles do not collide; AGVs use less space than ALVs).
- Dynamic routing – layout management (determining realistic and efficient routes for the vehicles).
- Dual cycling procedures (under the QC).

We developed a simulation model that contains these rules. The entire set of rules is very close to a real implementation in a process control system (PCS).

Besides a valid representation of a PCS, the simulation model had to contain a valid representation of the equipment, i.e. the QCs, the AGVs, the ALVs, and the RMGs. In close co-operation with equipment suppliers, these equipment models were developed and verified. Validation has been done by means of animation (partly 3D, in the case of the RMG) and by means of statistical analysis of the results.

### SIMULATION SCENARIOS

In order to make a sound comparison we used two types of scenarios. The first scenario can be classified as a peak scenario, representing an operation that is likely to occur during less than 5% of the time (which means approximately 400 hours per year). Based on this scenario, we can determine how much equipment is required to meet the productivity requirements. The second scenario can be classified as a busy but regular operation. Based on this scenario, we can determine which type of system performs better. Of course, this can also be determined based on the first scenario, but this is a seldom case; we prefer to assess the quality of the handling system based on the average situation. The two scenarios are defined as follows:

- The peak scenario consists of a demand on the waterside of 17 dual hoist<sup>1</sup> deep-sea cranes and 3 single hoist barge cranes. The landside load – consisting of trucks serving the rail terminal and hauliers – is 455 moves per hour (mph). The stack filling rate is assumed to be an initial 80%. The filling rate, however, will vary throughout the simulation run as a result of the ongoing operation.
- The average scenario consists of a demand on the waterside of 9 dual hoist deep-sea cranes and 4 single hoist barge cranes. The landside load – consisting of trucks serving the rail terminal and hauliers – is 270 moves per hour (mph). The stack-filling rate is assumed to be an initial 70%.

The main output of the simulation consists of the following parameters:

- Waterside productivity level in moves per hour (mph).
- Landside service time of trucks on the interchange points in minutes.
- Equipment productivity on water- and landside, respectively of the transportation vehicles in moves per hour (mph), and the RMGs in mph.

All results will be gathered for various amounts of equipment. More detailed results can be acquired, but are not relevant for the final decision-making.

## RESULTS

As is shown in Figure 3 and Figure 4, the productivity of the three systems, manually operated shuttle carriers (SC), ALVs and AGVs reaches almost the same level. However, the productivities are reached with different amounts of equipment, with a slight benefit for the manually operated shuttle carriers in the peak scenario, and a similar benefit for the AGVs in the average scenario. A general rule could be that for the same performance, one needs either 3 shuttle carriers, or 4 ALVs or 5.5 AGVs per QC.

We also found that the dual-RMGs are perfectly capable of serving both the waterside end and landside end with an acceptable service level (average service on the landside 10 minutes). However, to reach a productivity level of approximately 24 (productive) moves per hour per stack module, the RMGs have to co-operate with each other. This means that in busy times on the waterside, the landside RMG has to support the waterside RMG and vice versa. This support consists of bringing export containers closer to the waterside (pre-positioning) or executing shuffling moves for the other

<sup>1</sup> The quay cranes are similar to the design at CTA, which means that they have a platform within the crane where the removal of the semi-automated twistlocks is done. The platform has two spaces for containers. The landside hoist is automated; the waterside hoist is manually operated.

crane. Furthermore, it appeared to be important that the RMGs have a large enough look-ahead, to be able to cope with peaks. Doing so enables the RMG to execute shuffles in advance, so that the productive move can be performed faster at the time a peak occurs.

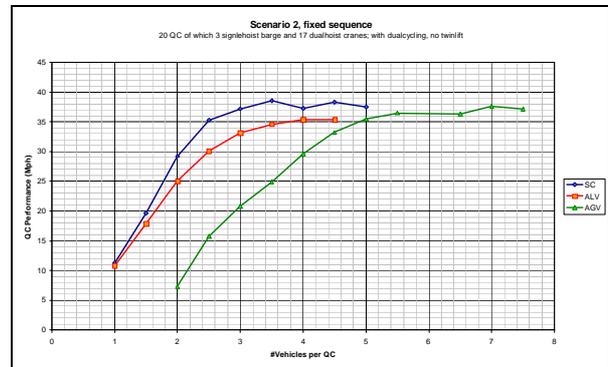


Figure 3: Waterside productivity in peak scenario; on the horizontal axis the number of transportation vehicles is depicted; on the vertical axis the realised net QC productivity in bx/h.

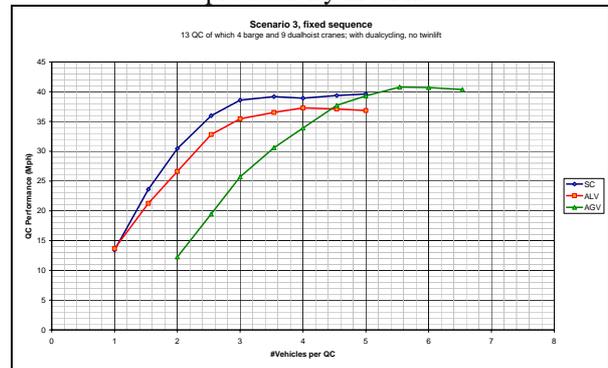


Figure 4: Waterside productivity in average scenario

A critical point in all three handling systems is caused by the limited handling capacity of the RMGs. Because they can only serve over the top ends of the stack modules, the handling capacity per stack is limited in comparison with for instance an RTG or straddle carrier operation, in which it is possible to assign more equipment to the same stack to increase the handling capacity. During loading of the vessel (or loading trucks) peaks that exceed the handling capacity occur in the stack-handling load, which causes delay in the vessel loading process. Because principally all QCs can be loaded from a certain stack, this can influence the service to many QCs at the same time. This effect can be reduced by category loading – exchanging containers with similar characteristics – or flexible loading – real-time changes in the vessel stowage plan. These measures can lead to a better spread over the stack modules in time and thus to an improvement of the QC productivity (up to 15%), but heavily depend on the co-operation of shipping lines and captains.

## COST COMPARISON (see table 2)

When productivity and service levels are one side of the picture, then investment cost and operational cost are

the other side. Therefore, to complete the comparison, we developed a cost model consisting of the main cost drivers of the waterside transportation systems. Here, we leave the other system components out of the comparison, because they are to a very high degree equal. Of course, there are differences, for instance at the transfer points of QC and RMG, but the related costs are considered minor compared to the operational cost differences. Further examination into these aspects will provide more insight.

The cost comparison is built on three components:

- Personnel cost involved with the manning of the SCs.
- Cost of capital involved with the investment in transportation equipment.
- Cost for maintenance and repair of the vehicles

In table 2, the cost calculation is shown. Note that only the cost involved with the waterside transportation system are included. The costs are calculated per QC. In order to determine the operating hours of the transportation vehicles, we start with the planned annual productivity of the QC, i.e. 100,000 containers. With an assumed gross productivity over the year of 35 containers per hour, this leads to 2,857 operating hours per QC. The number of vehicles required originates from the simulation; we took the required number of vehicles so that in the peak scenario a net productivity of 35 bx/h could be achieved, which means 2.5 manned SC/QC, 4 ALVs per QC and 5 AGVs per QC. These vehicles are assumed to deliver in an average scenario 35 bx/h gross.

### **Personnel cost SC manning**

Since the QCs are served by 2.5 SCs on average 2.5 times as many operating hours for the manned SCs are required, i.e. 7,143. However, due to working shifts, the actual number of hours that the SCs are manned will be higher, approximately 10% higher, leading to the 7,857 SC manning hours per year per QC. For this approximately 7 people are required (based on the assumption of 1,200 working hours per year), which cost around 60,000 Euro in North-Western Europe and at least 100,000 Euro in the United States. This leads to the item “variable cost personnel”. For the unmanned vehicles no people are needed that are not needed in the manned situation, e.g. the process control operators have to be present in both cases.

### **Cost of capital**

The second cost component is the depreciation cost (linear depreciation over the life cycle; i.e. 10% per year) of the investment combined with the average interest (0.5 x common interest level of 8%).

### **Maintenance and other operational cost**

The third component is the cost for maintenance, repair, and other variable costs (mainly fuel). The operation with AGVs is assumed to be cheaper regarding

maintenance and repair, because of the absence of a hoisting machine and spreader. Furthermore, the automated operation is assumed to be safer (less accidents) than with a driver.

### **Results**

The three components add up to the total variable costs per QC per year. Divided by the number of container lifts, the cost per container move results. As a result, the AGV is by far the cheapest per container move; even with the high price of an AGV. The ALV is a good second (however, it depends on the actual prices when this vehicle is brought onto the market) with approximately 1 Euro per container move more (for a terminal like this one, this means 2,400,000 Euro on a yearly basis!). The manned SC is by far the most expensive operation, also with European labour costs. In order to reach the cost per container move of an AGV, the labour cost should sink to less than 15,000 Euro per year (12 Euro per hour).

Table 2: Cost comparison of waterside transportation systems

Assumptions	Calculation	manual SC (1 over 1)		ALV (1 over 1)		AGV	
		personnel high	personnel med.	investment high	investment low	investment high	investment low
<b>Quay crane</b>							
A1	Production QC	bx/year	100,000	100,000	100,000	100,000	100,000
A2	Gross productivity QC	bx/h	35	35	35	35	35
A3	Gross working hours QC	per year	2,857	2,857	2,857	2,857	2,857
<b>Vehicles</b>							
B1	Operational vehicles/QC	stack -> quay crane	2.5	2.5	4	5	5
B2	Additional vehicles/QC	maintenance & repair	0.5	0.5	0.5	0.5	0.5
B3	Total vehicles/QC		3.0	3.0	4.5	5.5	5.5
B4	Gross working hours SC	per year	7,143	7,143	11,429	14,286	14,286
B5	Investment	cost per vehicle	€425,000	€425,000	€550,000	€350,000	€300,000
B6	Depreciation	number of years	10	10	10	10	10
B7	Var costs eqp	% of investment	7%	7%	6%	5%	5%
B8	Interest	% of investment	4%	4%	4%	4%	4%
<b>Manning vehicles</b>							
C1	Operational hours / QC	shift effect: 10%	7,857	7,857			
C2	Manning	fte (1200h = 1 fte)	7	7			
C3	Personnel costs	cost per fte per year	€100,000	€60,000			
<b>Costs vehicles</b>							
D1	Depreciation	vehicle investment	€127,500	€127,500	€247,500	€192,500	€165,000
D2	Var cost vehicle	per QC	€140,250	€140,250	€247,500	€191,250	€148,500
D3	Var cost personnel	per QC	€700,000	€420,000	na	na	na
<b>Totals</b>							
E1	Var costs total	per QC	€967,750	€687,750	€495,000	€382,500	€313,500
E2	Var costs / bx		€9.68	€6.88	€4.95	€3.83	€3.14
E3	Investment vehicle	per QC	€1,275,000	€1,275,000	€2,475,000	€1,912,500	€1,650,000

## QUALITATIVE COMPARISON

Besides productivity and costs, there are a number of other aspects that should be considered when investing in equipment. These aspects, completed with productivity and costs are presented in table 3. The first aspect is the risk that is inherent to the type of operation. It covers the degree to which the equipment is proven, and the complexity of the process control software. The second aspect is the complexity of the operation, which covers issues like sensitivity for breakdown, redundancy of equipment, degree of decoupling within the operation, and feasibility to transform into manual operation. Other aspects of importance, such as environmental aspects are not taken into consideration because they are assumed to be equal for all waterside transportation systems. Given the specific characteristics of the three concepts, ALV, SC and AGV, we come to the following assessment:

Table 3: MCA of alternatives waterside transportation systems

	AGV	ALV	Manned SC
Operational cost	++	+	--
Investment cost	0	-	++
Productivity	0	0	0
Project risk	0	-	0
System complexity	-	0	0
<b>Unweighed sum</b>	<b>+</b>	<b>0</b>	<b>0</b>

Cost and investment of the three systems have been discussed already, as is the productivity. Because the cost calculation of performed at a equal productivity level, there are no difference here.

The project risk is a theme of increasing importance, and extremely relevant in the case of automation. First due to the increasing scale of terminals, the financial risk is increasing. Secondly, due to the automation and the tendency to apply RMGs rather than straddles carriers or RTG, increases the investment volume, which increases the risk as well. Thirdly, the dependency on software affects the project risk in a negative way; the commissioning of manually operated terminal is simpler than of an automated terminal, mainly due to software problems. The risk with manned SCs lies in the connection to the automated RMGs; the link between a manual operation and an automated one, can better be avoided and therefore the risk is at a similar level as with the AGV system. This system is proven at ECT and CTA, where most initial complexities have been solved. However, this is not the case for ALVs, which cannot be considered as proven technology. Although there is an operation with automated straddle carriers in Brisbane, this cannot be compared to the dense operations at ECT or CTA. Especially the collision avoidance in this application is too simple for a dense operation. Furthermore, a machine that has to pick-up a container by itself is more difficult to automate than a machine that does not more than drive from A to B, as the AGV does.

- Although important, the risk of automation is decreasing because of a number of developments:
- The supplier market of robotised solutions is steady growing with the increased interest for this kind of solution.
- There is more and more experience with the design, realisation and commissioning of robotised terminals.
- The software is increasingly mature, and the big terminal control software providers are now developing components for controlling automated operations as well.
- A simulation based approach, using similar models from initial design to final commissioning increases insight (also for non- experts) and shortens the feedback loop.

The final criterion is addressed as system complexity, covering the flexibility of the operation, the redundancy within the concept, and the sensitivity to disturbances and breakdowns. Here, the manually operated concept is clearly beneficial compared to the automated competitors: due the fact that late-minute changes still can be coped with by the drivers, the flexibility is high and the vulnerability to disturbances of relatively low. Of course, all concepts depend on the service by the RMGs and have no possibility to access containers in the yard themselves. However, during the transportation process, the lifting vehicles can easier take over jobs of vehicles with a failure. The container is put down and taken over by another vehicle. In the case of AGVs, this is not possible.

In conclusion, one can say that each concept has its pro's and con's and the final assessment depends on the weight of certain criteria. In most terminals the cost per move and the investment volume are crucial, but aspects such as environmental impact are of increasing importance.

## Conclusion

Is the 1 over 1 automated lifting vehicle – or automated shuttle carrier - the productivity bringer when compared to AGVs? No. Nor will it lower the investment costs. This simple answer can be made after a detailed comparison between an RMG-AGV and an RMG-ALV operation. Although the latter combination can do with less equipment because of the decoupling between RMG operation and ALV operation, the cost advantages of AGVs over ALVs compensates the bigger amount of equipment.

In addition, we have to say that the AGV is *proven technology*, whereas the ALV is *not*. It is a vehicle more complicated to automate, and certainly more expensive regarding operating costs. Finally, since the spreader is the most vulnerable component, we should try to reduce the number instead of increasing it.

When looking at the state-of-the-art layout with twin RMGs and AGVs, we have to conclude that the

system's potential is hindered by the inflexibility in the operation and the poor information available to the terminal. To utilise this concept's potential to a maximum, shipping lines should agree with flexible loading and provide terminals with accurate information well in advance. Only then, we see the possibility to increase productivity on a constant basis to numbers in the range of 45 to 50 lifts per hour.

When the two automated concepts are compared to the manned 1 over 1 straddle carrier, the conclusion must be that although the project risk may be higher, the overall cost of the automated alternatives is significantly lower than of the manned alternative. With a difference in cost per move of approximately 3 Euro, the additional investment pays back after 100,000 QC moves. Therefore, our conclusion is that robotisation pays off and is the right concept for the future.

### Biography

**Yvo A. Saanen** (MSc in Systems Engineering) works as principal consultant at TBA, a leading simulation consultancy company in The Netherlands and supports ports all over the world in their design process by means of simulation. He is the main architect behind the TBA port simulation suite that enables terminal operators, shipping lines and integrators to design, and optimise their terminal and plan their operation in a more efficient way. Besides Yvo Saanen is finishing his PhD on a design approach for robotised maritime container terminals. The PhD is performed at Delft University of Technology.

**Jeroen van Meel** is project manager for the realization and construction of the Euromax container terminal in Rotterdam, a joint venture between P&O Nedlloyd and ECT. He has been head of department of projects and equipment acquisition and ICT manager robotized terminals for the Delta Terminal of ECT, Rotterdam. He has a degree in Information Science and a PhD in Policy

Analysis and Management of the University of Technology Delft.

**Alexander Verbraeck** is a research professor at the R.H. Smith School of Business, University of Maryland, College Park, U.S.A. He is also associate professor at the Systems Engineering section of the Faculty of Technology, Policy and Management of Delft University of Technology. Alexander Verbraeck has a master degree in mathematics and a PhD in information systems. His research focuses on the application of simulation in all kinds of business sectors, as well on innovation of the simulation techniques as well.

### REFERENCES

- Drewry Shipping Consult, Global Container Terminals, Profit; performance and prospects, United Kingdom, October 2002
- Ocean Shipping Consultants, Containerization in North Europe to 2015, United Kingdom, London, 2002
- Rijsenbrij J.C., Double or Quit? New concepts in Terminal Design, Terminal Operating Conference (TOC) 2002 Europe, 11th June – 13th June 2002, Antwerpen, Belgium
- Saanen Y.A. and U. Franzke (2000), Preparing simulations for more advanced purposes: design of an automated container terminal. In: Kai Mertins, Markus Rabe (eds.). The New Simulation in Production and Logistics – Prospects, Views and Attitudes. Proceedings 9. ASIM Steenken D., Optimising Straddle Carrier Operations to Achieve High Productivity, Terminal Operating Conference (TOC) 2002 Europe, 11th June – 13th June 2002, Antwerpen, Belgium
- Fachtagung "Simulation in Produktion und Logistik". Berlin, Germany, 8-9 March 2000. IPK, Berlin. ISBN 3-8167-5537-2. p. 233-244.
- Saanen Y.A. (2001), Improving Complex Control System Development using Simulation, AAPA Oakland, 2001.
- Saanen Y.A. and Dobner, M. (2002), Robotics and Terminal Design; TOC Antwerp 2002. Balci O. and R.G. Sargent. 1983. "Validation of Multivariate Response Trace-Driven Simulation Models". In *Performance 1983*, A.K. Agrawalla and S.K. Tripathi (Eds.). North-Holland, Amsterdam, 309-323.

# AI TECHNIQUES FOR THE IMPLEMENTATION OF NEW ORGANIZATIONAL STRUCTURES IN THE RETAIL INDUSTRY

Alessandra Orsoni  
Kingston University  
Kingston upon Thames, Surrey KT1 1LQ, UK  
E-mail: aorsoni@alum.mit.edu

**KEY WORDS:** Island-based Organizations, Decision Support Systems, Genetic Algorithms, Logistic Management, Optimization Techniques.

## ABSTRACT

This paper introduces a hybrid decision support system (DSS) for the logistic management of island-based organizational structures. The main benefits expected from the implementation of this innovative organizational structure are improved workers satisfaction, productivity, and quality of service to the customer, and reduced risks of manpower shortage. The DSS is based on a hybrid architecture combining the propositive capabilities of genetic algorithms (GAs) and the scenario-testing strength of simulation for the iterative optimization of resources allocation to dynamic workloads. The design and the customization of the DSS are discussed in the paper with reference to an example application relevant to a large Italian retail chain. The application consists of a feasibility study for the implementation of the island-based concepts in a prototype retail store. In this context the DSS optimizes the allocation of the current pool of resources, in terms of quality of service to the customers, workers' job satisfaction and resources cost. In addition, the DSS provides guidelines and recommendations for future training and hiring.

## INTRODUCTION

The concept of island-based organizational structures is a new approach to business organization and management developed and first introduced in France by a small consulting firm, Consilium 2000, operating in the field of human resources management. Islands are work groups built on the basis of complementary workers' skills and schedule requirements.

Some of the readers may be familiar with other types of organizational structures based on workgroups. Cell-based manufacturing systems (Prickett, 1994; Slack, 1988; Schonberger, 1986), for instance, are well-known exmples of organsations based on workgroups. What set manufacturing cells and islands apart are the purposes of their implementation and, consequently, their grouping criteria. Manufacturing cells are typically introduced to achieve production related objectives such as reducing lead time and Work In Progress (WIP) by increasing flexibility (Slack, 1988) and streamlining

production (Schonberger, 1986; Prickett, 1994). Islands, instead, address the needs of labour intensive business environments with the objectives of reducing the cost of resources overtime, increasing resources productivity, and reducing the risks of manpower shortage, by better meeting workers schedule and job preferences. To a large extent, islands are self managed in that workers get to choose their working hours and duties on a bi-weekly basis as long as the overall work-plan for the island is feasible and covers the entire workload allocated to the island for that (the 2-week) period.

While the concept of islands is fully developed in theory and a prototype island-based hyper-market is now operating in France, the implementation of this business structure in existing organizations represents a totally different challenge as it raises the issues of defining the islands around rather specialized resources and of deciding the most effective way to expand their skills and competences in order to meet the poly-functional needs of each island. The major difference between the two situations is that the prototype French hyper-market was created around the concept of islands and, thus, specifically designed to operate according to this innovative business structure, meaning that the personnel was hired and immediately cross-trained for these purposes. The optimization of resource allocation to islands in a traditional work environment is a complex example of logistic management problem, which involves the definition of the islands in terms of types and amounts of workload, and mix of allocated resources, as dynamic variables following the patterns of customers' flow in the store. Several are the performance drivers, which are to be decided upon in the formulation of a feasible and cost-effective allocation plan. At the whole service level the relevant performance measures include resources cost, productivity, and customer service, which are driven by other parameters including workers job satisfaction, average customer queuing time for different services, and product display/availability for purchase. The first and most important step to be competed prior to the design and implementation of the islands, is a thorough assessment of the workloads in the different areas of competence, with their seasonal and daily fluctuations. Second it is necessary to clearly assess the current skills and competences of the available resources, their potential and willingness to undergo training and learn new skills, matching them with the identified types of workloads. Third it is necessary to understand workers

priorities in terms of work schedule and preferences in terms of duties. The efficient allocation of a specific resource to a job on a given work-shift is highly dependent not only on the expected performance of such an allocation, but also on the implications that this choice may have on the performance of all the remaining allocations. These interdependencies are mainly due to the fact that the number of available resources per skill/competence area is limited and, therefore, the allocation of a first-choice resource (e.g. the most efficient one) to a job may leave second or third choice resources available for allocation to other duties. In such a sense the optimization of the process at the individual job level is likely to produce a resource allocation plan which is sub-optimal from the perspective of the service as a whole (Orsoni, 2000). Simulation-based testing is the only approach by which each resource allocation scenario can be thoroughly tested taking into account primary, secondary, and tertiary impacts of each allocation choice on the performance of the whole service over the 2-week period. However, when looking at business organizations counting over 50 employees, exhaustive testing, on all possible combinations of resources allocations to jobs and workshifts, becomes computationally intensive and time consuming if handled by simulation alone. The approach proposed in this paper employs AI techniques based on Genetic Algorithms (GAs) for preliminary screening of the resource allocation scenarios to be subsequently tested through simulation. The iteration of the procedure feeding the simulation outputs back into the GAs effectively leads to the identification of an optimized solution for the entire service which is the result of best trade-offs among workloads rather than workload-based optimization

## **BACKGROUND**

The integration of AI and simulation techniques constitutes a powerful approach that the authors have successfully implemented in the development of decision support systems (DSSs) for complex industrial applications (Mosca et al, 1998; Bruzzone et al, 2001). Relevant applications to supply chain management rely on the anticipatory capabilities of artificial neural networks (ANNs) for demand forecasting (Giribone et al, 1997; Giribone and Bruzzone, 1997), for resources/materials usage prediction (Giribone and Bruzzone, 1998), and for the definition of projected scenarios providing the boundary conditions for simulation-based testing of alternative management policies. Other applications (Bruzzone et al., 2002; Bruzzone and Signorile, 1998) use AI techniques, based on either ANNs or genetic algorithms (GAs), as DSS planning/optimization modules. Two major factors drive the choice between ANNs and GAs, these are: the nature of the decision context and the availability of historical data. ANNs are capable of establishing correlations between corresponding sets of input and output parameters even when explicit analytical or

logical formulations may not be found (Padgett and Roppel, 1992; Hillis, 1989). This ability is based on “experience”, which in turn is provided to them by training on extensive sets of historical data (Anderson and Rosenfield, 1988). When applying ANNs to complex decisional problems, adequate precision is difficult to achieve in reasonable learning times (Padgett and Roppel, 1992; Anderson and Rosenfield, 1988): these objectives can only be achieved in situations involving a limited number of input and output parameters, if a large enough set of representative training data is available (Bruzzone et al, 2001). In contrast, GAs are well-suited to address complex combinatorial problems involving multiple variables and do not require preliminary training (Bruzzone et al, 2001; Goldberg, 1989). They iteratively seek performance improvement by trial and error (Bruzzone and Signorile, 1998; Goldberg, 1989; Koza, 1992). At each iteration an “evolutionary step” is made: the least efficient solutions are discarded, while attempts are made to improve the best performing ones by mixing (cross-over) and modification (mutation) (Goldberg, 1989). Both ANNs and GAs may be effectively combined with simulation to address complex optimization problems, however, for the particular application described in this paper, GAs appear better suited because of the lack of historical data on resource allocation, and because of the large number of parameters involved (including the number and types of resources with their specific skills and availability constraints, and the number and types of workloads with their seasonal and daily fluctuations.) The optimization problem examined in this paper is especially complex because a global optimum needs to be found based on multiple interdependent decisions, which cannot be separately optimized: in the island-based organizational structure, the efficiency of the allocation of a particular resource to a workload is reflected in the processing performance of other workloads and, through them, in the performance of the entire service.

## **STRUCTURE OF THE APPROACH**

The DSS consists of a hybrid architecture, which combines discrete event stochastic simulation and Artificial Intelligence (AI) techniques based on Genetic Algorithms (GAs) in the form of two interactive modules (Figure 1). In this architecture the simulation module tests the feasibility and the effectiveness of alternative scenarios of resource allocation to the various jobs and workloads and to the scheduled work shifts with respect to the performance of the entire service on the 2 weeks. Starting from the measures of performance provided by the simulation module on a first set of scenarios, the GA-based optimization module generates new, improved scenarios for further testing. The procedure is iterated until an optimum combination of resources scheduling and job allocation is reached. The major advantage of using GAs in the optimization process is that the search for the optimum solution begins from an entire “population” of scenarios

(Goldberg, 1989) and, thus, from multiple points in the space of the possible solutions, which highly increases the chances of finding the actual optimum, rather than a sub-optimum. In addition, GAs are based on stochastic rather than deterministic rules, which further improves the effectiveness of the search (Goldberg, 1989). In order to define a suitable set of initial scenarios, the optimization tool is first run under the hypothesis of ideal conditions (independent workloads and unlimited number of resources) to determine the most suitable resource for each job-unit/workload (i.e. the resource of first choice based on the skills/competences available). Based on this information, the mix of resources and their allocation are adjusted until the feasibility of the resource schedule is reached in the real conditions (i.e. each workload unit is allocated to an available resource), as represented by the process defined in the simulation tool. The feasibility of the allocation plan is simply the ability to meet customer demand to a user-defined extent (e.g. 90% of customers serviced with wait time under 7 minutes). The performance measures extracted from the simulation tool for the feasible scenario are then used as input to the GAs to generate new scenarios that progressively move the performance of the solution from mere feasibility to maximum job satisfaction, in terms of schedule and duties, compatible with the expected workloads, and minimum cost, until the most suitable solution is found.

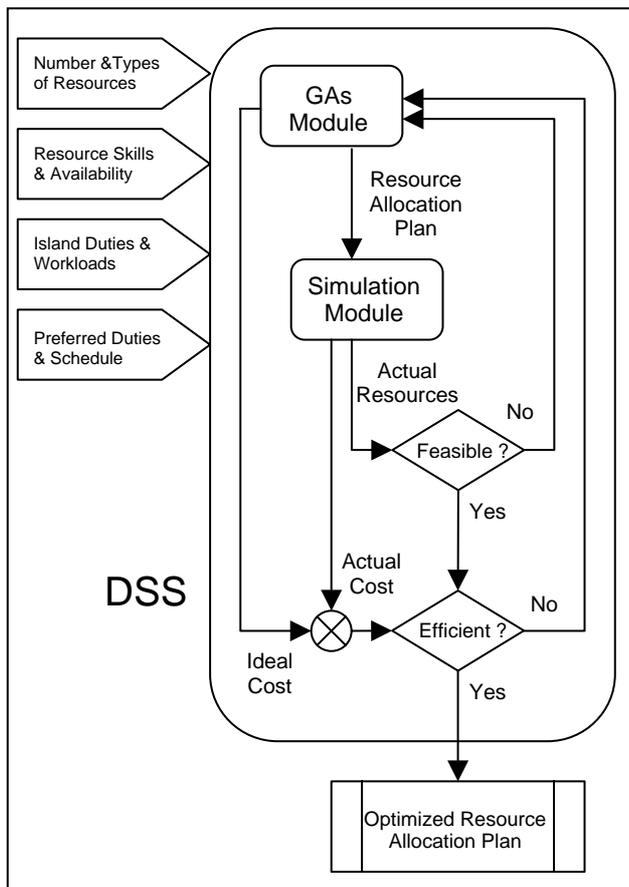


Figure 1: Functional Schematic of the Hybrid DSS

With reference to the case study, the measure of performance chosen as reference for the optimization process combines the percentage of unmet workers schedule and job preferences over the 2-week period and the extra cost to the company in terms of percentage resources overtime. The optimal solution will be the one capable of minimizing this performance measure. In the first instance this measure is taken as a weighed sum of the two performance drivers, assigning the same importance, namely weight, to each one of them.

The analysis conducted using the DSS leads to the identification of the most effective island design for the examined store, and to the formulation of guidelines and recommendations for future training and hiring of resources aimed at achieving the most effective mix.

### THE IMPLEMENTED SIMULATION MODEL

The evaluation of a resources allocation plan with respect to its feasibility and efficiency requires a tool capable of simulating the various activities according to variable resource requirements, availability, and utilization. In order to support this testing phase a discrete-event simulation model was built in the C++ environment. The simulation model is quite simple in nature as it basically handles each service station as a queuing system subject to the stochastic variability of the number of servers available and of the respective service time. These aspects of the service could easily be modeled using standard process simulation packages, such as SIMPROCESS or ARENA, capable of handling queuing systems and stochastic variables. However it was preferred to implement the model in the C++ environment for flexibility and ease of integration with the GAs-based optimisation module.

In the model, the daily activities of the store are represented in terms of product handling and display, product preparation, customer service and cashier's duties across the main product categories of the store. Each of the available resources is characterized with a resource profile containing the suitability (yes/no) for each type of duty/product category and a corresponding skill level, namely a coefficient, capable of influencing the duration of each activity and, thereby, the number of customers serviced and/or the number of product units handled in the reference time unit. Specifically each activity is characterized by a reference triangular distribution representing its baseline time to complete, intended as an averaged distribution across the range of suitable resources. Whenever in the course of the simulation run a value of activity duration is extracted out of the baseline distribution, its value is scaled using the skill factor of the resource currently working, which in turn depends on the number of years of experience on the job as well as on the specific competences of such a resource as indicated in the resource's profile. This is the extent to which the individual resources are personalised in the simulation model: the model does not involve the detailed modeling of human resources in their multiple behavioural aspects, as the worker is only

relevant to the testing of the resources allocation plan proposed by the GAs module in terms of his/her availability and service time. By talking to the store managers it was found that the optimal resolution in resource allocation testing is obtained considering a reference time unit of one hour, as it would not be efficient to consider resources re-allocation for shorter times. Finer scheduling resolution may be considered in the future for the management of store contingencies/emergencies, but these are not accounted for in the study at this stage. Prior studies conducted in the reference store, and additional data collection through their information systems database, enabled the re-construction of the dynamic patterns of customer flows across the different areas of the store. The main sources of information were the cashier's receipts of which both electronic and printed records are available in the store's database. The electronic records of cashier's receipts for the past two years were sorted to extract the number of customers accessing the store with its seasonal and daily fluctuations and the corresponding purchase patterns by product category. This data was then analysed and discussed with the store managers to re-construct reasonable patterns of workload by store area and duty to be used as reference in the simulation model. Seasonal, weekly, and daily patterns were extracted for two reference periods: Summer and Winter, broadly intended, as activities and customer flow patterns vary dramatically between the two periods in relation to tourism and seasonal changes in lifestyle. In the first instance the study did not address special periods such as major holidays, strikes, and other events, as it was mainly focused on the identification of generalized patterns for each one of the two broadly defined seasons. Special attention is paid to the handling of scheduling emergencies, intended as the unavailability of scheduled resources. For each resource an unavailability rate was defined based on cumulative store trends and estimates provided by the managers. For the purposes of the simulation model the resources unavailability rate was defined as:

$$1 - A_v = \frac{MABS}{MTBE + MABS} \quad (1)$$

where:

$A_v$  = resource availability rate

MABS = mean time of absence

MTBE = mean time between emergencies

At the beginning of each simulated work-shift the pool of scheduled resources actually available is updated using the Monte Carlo technique to extract punctual values from the corresponding MABS and MTBE distributions. For the purposes of the current work-shift the duties of the missing resources are shared and reassigned to the suitable resources actually present, with an impact on service efficiency (i.e. customers queuing time). For the purposes of the following work-

shifts, for the entire duration of the resource MABS, a spare resource is introduced and the corresponding overtime costs are computed.

At the end of each simulation run a report file is generated, which records all the relevant simulation details. These include resource availability, percentage utilization and overtime costs incurred and quality of customer service as defined in the previous section (percentage of customers who waited less than 7 minutes for service). This set of information is fed back into the GAs module where it is used to compute a performance index for the given resource allocation plan.

## EXAMPLE APPLICATION

The DSS was applied to perform a feasibility study for the implementation of the island-based organizational structure in the examined context (i.e. the reference store described in the previous sections). In particular the analysis was aimed at highlighting the potential benefits from the introduction of such a business structure, by comparing its performance to the average performance observed with the current organization. For the purposes of this analysis a reasonable extent of resources interchangeability was assumed across product categories within the three major types of activities: product handling and display, product preparation and customer service, while it was assumed, that cashier's duties would be shared among all the available resources who would dedicate approximately 20% of their time to cover cashier's workloads. Three product groups were defined aggregating product categories which have similar requirements in terms of resources skills. Based on such product grouping, six islands were defined as aggregation of duties and workloads, three pertaining to product handling and display (one for each product group) and three pertaining to customer service (again one for each product group). An additional island was defined grouping duties and workloads related to product preparation. This last island is treated separately from the others as only limited interchangeability is allowed for the resources involved in product preparation, because the activities are highly product-specific and require special resource training. In the future it may be possible that resources cross-training will lead to full interchangeability also in this area, but for the time being this option was not considered as it is not of immediate implementation, given the level of specialization of the current resources.

Based on the current pool of resources the GA-based module of the DSS was customized to manipulate binary input/output strings in which "1" bits indicate the presence of given resources for a given scheduling interval and "0" bits indicate the absence of the remaining resources for the same time interval. The objective function for the optimization process was set to be a percentage performance indicator combining the

percentage workers satisfaction, in relation to their preferred duties and schedule, and the percentage of extra costs related to resources overtime. The number of strings for each island is equal to the number of scheduling intervals included in the scheduling horizon. For the purposes of this application the scheduling horizon was set to one day and the scheduling interval was fixed to one hour. The procedure was then iterated for each day in the 2-week period of analysis, as the store manager typically finalize their resource allocation schedules on a bi-weekly basis.

The logical steps for the optimization of the resource allocation schedule are explained in the following. The system is fed with the bi-weekly duties and workloads by island and with the corresponding preferences indicated by the resources. Based on this information the GAs-based module produces a first attempt resource allocation plan selecting the resource of first choice for each duty according to resources skills. Starting from this preliminary resource allocation plan, the GAs-based module proposes a schedule for the workloads of the different islands distributing them over the designated scheduling intervals. The scheduling constraints are chiefly related to the number of equally skilled resources available for island. If the resource of first choice is not available for all workloads on each scheduling interval, the original plan is not feasible and, thus, a resource of second choice is assigned to some of the given workloads. This process is iterated until a feasible solution of minimum cost is found. The resource allocation plan and schedule proposed by the GAs-based module are ideal ones because they are tested for feasibility and cost-effectiveness without considering the impact of resources skills on their productivity in relation to the actual customer flows in the store. The actual feasibility and cost of each solution can be verified through the discrete event stochastic simulation module, which explicitly accounts for these performance drivers. Realistic service times and costs are computed in the simulation model by associating a stochastic duration to each activity, which is further influenced by the specific resource's skill level, and availability.

## APPLICATION RESULTS

The customization of the DSS, as described in the previous section, and its application to the analysis of the potential benefits from the implementation of an island-based organizational structure in the reference store, led to an interesting comparison between the current and the projected performance for the existing and the proposed structure, respectively. Three performance measures were defined for the purposes of this comparison. These include, quality of customer service, workers job satisfaction and overtime cost incurred. The quality of customer service is measured as the percentage of customers waiting less than 7 minutes for service. Worker's job satisfaction is measured in terms of percentage ability of the allocation plan to meet

the workers' preferred work schedule and duties. Finally the impact of resources overtime cost is calculated as a percentage of the ideal cost of the service assuming that none of the resources worked overtime.

Prior to actually using the simulator to test alternative resources allocation plans, an estimate of the number of simulation replications needed per simulated scenario, was obtained using the classic methodology based on the analysis of the temporal evolution of the Mean Square pure Error (MspE). As mentioned in the previous paragraphs of this same section, for the examined application the length of the simulation run is fixed to fourteen days of activity as the actual planning horizon, which store managers typically refer to, is two weeks. The MSpE-based method was then applied to determine the number of replications needed to obtain sensible results out of the simulator or, in other words, the number of 2-week periods that need to be simulated for such purposes. Plotting of the temporal evolution of the MspE for the three designated output variables led to the identification of the optimal number of replications required, which was found to be 7. This may appear a relatively small number even considering the rather specialized and self-contained nature of the application context. However, it should be considered that 9 replications involve the simulation of 126 work-days, which in turn correspond to 2016 time units for scenarios involving two eight-hour work-shifts per day.

Figure 2 summarizes these results. As shown in the figure, the major benefits expected from the introduction of the optimized island-based structure in the examined store are a significant reduction in resource overtime cost (approximately 68 %), a relevant increase in workers job satisfaction (approximately 32%), and a marginal improvement in the quality of service (approximately 11%)

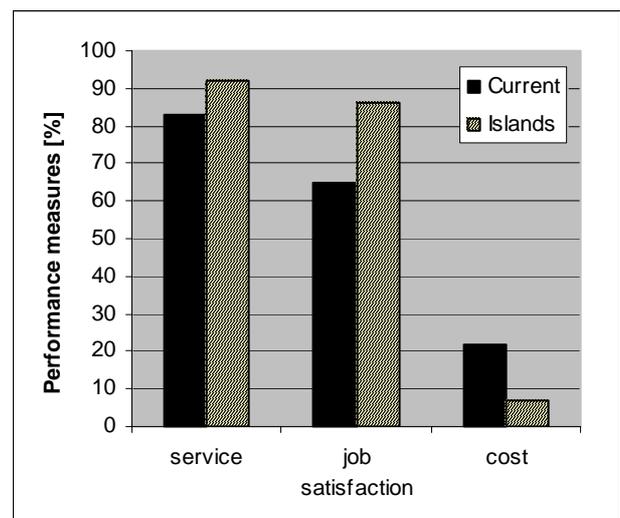


Figure 2: Performance Comparison between the Current and the Optimised Island-Based Structure

The output values shown in the figure correspond to the average value of each output calculated over the 9 simulated replications. The level of confidence associated to such estimates of the real output can be measured through the dispersion of the punctual data around each average value. For this scenario it was found that each individual output would fall within the range of  $\pm 5.8\%$  of the corresponding average: specifically,  $\pm 5.1\%$  for the quality of service,  $\pm 4.6\%$  for job satisfaction, and  $\pm 5.8\%$  for overtime cost.

It is important to observe that these results only represent the marginal improvements that may be achieved at zero cost, by simply revising the allocation criteria for the existing pool of resources. Far more interesting will be the benefits that may be obtained from the implementation of other, more efficient solutions proposed by the GAs module during the optimization process, which are currently not feasible due to the lack of skills/resources. Such solution may be implemented at the cost of resources cross-training and/or new hiring and proper cost-benefit analysis will be conducted in the course of further research to provide management with more effective recommendations for the successful implementation of the island-based structure.

## CONCLUSION

The paper presented a hybrid decision support system (DSS) to optimize resources allocation in island-based organizational structures. The DSS architecture integrates GA-based optimization and stochastic discrete event simulation to assess the feasibility and the effectiveness of dynamic resource allocation plans. The paper also illustrated the application of the DSS to the conversion of the existing organization of a reference retail store to an island-based structure. The customization of the tool for the example application, led to an operative definition of islands, based on the actual types of workload and skill requirements, and to the formalization of an improved resource allocation methodology capable of best meeting workers preferred schedules, while minimizing service cost. The experimental results for this application demonstrate the robustness of the methodology and its applicability to a wide range of business organization and logistic management problems.

## REFERENCES

Anderson, J. and E. Rosenfeld. 1988. *Neurocomputing - Foundation of Research*. MIT Press, Cambridge, MA.

Bruzzone, A.G.; R. Mosca; A. Orsoni,; and R. Revetria. 2001. "Forecasts Modelling in Industrial Applications Based on AI Techniques", In *Proceedings of CASYS2001* (Liege, Belgium).

Bruzzone, A.G.; R. Mosca; R. Revetria; and A. Orsoni. 2002. "System Architecture for Integrated Fleet Management: Advanced Decision Support in the

Logistics of Diversified and Geographically Distributed Chemical Processing". In *Proceedings of the 2002 Conference on AI, Simulation and Planning in High Autonomy Systems* (Lisbon, Portugal).

Bruzzone, A.G. and R. Signorile. 1998. "Simulation and GAs for Ship Planning and Yard Layout." *SIMULATION*, 71(2), 74-83.

Giribone, P.; A.G. Bruzzone; M. Antonetti; and G. Siciliano. 1997. "Modelling Innovative Energy Management Techniques in Telecommunication Stations Through the Application of Neural Models". In *Proceedings of the 1st World Congress on Systems Simulation* (Singapore).

Giribone, P. and A.G. Bruzzone. 1997. "Artificial Neural Networks as Adaptive Support for the Thermal Control of Industrial Buildings." *International Journal of Power and Energy Systems*, 19(1), 75-78.

Giribone, P. and A.G. Bruzzone. 1998. "Artificial Neural Networks as Meta-Modelling Techniques for Car Component Industrial Production". In *Proceedings of Neurap 98*. (Marseilles, France).

Goldberg, D.E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA.

Hillis, D.W. 1989. *The connection machine*. MIT Press, Cambridge, MA.

Koza, J.R. 1992. *On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA.

Mosca, R., P.Giribone, A.G.Bruzzone, A.Orsoni, and S. Sadowski, 1998. "Evaluation and Analysis by Simulation of a Production Line Model Built with Back-Propagation Neural Networks, *International Journal of Modelling and Simulation*, 17(2), 72-77.

Orsoni, A. 2000. "Dynamic Process Simulation for the Design of Complex Large-Scale Systems with Respect to the Performance of Multiple Interdependent Production Processes". Doctoral Thesis. MIT, Cambridge, MA.

Padgett, M.L. and T.A. Roppel. 1992. "Neural Networks and Simulation: Modeling for Applications" *Simulation* 58(5).

Prickett, P. 1994. "Cell-Based Manufacturing Systems: Design and Implementation." *International Journal of Operations & Production Management* 14(2). 4-17.

Schonenberg, R. 1986. "World Class Manufacturing: the Lessons of Simplicity Applied." Free Press, New York, NY.

Slack, N. 1988. "Manufacturing Systems Flexibility: an Assessment Procedure." *Journal of Computer-Integrated Manufacturing Systems*. Vol 1. 25-31.

# **SIMULATION IN MILITARY APPLICATIONS**



# AN ACTOR-BASED SIMULATION FOR STUDYING UAV COORDINATION

Myeong-Wuk Jang, Smitha Reddy, Predrag Tomic, Liping Chen, Gul Agha  
Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, IL 61801, USA  
E-mail: { mjang, sreddy1, p-tomic, lchen2, agha } @cs.uiuc.edu

## KEYWORDS

Actor, Simulation, Unmanned Aerial Vehicle (UAV), Coordination.

## ABSTRACT

The effectiveness of Unmanned Aerial Vehicles (UAVs) is being increased to reduce the cost and risk of a mission [Doherty et al. 2000]. Since the advent of small sized but high performance UAVs, the use of a group of UAVs for performing a joint mission is of major interest. However, the development of a UAV is expensive, and a small error in automatic control results in a crash. Therefore, it is useful to develop and verify the coordination behavior of UAVs through software simulation prior to real testing. We describe how an actor-based simulation platform supports distributed simulators, and present three cooperation strategies: self-interested UAVs, sharing-based cooperation, and team-based coordination. Our experimental results show how communication among UAVs improves the overall performance of a collection of UAVs on a joint mission.

## 1. INTRODUCTION

The effectiveness of Unmanned Aerial Vehicles (UAVs) is being increased to reduce the cost and risk of a mission [Doherty et al. 2000]. Some military UAVs, such as the Predator and the Global Hawk, were already used during the wars in Afghanistan and Iraq. Decreasing size of the UAVs and increased demand for more intelligent and autonomous behavior of UAVs are paving the way for consideration of a group of UAVs performing a joint mission. While the cost of UAVs is lower than that of real planes, the development cost of a UAV is still very high, and a small error in automatic control may result in a crash. Therefore, when we consider a large number of UAVs working together, it is necessary to design and verify the behavior of UAVs through software simulation prior to real testing.

Many simulators have been developed as single process simulators. However, a single process simulator has several limitations. First, the performance of a simulation depends on the computational power of one computer. Second, a single process simulator has an extensibility issue when a special component requires its own specific process. For example, if we want to simulate the

coordination behavior of many virtual UAVs with a few real UAVs, each real UAV is working as an independent process. In this kind of simulation, a single process simulator cannot work well. Therefore, a concurrent object-based distributed simulator provides a better simulation environment.

It is commonplace to say that human beings are disposed to cooperate. Biology and ethology show that “kin-altruism” and “reciprocal-altruism” can ground cooperative behavior in animals, such as wolves surrounding prey, termites nest building, and birds flocking. Drawing a parallel, intelligent UAVs that cooperate with one another are of high interest for their ability to search, detect, identify, and handle targets together. The old age tenets of pre-planning and central control have to be reexamined, giving way to the idea of coordinated execution. In this paper, we describe and analyze three different strategies to coordinate tasks among UAVs in a dynamic environment to achieve their goals.

The outline of this paper is as follows. Section 2 sketches a simulation scenario and explains basic concepts about the actor model and the metrics in our simulation. Section 3 describes architecture for our simulation, and three cooperation strategies for a joint mission are presented in Section 4. Section 5 explains our implementation and experimental results. Then, in Section 6 and 7, we discuss related work and our future work. Finally, we conclude this paper with a summary of our simulation framework and our major contributions.

## 2. TERMINOLOGY

### 2.1 UAV Simulation Scenario

Prior to embarking on the architecture of our UAV simulator, we present a simple scenario in order to explain the meaning of basic terms. The application of our simulation is a UAV surveillance mission. For example, 50 UAVs might be launched into a certain area by *Ground Control System* (GCS) to detect targets in the area. For example, *targets* may be civilians to be rescued. In the simulation, UAVs have the autonomy to perform their mission without interaction with the GCS, except during the initial stage when message exchange is necessary to get each UAV started by sending them some default air routes. When UAVs are launched, the UAVs do not have any information about locations of targets. However, each

UAV is equipped with some sensors which can detect objects within the certain range. We assume that all UAVs start from the same location, called an *air base*. Controlling the sequence of takeoffs and landings of UAVs is managed by the control center, called *Air Base System (ABS)*. The main task of a UAV is to detect locations of targets in a mission area and investigate them. Therefore, even though they navigate according to the given air routes, they can change their trajectories to handle targets once they detect those targets. In addition, when UAVs encounter *obstacles*, such as tall towers or airplanes, they should change their air routes to avoid a collision. Therefore, in our UAV simulation, there are five types of important components: Ground Control System (GCS), Air Base System (ABS), Unmanned Aerial Vehicles (UAVs), targets, and obstacles.

## 2.2 Actor

Our UAV simulator is based on the *Actor system*, a concurrent object-based distributed system, and hence, we use the actor model to describe each component in the simulation. An *actor* is a self-contained active object which has its own control thread and communicates with other actors through asynchronous message passing [Agha 1986; Agha et al. 1997]. In addition, an actor can create other actors, just as an object can create other objects. In our UAV simulator, each component, such as a UAV or a target, is implemented as an actor. Since these components in real situations operate concurrently and communicate with one another, their behavior can be captured very well by the actor model. Each software component in the simulation progresses its state independently of the progress of others in response to the environment information gathered either through its own sensor or by communicating with others.

## 2.3 Attractive Force Value and Utility Value

In our UAV simulation, each target has its own value. This value could be interpreted in several different ways. The value might correspond to the number of soldiers or the importance of a building. Also, we can consider this value as the time required to investigate a target by a UAV. For the simplicity of our simulation, we use a single numeric value instead of symbolic information or time information about a target.

In our simulation, we make the following assumptions. A UAV handles only one target at a time, although the UAV holds and manages information about several targets. In the advent of multiple targets to be handled, the UAV should select one of them. For this purpose, a UAV uses the attractiveness function to decide on a target. The *attractiveness function* maps the value of a target to the *attractive force value*, which represents a UAV's preference. This function depends on the value of the target and the distance between itself and the UAV, and is used to select the best target as follows:

$$\Theta_i(t) = \arg \max_j \left\{ \frac{\Pi_j(t)}{\|x_i(t) - \psi_j(t)\|} \right\}$$

where  $\Pi_j(t)$  denotes the value of target  $j$  at time  $t$ ,  $x_i(t)$  is the location of UAV  $i$  at time  $t$ , and  $\psi_j(t)$  is the location of target  $j$  at time  $t$ . If target  $j$  is stationary,  $\psi_j(t)$  is always the same regardless of time. The value between braces is called the attractive force value of target  $j$ , and  $\Theta_i(t)$  returns the index of the target that has the maximum attractive force value to UAV  $i$  at time  $t$ .

As a UAV approaches a target, the UAV starts consuming the value of the target once the UAV is within a certain distance of the target. We call the value consumed by the UAV the *utility value*. The *utility value function* and the *target value function* of the target  $i$  at time step  $t+1$  are defined as follows:

$$U_i(t+1) = \Pi_i(0) - \Pi_i(t+1)$$

$$\Pi_i(t+1) = \max\{\Pi_i(t) - d \cdot n_i(t), 0\}$$

where  $U_i(t)$  means the utility value of the target  $i$  at time  $t$ ,  $d$  is a discount factor, and  $n_i(t)$  is the number of UAVs which are near to the target  $i$  at time  $t$ . Therefore, in our simulation, when several UAVs are within the range of a target, the value of the target is consumed more quickly.

After a UAV reaches a target, it will fly around the target until the whole value of the target is consumed, either by the UAV alone or in conjunction with a group of UAVs. In our UAV simulation, one purpose of collective behavior of UAVs is to maximize the accumulated utility value within as short a time as possible. Here, the *accumulated utility value* means the whole value of targets consumed by all the UAVs.

## 3. SIMULATION ARCHITECTURE

Our distributed simulation is comprised of three layers: user interface, UAV simulator, and actor-based distributed platform (Figure 1). The *user interface layer* consists of two programs: Configuration Interface Program and Simulation Viewer. *Configuration Interface Program* provides an easy means of defining important attributes for the simulation. *Simulation Viewer* is a tool to check and verify the simulation results. All task oriented components, such as UAVs and targets, and simulation oriented components, such as Simulation Control Manager (SCM) and Active Broker (AB), are implemented as actors in the *UAV simulator layer*, which will be further explained in section 3.2.2. Each actor has its own thread to progress its state. The thread execution and communication of actors are

controlled by the Actor Foundry, an *actor-based distributed platform*.

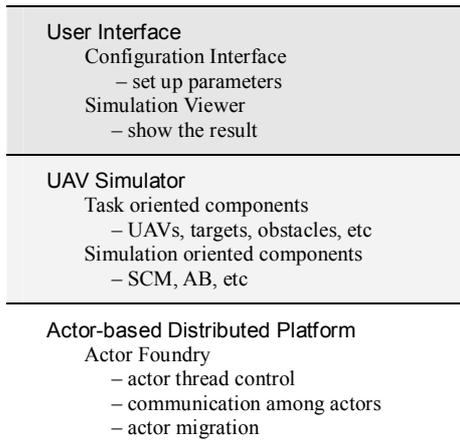


Figure 1: Three-layered Architecture for Distributed Simulation

The actor-based distributed platform is a middleware to support several distributed applications and is not tailor made for a specific simulation, such as a UAV simulation. The UAV simulator defines specific behaviors of UAVs, but does not include all the parameters to test and verify a behavior. These parameters are defined in user interface programs by a user and used in the UAV simulator. The functions of each layer are explained in detail below.

### 3.1 Actor-based Distributed Platform

The Actor Foundry is implemented in the Java programming language, and supports actor execution, communication between actors, and actor migration [Astlery 1999; Clausen 1998].

In the Actor Foundry, an actor is created by another actor or by a user. When an actor is created, the actor name of the new actor is returned. This name would be used to refer to the receiver actor in message passing or deliver the reference of another actor to the receiver actor. The actor name is unique in the actor world. Therefore, even though an actor migrates from one host to another, the name is always transparent to other actors, and hence, other actors can continuously use the same name to refer to the given actor irrespective of that actor’s current location, thereby providing a means for location transparency.

An actor in the Actor Foundry is running as a Java thread, and an actor communicates with other actors through asynchronous message passing. This is the main difference between the Actor Foundry and other object-based distributed platforms, such as CORBA and DCOM [Grimes 1997; OMG 2002]. In other object-based distributed platforms, one thread control is assumed: when an object is called by another object, the caller object is blocked until the called object returns the thread control. In the Actor Foundry, since every actor has its own control thread

to perform its operation and communicates with others through the asynchronous communication, the execution of an actor does not depend on those of others. Due to these features, we can easily use the power of distributed systems. Simulation components implemented as actors run on different computers independently, and they can communicate with others through the unique actor name, even though the distributed platform migrates some components from one host to another.

When distributed components interact with each other through asynchronous communication, analyzing the delivery sequence of communication messages is burdensome because asynchronous communication does not guarantee the message delivery order requirements, such as FIFO order, causal order, or total order [Hadzilacos and Toueg 1993]. Our distributed platform makes a log for message passing among actors, so that users can easily analyze the delivery sequence of messages.

### 3.2 UAV Simulator

All simulation components in our UAV Simulator can be classified into two categories: task oriented components and simulation oriented components (Figure 2). Task oriented components simulate objects in real situations. For example, a UAV component maps to a UAV object in a real situation while a target component maps to a target object. For the purpose of simulation, we need some virtual components, such as Simulation Control Manager and Active Broker. The following sub-sections explain these two categories of components in detail.

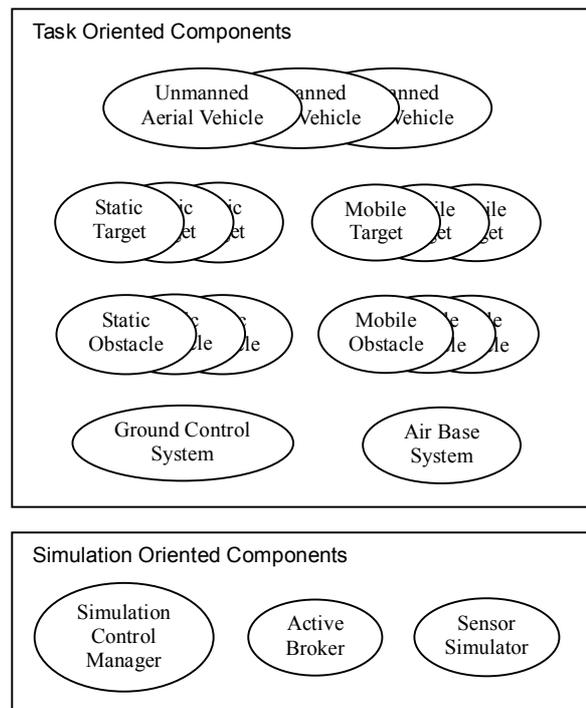


Figure 2: Simulation Components in UAV Simulator

### 3.2.1 Task Oriented Components

Task oriented components in our UAV simulator consist of five types: Ground Control System (GCS), Air Base System (ABS), Unmanned Aerial Vehicles (UAVs), obstacles, and targets. *GCS* is a central manager of UAVs and is aware of the mission area so as to indicate each UAV its air route in the area. However, *GCS* may not communicate continuously with UAVs to decide behavior of the UAVs at each time step because UAVs are supposed to perform their mission autonomously. *ABS* represents a control center of an air base and controls the sequence of take-offs and landings of UAVs. *UAVs* perform a given mission autonomously within certain restrictions, such as their kinematics and communication capability. *Obstacles* represent objects in which UAVs are not interested and with which a collision can happen. According to whether an obstacle can move or not, they are classified into two classes: *a mobile obstacle*, such as an airplane, and *a static obstacle*, such as a tall tower or a building. *Targets* represent objects of interest for the UAVs, such as, civilians to be rescued. According to its mobility characteristics, there are mobile targets and static targets.

### 3.2.2 Simulation Oriented Components

#### 3.2.2.1 Simulation Control Manager.

Each component manages its virtual time because each actor has its own control thread. However, this situation can cause inconsistency in virtual times of components. To maintain consistency between virtual times, *Simulation Control Manager (SCM)* manages local virtual times of the simulation components. When every component starts its execution, the initial value of each local virtual time is set to 0. After every component starts, *SCM* broadcasts a *virtual time clock message* to the other components. When a component receives the message, the component increases its local time and performs a small portion of its task that should be completed during the predefined time slice unit. For example, when a UAV receives the message, it updates its location and direction vector, and also checks whether or not new objects, such as other UAVs, targets, or obstacles, are detected. If a new neighboring UAV is detected, the UAV might exchange some information with the new neighboring UAV. After a component finishes its computation, it sends a reply message to *SCM*. When *SCM* receives reply messages from all the other components, *SCM* increases its virtual time, and rebroadcasts another virtual time clock message.

#### 3.2.2.2 Active Broker

In order for a UAV to perform a group mission, the UAV needs to communicate with its neighboring UAVs through local broadcasting. *Active Broker* simulates a local broadcasting mechanism. In general, the brokering service supports attribute-based

communication. For example, if every UAV registers information about its current flying area with its actor name on a shared space, then when a UAV requests a broker for a message passing with a template that describes a certain area, the broker delivers the message to other UAVs which are in the area. However, this approach is not very accurate for finding the neighboring UAVs. Therefore, we have extended the function of the brokering service. In the active brokering service, every UAV registers information about its current location with its actor name on the shared space, and a UAV sends a special object instead of the template to request a message delivery to Active Broker. The object includes a specific method to be called by Active Broker. The method computes the distance between the location of the sender UAV and other UAVs and selects some which are within the local communication range. When the method returns actor names of neighboring UAVs, Active Broker delivers to them the message received from the sender UAV.

#### 3.2.2.3 Sensor Simulator

Although each real UAV is supposed to be equipped with its own radar sensor, the radar sensors of all UAVs is simulated by a single simulation oriented component, *Sensor Simulator*. In the simulation, UAVs, targets, and obstacles register their current locations on a shared space every second in virtual time. *Sensor Simulator* periodically computes the distance between any two objects. If some components are within the sensor range of a UAV, *Sensor Simulator* reports information about these components to the UAV. Each UAV regards this information as its sensor input.

### 3.2.3 UAV Architecture

The most important simulation component is a UAV component. Therefore, we explain the architecture and the main behavior of a UAV in this subsection. A UAV is comprised of four modules: the physical process module, the trajectory planning module, the cooperative module, and the global control module (Figure 3).

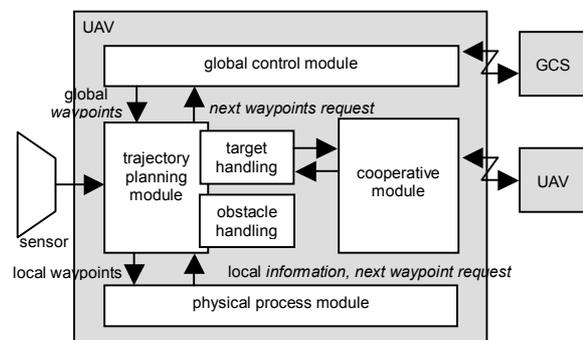


Figure 3: The Architecture of the Unmanned Aerial Vehicle Actor

When a UAV starts its mission, it does not have any information about its air route or the mission area. In our simulation, an air route is defined as a set of *waypoints* that need to be traversed by the UAV. Therefore, the first task of a UAV is to request the waypoints from GCS. The *global control module* of a UAV takes charge in communicating with GCS and managing the waypoints received. We call these waypoints *global waypoints*. When a UAV detects targets or/and obstacles, this information is delivered to the trajectory planning module from Sensor Simulator. The *trajectory planning module* handles them according to the predefined rules. For example, when a UAV detects several targets, it selects one target which has the best attractive force value, and then modifies its air route to reach the target. This function is performed by adding a waypoint to the list of UAV's current waypoints. The set of waypoints used inclusive of the additional waypoints are called *local waypoints*. The *cooperative module* is used when several UAVs want to handle a set of targets. To decide which UAV handles which target, the UAVs communicate with each other through the cooperative module. The kinematics of a UAV is implemented in the *physical process module*. Therefore, whenever this module receives a virtual time clock message, the physical process module computes the next location and the next direction of the UAV. When a UAV reaches the current waypoint, this module starts a turn toward the next waypoint according to the predefined kinematics.

### 3.3 User Interface

If we have to modify the UAV simulator whenever we execute it with different parameters, it is quite burdensome. Besides, modification at the code level requires comprehension making it hard for novice users to modify the code. In our architecture of UAV simulation, we separate the parameter modification part from the UAV simulator code as the user interface layer. Moreover, we separate the simulation checking part from simulator code. Therefore, the user interface layer consists of two programs: Configuration Interface Program and Simulation Viewer.

#### 3.3.1 Configuration Interface Program

For the convenience of novice users, we have separated the configuration for UAV simulation parameters from the simulator code as a configuration file. This file can be modified by the Configuration Interface Program (Figure 4). Therefore, although a user does not look at and understand the source code for UAV simulation, they can change important parameters of simulation and run it without recompiling the source code. With this program a user can set up the number of UAVs, the size of mission area, the attributes of targets and obstacles, maximum simulation time, and the size of simulation time slice unit.

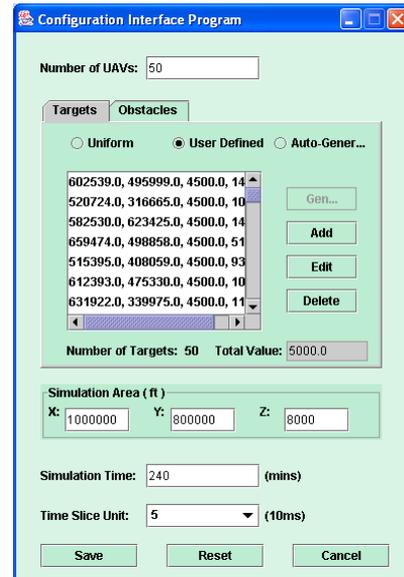


Figure 4: Configuration Interface Program

#### 3.3.2 Simulation Viewer

Because of the characteristics of large scale simulations whose durations may sometimes be so long that we cannot monitor the simulation results continuously, we have separated the simulation checking from the simulation execution. Therefore, we look at and check the simulation results through Simulation Viewer (Figure 5). Another advantage of this approach is that the simulation results can be viewed back and forth with respect to the simulation virtual time.

While our UAV simulator is running according to the given parameters, the simulator generates simulation results on data files. The data files contain the locations and directions of UAVs, targets, and obstacles at every simulation virtual time step. The Simulation Viewer is used to check and verify the simulation results.

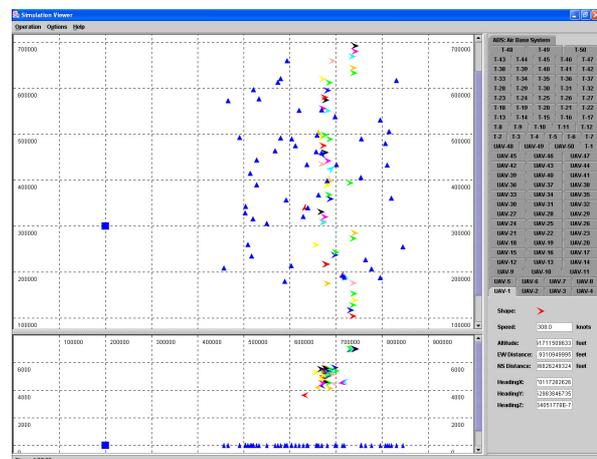


Figure 5: Simulation Viewer

## 4. COOPERATION AND COORDINATION AMONG UAVS

Cooperation among the UAVs is essential in directing the adjustment of policies in the globally most beneficial direction. In addition to cooperative dissemination of information, coordination of actions in larger teams is essential. With elements of uncertainty existing in the environment, coordination among UAVs has to be adaptive. The UAVs need to dynamically allocate responsibilities for different subtasks depending on the changing circumstances of the overall situation. For example, if additional targets are detected during a group mission, a team of UAVs needs to be able to handle them either by recruiting new member UAVs or changing the previous assignment of targets. In our UAV simulation, we use three strategies: the self-interested UAV strategy, the sharing-based cooperation strategy, and the team-based coordination strategy.

### 4.1 Self-interested UAVs

In the self-interested UAV strategy, a UAV senses a target and approaches it with the intention of consuming its entire value. When another UAV detects the same target, it also proceeds to consume the value of the target, irrespectively of what other UAVs do. Incessant polling of the target value till such time it is consumed completely serves as a means of interaction among the UAVs. It is not unusual to have more than one UAV concentrated on a target resulting in quicker consumption of its value, but also possibly in duplication of service.

### 4.2 Sharing-based Cooperation

In this strategy, once a UAV has discovered and located a target, it broadcasts this information so that other UAVs could direct their attention to the remaining targets. Reception of such information will result in the UAVs purging the targets that were advertised. This approach allows for a larger set of targets to be visited in a given time interval and is thus expected to be faster in accomplishing the mission goal. Exchange of information between UAVs referring to the same target will result in a UAV with a lower identification number to determine the UAV that would be responsible for this target based on parameters such as the distance from the target.

### 4.3 Team-based Coordination

In the team-based coordination strategy, certain UAV takes on the mantle of the leader of its team and dictates course of action to the other UAVs about the targets they need to visit. A team is dynamically formed and changed according to the set of targets detected; i.e. when a UAV detects more than one target, the UAV tries to handle the targets together with its neighboring UAVs. At this time, the main concern is

how to select an optimum UAV and decide the number of UAVs required to accomplish a task, when there are a sufficient number of neighboring UAVs. As the basic coordination protocol, we use the Contract Net protocol [Smith 1980; Smith and Davis 1981]. The UAV initiating the group mission works as the *group leader UAV*, and the other participant UAVs are called *member UAVs*. When a member UAV detects another target, the UAV delivers information about the new target to the leader UAV, and the leader UAV will add the target to the set of targets to be handled. The leader UAV considers the distance between a target detected and neighboring UAVs to assign the target. When a member UAV consumes the entire value of a target the UAV secedes from its group.

## 5. EXPERIMENTAL RESULT

We have developed the UAV simulator and two interface programs in Java programming language. Our UAV simulator is running on the Actor Foundry, but interface programs do not require the Actor Foundry. In order to simulate the flying and turning behavior of UAVs, we use the basic kinematics model of airplanes, but we abstract away the detailed dynamics and kinetics of aircraft.

For the UAV simulation, the size of the simulation area is set to  $1,000,000 \times 800,000 \times 8,000$  cubic feet (length  $\times$  width  $\times$  altitude), size of the mission area to  $400,000 \times 500,000 \times 8,000$  cubic feet, the radius of local broadcast communication of a UAV to 50,000 feet, and the radius of radar sensor to 25,000 feet. There are 50 targets in the mission area, and they are normally distributed. Half of the targets are static and the others are dynamic targets. When a UAV is within 1,000 feet from a target, the UAV consumes the value of the target. The initial value of each target is 100, and the discount factor  $d$  in the target value function is 5 per second.

To investigate how different cooperation strategies influence the performance of a joint mission, we use Average Service Cost (ASC) defined as follows:

$$ASC = \frac{\sum_i^n (NT_i - MNT)}{n}$$

where  $n$  is the number of UAVs,  $NT_i$  means navigation time of UAV  $i$ ,  $MNT$  (Minimum Navigation Time) means average navigation time of all UAVs required for a mission when there are no targets.

Figure 6 shows Average Service Cost for three different cooperation strategies. When the number of UAVs is increased, ASC is decreased in every case. However, the sharing-based cooperation strategy and the team-based coordination strategy are better than the self-contained UAV strategy. From this result, we conclude that communication of UAVs is useful to handle targets, even though UAVs in the self-

contained UAV strategy consumes quickly the value of a target when they handle the target together. Another interesting result is the performance of the team-based coordination strategy is similar to that of the sharing-based cooperation strategy, even though the algorithm of the sharing-based cooperation strategy is much simpler. The overall ASC of the team-based coordination strategy is 3 or 5 seconds faster than that of the sharing-based cooperation strategy. When  $n_i(t)$  in the target value function is not used, the performances of the sharing-based cooperation strategy and the team-based coordination strategy are not changed very much while that of the self-interested UAV strategy is decreased (Figure 7).

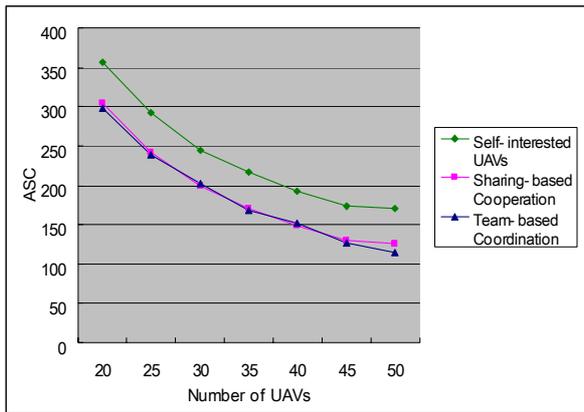


Figure 6: Average Service Cost (ASC) for three different coordination strategies.

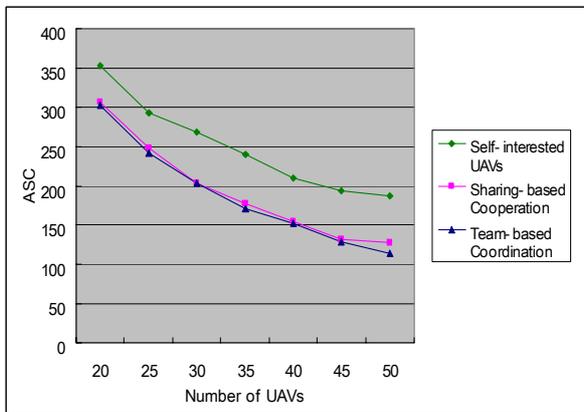


Figure 7: Average Service Cost when  $n_i(t)$  is not used.

## 6. RELATED WORK

Johnson and Mishra present a flight simulation tool for GTMax (Georgia Tech R-Max VTOL UAV) [Johnson and Mishra 2002]. Barney Pell and his colleagues describe the NMRA (New Millennium Remote Agent), architecture for a UAV. The NMRA integrates real-time monitoring and control with planning and scheduling, handles fault recovery and reconfiguration of component models, and simulates the autonomy of a UAV [Pell et al. 1997]. However, the type of the

GTMax UAV is a helicopter, and both papers do not handle cooperation among UAVs.

Altenburg and his colleagues present an agent based simulator to simulate UAV cooperative control [Altenburg et al. 2002]. In their approach, agents are reactive agents while UAV components in our simulation are deliberative agents. Therefore, their agents directly respond to signals from environment, while our agents change their intention about targets and automatically and proactively select a different action. Also, their agents communicate with others indirectly through the environment while our agents communicate with each others directly. Kolek and his colleagues present a simulation framework to evaluate the performance of real time tactical radio networks with a UAV [Kolek et al. 1998]. In this paper, the authors explain how much distributed simulation could be applied to solve military problems, but they do not handle the autonomy of UAVs and coordination among UAVs.

## 7. FUTURE WORK

The Actor system supports distributed computational environment and actor mobility. In the current platform, it is the programmer's role to determine actor placement. However, this is hard to do when we do not know the CPU speed and the communication speed among different machines. Specifically, when the communication pattern among actors is changed, the initial placement of actors might prove to be a deterrent to cross boundary communication. For this, we are developing dynamic actor reconfiguration algorithm. In the new actor platform, the communication pattern among actors will be monitored, and actors will be dynamically reallocated by the platform.

Another problem of the current actor system is the existence of Simulation Control Manager (SCM) to control the virtual times of UAVs globally. This component may be a bottleneck of the distributed simulation, and if this component were to fail, the simulation would collapse completely. To counter this, the Jefferson's virtual time [Jefferson 1985] based actor platform can be used. In this actor platform, each actor maintains its own virtual time, and when an actor communicates with another actor and the time difference is more than the given threshold, the platform performs the rollback.

As another extension, we are looking to merge a few real UAVs into UAV simulation. That is, we are going to build a UAV simulator with the possibility of real time input from real UAVs and virtual UAVs. In this simulation, a real UAV can communicate with other real UAVs and virtual UAVs to perform a virtual task. This approach can overcome the problem of computer simulation, such as the inaccuracy of UAV kinematics and the communication delay defined by programmers.

In our simulation, we use Contract Net Protocol. It means if a UAV accepts the order from a leader UAV,

the UAV must handle the target. However, the belief about environment changes when UAVs detects more targets or additional UAVs become available after having consumed value of their respective targets. Therefore, when any change in the environment is detected or any UAV becomes available, this information is delivered to the leader UAV, and the leader UAV may reconsider and change the target assignment. Also, a member UAV may secede from its team to handle a new target with a more attractive force value. This idea is motivated from the leveled commitment in Contract Net Protocol [Sandholm and Lesser 1995].

## 8. CONCLUSIONS

In this paper, we have described the design and development of a distributed UAV simulator using an actor-based platform, a utility function, and Contract Net Protocol. The three layered architecture for our UAV simulation is presented: the actor-based distributed platform, the UAV simulator, and the user interface layer. We have described three strategies to perform a joint mission: the self-interested UAVs strategy, the sharing-based coordination strategy, and team-based cooperation strategy. This has been supplemented by our experimental results and outline of the future work.

Our UAV simulator is working on an actor-based distributed platform, and hence, it naturally adapts to the behavior of a distributed and concurrent situation. We can easily improvise the execution environment without changing the UAV simulator by separating the distributed platform from the simulator. For example, we can migrate some actors from a computer to another during the execution time. Other possible means for improvising the working environment have been presented in the future work section. When we consider multiple UAVs working together, their cooperation mechanisms are of utmost importance. In this paper, we have presented three different approaches, and compared and contrasted them. The experimental results suggest that cooperation and coordination strategies are better than the self-interested UAV strategy. Last but not least, we have introduced the active brokering service to support application oriented searching.

## ACKNOWLEDGEMENT

This research is sponsored by the Defense Advanced Research Projects Agency under contract number F30602-00-2-0586. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

## REFERENCES

- Agha, G.A. 1986. *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, Cambridge, Mass.
- Agha G.A.; I.A. Mason; S.F. Smith; and C.L. Talcott. 1997. "A Foundation for Actor Computation." *Journal of Functional Programming*, Vol. 7, No. 1, 1-69.
- Altenburg K.; J. Schlecht; and K. Nygard. 2002. "An Agent-based Simulation for Modeling Intelligent Munitions." In *Proceedings of the Second WSEAS International Conference on Simulation, Modeling and Optimization*, Skiathos, Greece (Sep). Available at <http://www.cs.ndsu.nodak.edu/~nygard/research/munitions.pdf>
- Astlery M. 1999. *Actor Foundry*. Department of Computer Science, University of Illinois at Urbana-Champaign, IL (Feb. 9). Available at <http://yangtze.cs.uiuc.edu/foundry>
- Clausen T.H. 1998. *Actor Foundry - a QuickStart*. Department of Computer Science, Institute of Electronic Systems, Denmark (Nov. 9). Available at <http://yangtze.cs.uiuc.edu/foundry>
- Doherty P.; G. Granlund; K. Kuchcinski; E. Sandewall; K. Nordberg; E. Skarman; and J. Wiklund. 2000. "The WITAS Unmanned Aerial Vehicle Project." In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, Berlin, Germany (Aug), 747-755.
- Grimes R. 1997. *Professional DCOM Programming*. Olton, Birmingham, Canada, Wrox Press.
- Hadzilacos V. and S. Toueg. 1993. "Fault-Tolerant Broadcasting and Related Problems." In *Distributed Systems*, S. Mullender (Ed.). ACM Press, New York, 97-145.
- Jefferson D. 1995. "Virtual Time." *ACM Transactions on Programming Languages and Systems*, Vol. 7, No. 3 (Jul), 404-425.
- Johnson E.N and S. Mishra. 2002. "Flight Simulation for the Development of an Experimental UAV." In *Proceeding of the AIAA Modeling and Simulation Technologies Conference and Exhibit*, Monterey California, CA (Aug), 5-8.
- Kolek S.R.; S.J. Rak; and P.J. Christensen. 1998. "Battlefield Communication Network Modeling." *The DIS Workshop on Simulation Standards*. Available at <http://dss.ll.mit.edu/dss.web/98F-SIW-143.html>
- OMG. 2002. *The Common Object Request Broker Architecture: Core Specification*. Version 3.0.2 (Dec).
- Pell B.; D.E. Bernard; S.A. Chien; E. Gat; N. Muscettola; P.P. Nayak; M.D. Wagner; and B.C. Williams. 1997. "An Autonomous Spacecraft Agent Prototype." In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, 253-261.
- Sandholm T. and V. Lesser. 1995. "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework." In *Proceedings of the 1st International Conference on Multiagent Systems*, San Francisco, CA, 328-335.
- Smith R.G. 1980. "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver." *IEEE Transactions on Computers*, Vol. 29, No. 12, 1104-1113.
- Smith R.G and R. Davis. 1980. "Frameworks for Cooperation in Distributed Problem Solving." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 11, No. 1, 61-70.

## **AUTHOR BIOGRAPHIES**

**MYEONG-WUK JANG** is a doctoral candidate and research assistant in the Open Systems Laboratory at the University of Illinois at Urbana-Champaign. His research interests include multi-agent system and task allocation in open distributed computing. He received a BS in Computer Science from Korea University in 1990 and an MS in Computer Science from KAIST (Korea Advanced Institute of Science and Technology) in 1992. He worked for ETRI (Electronics and Telecommunications Research Institute), Korea, until 1998. His web page can be found at <http://www.uiuc.edu/~mjang/>.

**SMITHA REDDY** is a Master/PhD student and research assistant in the Open Systems Laboratory at the University of Illinois at Urbana-Champaign. Her research interests include distributed systems, high speed networks, and dynamic resource sharing. She received a BE in Computer Science from University of Pune in 1999.

**PREDRAG TOSIC** is a doctoral candidate and research assistant in the Open Systems Laboratory at the University of Illinois at Urbana-Champaign. He received a BS in Mathematics and Physics and an MS in Applied Mathematics, both at University of Maryland Baltimore County, UMBC, in 1994 and 1995, respectively, and also holds an MS in pure Mathematics from University of Illinois at Urbana-Champaign in 1997.

**LIPING CHEN** is a doctoral candidate and research assistant in the Open Systems Laboratory at the University of Illinois at Urbana-Champaign.

**GUL A. AGHA** is Director of the Open Systems Laboratory at the University of Illinois at Urbana-Champaign and Professor in the Department of Computer Science. His research interests include models, languages, and tools for parallel computing and open distributed systems. He received a BS in an interdisciplinary program from the California Institute of Technology, an MA in Psychology from the University of Michigan, Ann Arbor, and an MS and PhD in Computer and Communication Science, from the University of Michigan, Ann Arbor.

# LIMITATIONS OF THEORETICAL AND COMMONLY USED SIMULATION APPROACHES IN CONSIDERING MILITARY QUEUEING SYSTEMS

Nebojsa Nikolic  
Military Academy, Postgraduate School, Army of Serbia and Montenegro  
Ratka Resanovica street, No.1, Belgrade  
E-mail: nidzan@ptt.yu

## KEYWORDS:

Military queueing systems, Stochastic process simulation, Steady state detection, VV&A&C

## ABSTRACT:

This paper is about military queueing systems that are characterized by finiteness, heavy traffic, and even overloading. Queueing theory deals with infiniteness. Simulation methods have serious problems with heavy traffic, and with accuracy and reliability of simulation results. Both do not concern themselves with overloading. Neglecting the fact that military queueing systems performed their missions in a finite time period, and by applying only steady state results, can produce big mistakes in considering their behavior and determining performance measures. Considering above problems leads to a need for an effective solution of the queueing transient phenomenon.

## 1. INTRODUCTION

Many military situations, processes or systems can be considered as queueing systems. Those can be of various types and sizes, and related to:

- Battle situations;
- Weapon systems and various technical items;
- Military logistic functions;
- Command processes, and so on.

Some very good examples can be found in literature, like [13.], Shephard and others, 1988. Military research studies usually deal with complex situations, but before studying those, are we able to completely solve some relatively simple situations?

### Observation on finiteness

Military units in war, are not continually engaged; enemy tanks, rockets and airplanes are not continually in sight through war; one attack or defense operation usually are planed and executed for a limited part of time. Observation about finiteness of reality is keystone that caused all these research efforts. Constellation of applicable queueing theory knowledge (steady state solutions) and observation on finiteness implicate the main hypothesis: "Real system's behavior, in a finite working time, can differ from its steady state behavior".

## Example

Here, evacuation process of battle damaged tanks in an separate armored brigade will be considered. It is supposed that brigade performs its full battle mission. In general, duration of brigade's full engagement is limited to about 3 to 5 days; after that time brigade needs some rest. Heavy-damaged tank needs special vehicle-HET (Heavy-Equipment Transporter; this is usually one wheeled wrecker) for its transportation from forward combat zone to rear zone, in order to be repaired. It is supposed that brigade has only one HET.

From practical point of view, relatively simple situation in this example is of triple importance: firstly, service channel represent HET (whose price is high); secondly, clients are heavy-damaged tanks, whose price is also high; thirdly, tanks' battle importance can be much greater: evacuation, repair, and come-back to the same battle! Example: fantastic score of German maintenance units in battle for Tobruk, North Africa, April 1941, 100 tanks damaged, 88 recovered. Finally, this is approved in many FMs (Field Manuals), such as: the goal is to manage limited resources to return the maximum number of critical items to the battle.

This situation is chosen as an important and concrete enough example, on which the problem stated in title will be demonstrated. Table 1 presents three possible variants of traffic intensities.

Table 1. Input values for queueing system

Variants	Queueing system type is: M/M/1/∞		
	Average times between demands	Average service times	Traffic intensity
1.	240	200	0.833
2.	200	190	0.95
3.	200	240	1.2

Using the language of military reality, solving of this task gives answers on the next questions which can be putt to S4 and/or maintenance officers:

1.) "How much shall I have to wait for my tank to come back repaired?" (battalions' commanders).

- 2.) "How many places shall I have to prepare on the collecting point for evacuated items?" (maintenance unit's commander).
- 3.) "Do you need support (more HETs) for evacuation in your brigade?" (G4 officer in war; TOE makers in peace (Table of Organization and Equipment)).

## 2. SOLVING BY QUEUEING THEORY

From theoretical point of view this is a single server queueing system. At the first sight this is very simple case, almost trivial, but this can be true only for its structure (one queue, one server), but not for its behavior (Cohen's "Single server queue" has about 600 pages!).

Above example can be easily solved using correspondent queueing theory formulas, but they are valid only in case of steady state conditions, and for traffic intensity of up to 1 ( $\rho < 1$ ). That means that the task for third variant, cannot be solved, even if that situation is really possible. Table 2 presents a few usually treated measures of performance of such queueing system.

Table 2. Queueing theory steady state results

Variants	Average waiting time in queue $W_q = T_\mu \cdot \rho / (1 - \rho)$	Average queue length $L_q = \rho^2 / (1 - \rho)$	Average server utilization $\rho = \lambda / \mu$
1.	1000	4.17	0.833
2.	3610	18.05	0.95
3.	Inapplicable for $\rho > 1$ ! (that is: $\infty$ , and $\infty$ )		1.2
Queueing system type is: M/M/1/ $\infty$			

One of the rare good things in every battle and war as a whole, is the fact that its duration is finite. Like in a sport match, playing and results are only important during the game time; after that it is another story. This fact is taken as one of the crucial moments for studying military queueing systems: their engagement is time limited. Models created for investigating such reality must respect this fact. Usually calculated RESULTS ABOVE ARE QUESTIONABLE, because it cannot be known in advance that our system reached steady state for a finite time engagement (in this example, it is a 5 days).

Queueing theory uses exact mathematical approach, but not for all types and size of queueing systems, and not for all conditions (Larson, Odoni, 1981), [7.], and this is "state of the art" until today. Main reason for this is a simple fact that "queueing theory is hard" (Kleinrock, 1979), [6.], especially if one wants to know more about behavior of queueing system in the period before steady state. Complexity of mathematical analytic approach

causes serious difficulties in practical application, even for mathematicians, and even for so-called simpler queueing systems. There is no doubt that dealing with "hard mathematics" takes a care, time and energy of the researcher, and instead of being dedicated to main subject of investigation, he is dedicated to the method.

It can be summarized what the LIMITATIONS are, when QUEUEING THEORY should be applied in solving such real situations:

- 1.) Treating the whole busy-cycle (transient period and steady state period);
- 2.) Treating the complex systems (queueing networks);
- 3.) Treating queueing systems of general type;
- 4.) Treating overloaded systems (case when  $\rho > 1$ ); and
- 5.) Defining the EFFECTIVE method for solving all above problems; term "effective", here includes: universality, simplicity, reliability, accuracy and cost.

### Check-point

In many queueing theory books, problem of practical beginning of steady state was not treated too much, however some results could be found like [10.] by P.Morse, where he suggests (page 67) a very simple formula for relaxation time of queueing system type M/M/1. There is no comment about maximal error when formula is used, but its existence itself can help very much, as it will be shown. For easier application, that formula will be transformed like this:

$$T^* = \frac{T_\mu}{(1 - \sqrt{\rho})^2} \dots\dots\dots (1.)$$

where:  $T^*$  - relaxation time (steady state beginning);  
 $T_\mu$  - average service time ( $T_\mu = 1/\mu$ );  
 $\rho$  - traffic intensity ( $\rho = \lambda/\mu$ ).

By using this formula, steady state practical (approximate) beginning can be easily calculated, for various traffic intensities. Also, it can be expressed in non-dimensional, relative time units ( $T^*/T_\mu$ ), so the solution has universal character (it is valid for queueing system where time unit is hour, as well as for another with time units expressed in milliseconds, and so on). Let's calculate now approximate steady state beginning according to formula (1.), for a set of different traffic intensities. Results are in Table 3.

Table 3. Steady state beginning for type M/M/1/ $\infty$

Traffic intensity	Approximate Steady state beginning [expressed in relative units: $T_\mu$ ]
0.833333	132
0.95	1,560
0.99	39,800
1.2	110

How to use this results: In our example queueing system works for a finite time of 7,200 time units; this value should be divided with  $T_{\mu}$  (average service time); if that value is lower than the corresponding one from above table, than our system does not reach steady state during its busy-cycle. If that value is much, much higher, than steady state is practically reached. This is certainly not too much accurate procedure, but it can helps as a good orientation.

For our example it is clear that queueing system, for any variants of traffic intensity does not reached steady state. So, it can be concluded that theoretical solutions (Table 2) are NOT VALID for this example. Our system works all the time only in transient regime. Logical question now, is: if those results are not good, how to get correct solutions? A little poetry can help here:

*“Can we wait for steady state,  
or we must study the un-steady.”*

Effective answer will be obtained by another method – simulation modeling.

### 3. SOLVING BY SIMULATION

#### Simulation paradigm and crisis

Simulation itself is a phenomenon and deserves a few words more, but not to explore known things, rather specific ones, maybe new:

1.) *What is Simulation, is it "art or science?"*– Both! It is Science because it must be mathematically founded as a method and verified by the results. It is Art because it include specific “know-how“ skills which still can not be exactly expressed: two man painting (modeling, programming!), one is Leonardo, the other can be anybody! Knowledge on Simulation can be presented as vocabulary, but real good doing Simulation is poetry!

2.) *Whatever the Simulation is, where it belongs, to what known sub-field of art or science?* – Everywhere! There are a lot of various fields of engineering, but other areas too, where simulation takes its place. On the other side, there is no “Faculty of Simulation“, or “Simulation Academy“, so it is a paradox, but true that simulation is everywhere and nowhere. It might be the destiny, as well as any new discipline– “a new paradigm of science investigation”[15.].

3.) *Who deals with simulation?* It could be said, simulation community consists of three general groups. First ones- the practitioners are those who should say: What to do (by simulation). Second ones- the mathematicians, they should say: How to do. And third ones- informaticians, they should only: Do it (write the program). One simulation “dream-team“, certainly has to include specialists for all three areas. The better case is “dream-team” league, that is a few independent simulation teams, working separately on the same

problem. In that case there is a small possibility for monopoly on the science and truth.

4.) *Answer the questions linked on: goodness of a simulation model; and accuracy and reliability of simulation results,* that is, very shortly- VV&A&C (Validation, Verification, Accreditation, Credibility) questions. One of the first sharp warnings to the simulationists came from B.Gaither [5.], more as an impression but high qualified (he was editor in chief of ACM Performance Evaluation Review): he “...does not know any other field of engineering or Science, where similar liberties are taken with empirical data“. This impression is confirmed 10 years later, by an detailed investigation [15.], where was said that more than 70% of simulation papers are of “don’t care” type (clearly: don’t care on VV&A&C). Their conclusion on this situation was logically marked as Crisis. An aspect of the “Crisis“ is mentioned also in papers of M. Neuts [11.]. Thinking about Crisis in simulation, and remembering on Tomas Kuhn’s exciting book “*Structure of scientific revolution*“, it is logical to conclude that time must come for radical changes in simulation field.

#### Initial, transient, start-up, ....

Regardless on its generality, previous notations are deeply involved in this investigation. One of the consequences is locating efforts on effective investigation of period before steady state. Depending on point of view, this period is known in a literature as: *initial period; start-up period; transient period; non-stationary period; warm-up period; relaxation time*. It could be a very interesting story about why there are so many names for only one thing, which is, by the way, just known to simulationists and specialized mathematicians! Also, in much literature there is an opinion about fast reaching the steady state, that is neglecting the initial period. Anyway, this is one of the long-lived problems in queueing simulations (from seventies [4.] until today).

#### Example solving

Simple simulation model (using GPSS) was created for situation described in Example, and in Table 4. are presented simulation results only for first variant of traffic intensities. There is used so-called "one simulation approach", but not "long simulation run", then terminating (fixed time period). Many simulationists, certainly would have objection on this way of solving tasks like this one. But, this was necessary in order to demonstrate inferiority of "one simulation approach". Also, it should be noticed, that this approach, in some local scientific (or "scientific") societies is known as the only method of simulation!

Table 4. shows that for different RNG, results differ from each other, and from theoretical results too. A set

of logical questions arise: Which result is correct? Why do they differ? Why does this happen? The answer is simple: this approach is, principally wrong. Or, to say it in mathematically precise manner: above results are so good, as it can be an estimation of an stochastic variable

from sample of size- one element! The ratio of maximal error and confidence level, for “one-element sample size”, is entirely un-useable. Arbitrary choosing the RNG which obtains the best results, is not acceptable.

Table 4. One- not long- simulation run solutions for first variant ( $T_\lambda=240$ , and  $T_\mu=200$  time units)

type M/M/1/ $\infty$	Denotation of used Random Number Generator RNG(i,j))		Average waiting time in queue	Average queue length	Average server utilization
	input stream RNG (i)	output stream RNG (j)			
1.	2	6	380	1.66	0.692
2.	3	7	437	1.51	0.740
3.	4	5	300	1.70	0.870
Queueing theory steady state results			1,000	4.17	0.833
NOTES: - simulation run-length: 7200 time units (5 days expressed in minutes)					

### One long run results

Commonly used simulation approaches tend to obtain only steady state results. In spite of modeling the reality, the philosophy is to model the useable queueing theory, that is only steady state. There are a lot of sophisticated, but relatively complex statistical procedures, which obtain steady state results.

One of the problems is how to determine the simulation run length. For this example, the simplest procedure is chosen: run-length is increased 10 times, then 100 times, and so on. Also, as we have some theoretical results, let it exclude statistic from first period of length  $132 * T_\mu$ , and see what happens then. The results are in Table 5.

Table 5. Excluded initial transience, one long simulation run solutions for first variant

Simulation's run-length enlargements	Denotation of used Random Number Generator (RNG(i,j))		Steady state results (excluded statistic from initial period of length: $132 * T_\mu = 132 * 200$ time units= $26,400$ time units)		
	input stream RNG (i)	Output stream RNG (j)	Average waiting time in queue	Average queue length	Average server utilization
10 times	2	6	456	1.86	0.727
100 times	2	6	743	2.97	0.808
1,000 times	2	6	945	3.92	0.827
<i>Queueing theory steady state results</i>			1,000	4.17	0.833

Again, some poor results are evident. Steady state should begin after  $132 * T_\mu$ , but results are not good enough. But, it is clear better steady state results for very long simulation run length: the longer- the better. Let's translate now these modeled situations into the real situations: engagement duration of queueing system, is 50; 500; and 5,000 days respectively! Even entire wars, especially modern ones, do not have such long duration! Also, it cannot be investigated the case when traffic intensity is great than 1 (overloading). For the heavy traffic case ( $\rho \rightarrow 1$ ), it is needed much more increasing of run length.

It can be summarized what the LIMITATIONS are, when COMMONLY SIMULATION APPROACHES are applied in solving such real situations:

- 1.) Effective treating of the transient period;
- 2.) Effective treating of the heavy traffic situations;
- 3.) Investigation of overloaded systems (case:  $\rho > 1$ ); and

### 4.) Problems of VV&A&C of simulation model.

A little poetry, again (this time by Matthew Arnold, a real poet; taken from Mihram's book [9.]), can describe known simulation, its philosophy and problems:

*"We do not what we ought,  
What we ought not we do,  
And lean upon the thought,  
That chance will bring us through."*

### 4. IMPROVED SIMULATION APPROACH

In order to overcome described limitations, an specific simulation approach has been developed, and it is marked here as improved simulation approach. By its nature, it belongs to the statistical methods, exactly it is: simulation modeling of stochastic processes, with implemented possibilities for generating, gathering, displaying and analyzing the statistics of stochastic processes.

Finally, in the Table 6, there are correct results for considered example, for all three cases of traffic intensities. Sample size of 100,000 independent

replications of considered situations, obtain high level of reliability and accuracy of simulation results.

Table 6. Improved simulation concept results

- type of queueing system: M/M/1/∞; - battle duration: 5 days;		Solving method	
		Queueing theory (only steady state)	Simulation's results (sample size: 100,000)
<b>Variant 1</b> ( $T_\mu=200$ min., $\rho=0.833$ )	Average waiting time in queue	1000	411
	Average queue length	4.17	1.82
	Probability of server idle time	0.17	0.27
<b>Variant 2</b> ( $T_\mu=190$ min., $\rho=0.95$ )	Average waiting time in queue	3610	526
	Average queue length	18.05	2.81
	Probability of server idle time	0.05	0.20
<b>Variant 3</b> ( $T_\mu=200$ min., $\rho=1.2$ )	Average waiting time in queue	(∞)	854
	Average queue length	(∞)	4.66
	Probability of server idle time	(0)	0.12

Some experience from real situations can be useful in understanding presented results: "Our service channel is really heavy-loaded and we work almost all the time, but the queue is not so long, nor waiting is so great!". But, these results are so much different, and maybe surprising, that method which generates them, deserve a few words more. Or, to say it sharply: what are the guaranties, that method is good.

#### Verification of the method

Concept of the method completely corresponds to statistic of stochastic processes. Steady state detection is in a complete accordance with corresponding theoretical approximations. Performance measures of queueing system, obtain by simulation, have also good agreement with theoretical ones.

Primary performance measures. From basic queueing theory postulates, it can be raised that states probabilities are performance measures of highest importance, and here, they are marked as primary measures of performances. All other performance measures (queue length, waiting time, ...), depend on system states probabilities, and because of that, they can be called secondary measures of performance. But in practice, one wants to know just some of the secondary measures, and this fact is probably one of those which caused that primary measures are not studied too much. Another fact, maybe more important, is problem of creating one effective method for generating (solving, getting) states probabilities as time-dependent variables.

Comparison. Here, main idea for this problem is next: generate states probabilities as time-dependent variables. Then compare simulation results with those ones obtained by "stronger" (analytic or numerical mathematic) method. It should be clear, this is possible only for relatively simple queueing systems which can be solved completely by "stronger" method. One simple

queueing system of type M/M/1/7, is considered for a finite time engagement (6,000 time units). Average service time is 100, and average time between clients in input stream is 120 time units. This system is solved by simulation and by numerical mathematics. Results (states probabilities) are presented on Figure 1.

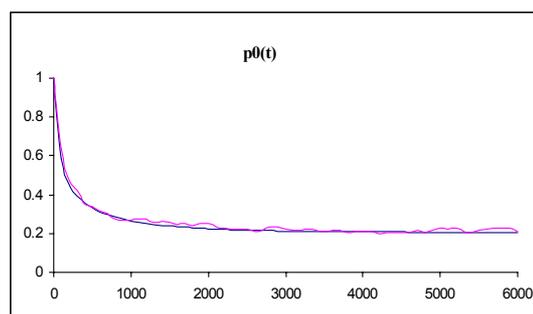


Figure 1. Comparison of numerical and simulated  $p_0(t)$

Quantification of differences. Besides the fact that good concordance is clear, it is possible to apply appropriate statistical tests to quantify differences. Simulation results based on sample size of 1075 elements (independent replications of stochastic process). There is only one system state probability presented, but the others are similar. For this sample size: maximal possible error of simulations results is 11.6% for confidence level of 95% (confidence coefficient for this confidence level is:  $z_c=1.96$ ) for  $p_0(t)$ . And, this can be tested by chi-square test.

How much (sample size) is enough. Created simulation approach obtains full controllability of ratio: desired accuracy of simulation results, and corresponding level of statistical confidence, for defined sample size. To be concrete, if we want to determine how large should the sample size be, for specified maximal error (distinction) and for desired level of confidence, in

proportions (probabilities) estimation process, than the next formula can be used:

$$N = \frac{q}{p} \left( \frac{100}{\varepsilon} \right)^2 z_c^2 \dots\dots\dots (2.)$$

where: N - sample size;  
 p - proportion (probability) to estimate;  
 q - complement to proportion (q = 1 - p);  
 ε - percentage maximal error of estimation;  
 z<sub>c</sub> – confidence coefficient.

Above formula can be generated from elementary statistical claims explained in basic courses for statistics, like in [14.]. As the matter of fact, this form is very rare in literature. Some authors even give wrong formulas, or conclude that it is not possible to determine the exact sample size.

*How small (probabilities) are enough.* Importance of considering rare events, grows in cases of heavy traffic. Then the question arises: how small probabilities are enough to consider? For answering this question, exceptional book [2.] can be of great help. The book suggests next scale for orders of magnitudes: human (10<sup>-6</sup>), earthly (10<sup>-15</sup>), cosmic (10<sup>-50</sup>), universal (10<sup>-1000</sup>). For the purpose of queueing research, only first level is enough. Exactly, it is enough to consider set of system states probabilities, which consists 99.99 % of possible system's states. States' probabilities inside this border, are up to 10<sup>-6</sup> order of magnitude, for one of the worst cases: 0.99 traffic intensity for M/M/1 queueing system.

### 5. Conclusions

Trustworthy modeling of military queueing systems declares a set of specific demands (finiteness, heavy traffic, over-loading), to the generally used methods of solving queueing problems. Those demands could not be satisfied by known methods, so the new method is created. One of the central problems was effective studying of queueing system's transient phenomenon.

In the sense of initial goals, whole study produced a set of collateral effects, all of which are very positive and important. Famous, long-lived problems in queueing simulations: start-up problem, or steady state detection problem, accuracy of simulation results, variance reduction, reliability of simulation results, and heavy traffic situations, can be easily solved using this suggested method.

### REFERENCES

[1.] Cohen J.W., "Single server queue", Nort Holand, Amsterdam, 1969.  
 [2.] De Finetti B., "Theory of probability: A critical introductory treatment", John Wiley&Sons, London, 1974.

[3.] Gross D., Harris M.C., "Fundamentals of queueing theory", John Wiley & Sons, New York, 1974.  
 [4.] Gafarian A.V., Ancker C.J., Morisaku T., "Evaluation of commonly used rules for detecting "steady state" in computer simulation", NAVAL RESEARCH LOGISTICS, 1978. page 511-529  
 [5.] Gaither B., "Empty empiricism", ACM Performance Evaluation Review, Vol.18, #2, august 1990. page 2-3  
 [6.] Kleinrock L., "Power and deterministic rules of thumb for probabilistic problems in computer communications", Conference Record, *International Conference on Communications*, Boston, Massachusetts, pp. 43.1.1 to 43.1.10, June 1979.  
 [7.] Larson R., Odoni A., "Urban Operations Research", Prentice Hall, 1981.  
 [8.] Law A., Kelton D., "Simulation modeling and analysis", McGraw Hill, New York, 1982.  
 [9.] Mihram G.A., "Simulation: statistical foundations and methodology", Academic Press, 1972. New York  
 [10.] Morse P.M., "Queues, inventories and maintenance", John Wiley & Sons, New York, 1958.  
 [11.] Neuts M., "Probability modeling in the computer age", International Conference on Stochastic and Numerical Modeling and Applications, January 1997  
 [12.] Odoni A., Ruth E., "An Empirical Investigation of the Transient Behavior of Stationary Queueing Systems", Operations Research, Vol.31, No3, May-Jun 1983. page 432-455  
 [13.] Shephard R.W., and others, "Applied operations research - Examples from defense assessment", Plenum Press, New York, 1988.  
 [14.] Spiegel M.R., "Theory and problems of statistics", McGraw Hill, New York, 1961  
 [15.] Pawlikowski K., Jeong H.D.J., Ruth Lee J.S., "On credibility of simulation studies of telecommunication networks", IEEE Communications Magazine, January 2002., page 132-139

### AUTHOR BIOGRAPHIES

**NEBOJSA NIKOLIC** is an assistant at Military Academy Postgraduate School, Army of Serbia and Montenegro. He graduated in 1988., at Military Technical Academy (Zagreb, ex-Yugoslavia), as the best student in generation. First 9 years in troops service, on various duties: from platoon and company commander, to S3, S1 and S4 officer. In year 2000. got Master of Science title in military-technical sciences. Last 7 years, interest fields: military logistic, simulation modeling, statistics; logistic in foreign armed forces. Current occupation: preparing PhD thesis.

# COMPONENT BASED MILITARY SIMULATION: LESSONS LEARNED WITH GROUND COMBAT SIMULATION SYSTEMS

*Dr. Marko Hofmann*

Institute for Technology of Intelligent Systems (ITIS) and  
Institute for Applied System Analysis and Operations Research (IASFOR)  
University of the Federal Armed Forces Munich  
Heisenbergweg 39  
Germany - 85577 Neubiberg  
0049 89 6004 3242  
marko@informatik.unibw-muenchen.de

## KEYWORDS

Components, combat simulation, pragmatics, granularity, abstraction level, reuse, repositories.

## ABSTRACT

Component based modeling is said to be one of the key technologies to improve design and development of software in general, and simulations in particular. The crucial question is which components are successful in special domains. During the last two decades scientists at our institute have designed and developed ground combat simulation systems – mainly for scientific purposes, but also to support the German army. Our experiences indicate that some assumptions of component based modeling are too optimistic with respect to directly reusable software components, especially if multifunctional. The main reason for this shortcoming lies in the problem of adjusting the pragmatics of different domain specific components. On the other hand, the reuse of concepts and algorithms has always been paramount in our model development.

## INTRODUCTION

In order to master the complexity of reality we decompose it into parts (Alexander 1964; Miller 1956; Simon 1962). In computer science modularity is regarded as one of the most promising approaches to improve design and development of complex systems (Balwin 2000; Czarnecki 2000; Szyperski 1998). As simulation systems get more and more complex, too, component-based approaches spread through all simulation domains (Dahman et al. 1998; Kuijpers et al. 1998; Zeigler 1993; Zeigler et al. 2000). There are both scientifically and practically interesting aspects of such approaches such as the granularity and the abstraction level of the components, the preconditions for their successful coupling and the structure of the repository for the storage and retrieval of the components. As these aspects are not independent they are discussed here in unity.

A crucial task in component based development regardless of the special domain is to ensure that the components can be used without knowing details of

their implementation. Ideally, it should be possible to use a component as a black box. However, during the development of our models the *technical* and *syntactical* aspects of coupling components didn't put the major challenge. All serious problems occurred on the level of *semantics* and especially *pragmatics*. It is definitely impossible to handle such problems with black boxes.

The remainder of this paper is organized as follows: Section 2 outlines the background of ground combat simulation systems. Section 3 discusses the problem of component coupling in complex simulation systems from a linguistic point of view which lays the foundation for the discussion of useful components in section 4, appropriate granularity in section 5, and how a repository should look like in section 6. The paper concludes by reiterating its main results and suggesting some future research directions.

## GROUND COMBAT SIMULATION SYSTEMS

Over more than two decades scientists at our Institute (IASFOR) have analyzed ground combat simulation systems used in the German and other armies (for example: JANUS, HORUS, SIRA, PAPST, KORA, IRIS (Stricom; IABG; CAE; Schwierz 1995) and designed and implemented own simulation systems (see below). The level of complexity of these models reaches from simplified test simulation systems and relatively simple simulations based upon cellular automata (ZEGA and ZELGAT (Hofmann 2000)) up to full scope aggregated land battle models (KOSMOS (Hofmann et al. 1992)) and high resolution ground combat models (BASIS (Hofmann et al. 1984), COSIMAC-P, COSIMAC-WS (Hofmann 2000)), which are in terms of system theory (Flood 1993) extremely complex. During that time the reuse of concepts, algorithms and code has been practiced intensively. In the following I have tried to sum up these experiences.

Ground combat simulation systems (GCSS) are a very heterogeneous class of models (Hofmann 2000; Hartman 1985), nevertheless they all share some

fundamental parts. Every GCSS has to model the following **aspects of combat**:

1. terrain and environmental representation,
2. movement,
3. attrition,
4. transportation (at least of ammunition),
5. communication and
6. reconnaissance.

Generally, GCSS are discrete event simulations based upon an event queue. The GCSS mentioned above aren't real time simulations, anyway internal time management is essential. Thus, the core of any GCSS will look roughly like Figure 1. If the GCSS is used for analysis in a closed simulation, it is necessary to add

7. command and control modeling,

which shouldn't be a part of the central simulator - for reasons explained in (Hofmann 2000).

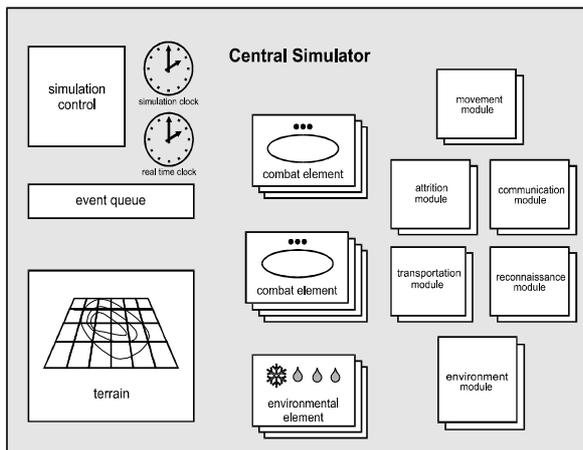


Figure 1: Essential parts of a GCSS

The major distinctions between the models, beside different **purposes** (acquisition, decision support, analyses, training), **scopes** and **user modes** (closed simulation or interactive), is their level of **resolution**: the level of detail at which the real world system and its behavior is modeled. Referring to (Davis and Bigelow 1998) and (Davis and Huber 1992) resolution in combat simulation systems has six "components":

1. temporal scale,
2. spatial scale,
3. processes,
4. entities,
5. attributes and
6. dependencies.

This classification is arguable, but useful to illustrate the degrees of freedom for the modeling. The range of two of these dimensions (spatial scale and entities) can

be easily depicted. Figure 2 shows how a combat scenario would look in an aggregated model and Figure 3 shows how a combat would look in a high resolution model. The rectangles in Figure 2 represent brigades (pale gray) and divisions (dark gray) as a whole. Attrition occurs, when two enemy rectangles overlap, and is calculated with Lanchester's differential equations (Taylor 1983). In Figure 3 the symbols represent single weapon systems of an attacking (pale) and defending (dark) force like tanks, armored infantry vehicles, mortars, armed helicopters and some other elements of combat like minefields, artillery impact areas and reinforcements of the ground.

Detailed explanations of these figures can be found in (Hofmann 2000). For further reading about the principles of ground combat simulation I suggest (Hartman 1985) and (Davis and Zeigler 1997).

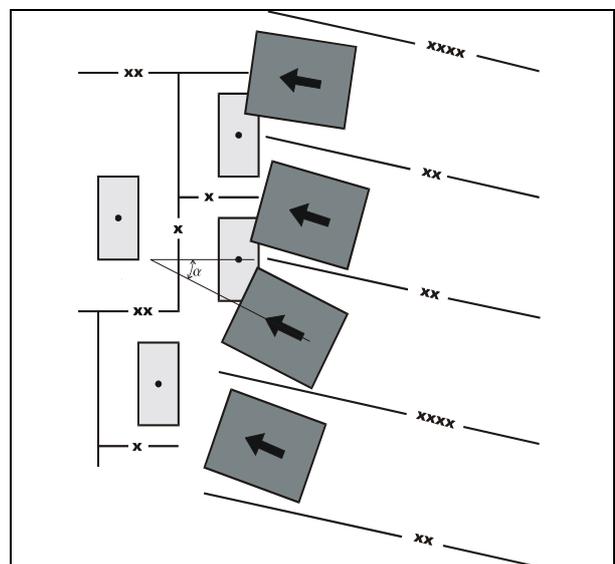


Figure 2: Depiction of a combat in an aggregated GCSS

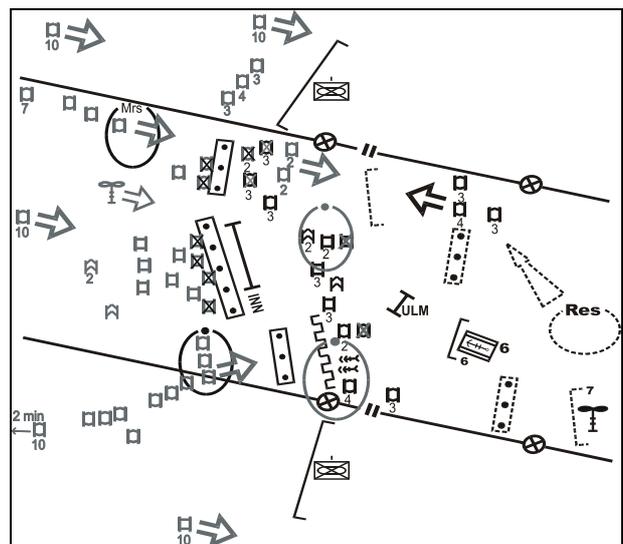


Figure 3: Depiction of a combat in a high resolution GCSS

## COMPONENTS IN GCSS: SOME FUNDAMENTAL ASPECTS

Taking into consideration the different purposes, scales, user modes and resolutions of combat simulation systems, the degrees of freedom within each of these aspects and the necessity to tailor each model to fit the purpose, it is not very surprising that **reusing** software components directly in our GCSS was and still is a rare possibility, except from the use of some domain independent components such as random number generators and data bases. These components were always relatively easy to integrated because they do not contain any semantic and pragmatic context information.

In the following, the concept of pragmatics is introduced to explain the difficulties with the coupling of domain specific and “meaningful” components as general as possible.

As the complexity of the real world combats is too large to be fully captured in a model, it is necessary to simplify. Actually, the hard part of developing GCSS is not code generation but appropriate **modeling (abstraction and idealization)**. Since the measure of this appropriateness must be the purpose of the model, the value of a component cannot be judged by technical or formal syntactic correctness only, but must be evaluated on the semantic and pragmatic level.

The **basic assumption** for the following explanations is that the purpose of a component within a model is similar to the pragmatics of an utterance in linguistics.

Since most computer scientists are not familiar with the linguistic concept of pragmatics, a short description may be helpful. In the semiotic trichotomy developed by Charles Morris, Rudolph Carnap, and C. S. Peirce in the 1930s, syntax addresses the formal relations of signs to one another, semantics the relation of signs to what they denote, and pragmatics the relation of signs to their users and interpreters (Levinson 1983, Mey 1993, MITECS).

**The central rationale for pragmatics** is that sentence meaning (semantics) in natural languages vastly underdetermines speaker’s meaning (intentions). The goal of pragmatics is to explain how the gap between sentence meaning and speaker’s meaning is bridged (Sperper 2003).

In “linguistics words” (which sometimes seem to me a little bit convoluted), pragmatic information concerns facts relevant to making sense of a speaker’s utterance of a sentence (or other expression). “The hearer thereby seeks to identify the speaker’s intention in making the utterance. In effect the hearer seeks to explain the fact that the speaker said what he said, in the way he said it”

(Bach 2003). Because the intention is communicative, the hearer’s task of identifying it is driven partly by the assumption that the speaker intends him to do this. The speaker succeeds in communicating, if the hearer identifies his intention in this way, for communicative intentions are intentions whose “fulfillment consists in their recognition” (Bach 1979). In other and much simpler words, pragmatics is concerned with whatever information is relevant, over and above the linguistic properties of a sentence, to understanding its utterance (Sperper 2003).

As an **example**, consider a mountain walk of an experienced climber and his friend, who has always stayed in flat land. During the walk the climber shouts “Stone” and expects his friend to seek for shelter. Unfortunately, his friend doesn’t even raise a hand. On which communication level occurred the error? We can assume that the flatlander heard what his friend said (transmission), understood the phoneme “stone” and mentally translated it into the correct word “stone” (syntactic level) and knew what a stone is (extensional meaning of the word, semantic level). Hence the fatal error must have occurred on the pragmatic level as an *failure of communicating the demand of action*.

It is obvious that the line between semantics and pragmatics cannot be absolutely definite and that some aspects of contextual information and other connotation could be placed into the semantic bucket, too. (In the example, one could argue that the semantic of the word “stone” in the context of mountain hiking has to be extended) But in general it is not recommended to extend the borders of semantics, because it quickly leads to person dependent ambiguity in semantic definitions (What if a geologist shouts stone during a mountain walk? Is he delighted or terrified?). It should be mentioned that even Noam Chomsky, the world’s most famous and influential linguist has stated that “a general linguistic theory must incorporate pragmatics as a central and crucial component” (Chomsky 1999).

However, taking the nature of pragmatics into consideration it is no surprise that it has been omitted in computers science. The general guideline in all natural and technical sciences is to reduce subjective factors down to zero. Hence scientists from this research areas seek to find or define a pragmatics-free (context and connotation free) experimental system. Unfortunately, that approach has seldom worked in human or social sciences or whenever human behavior and communication have to be regarded.

So far only the linguistic aspect of pragmatics has been discussed. The following sections change the focus to the relationship between models and pragmatics.

As an introduction to this relationship consider the definition of semiotic qualities of conceptual models (see Table 1) given by (Lindland et al. 1994).

Table 1: Definition of semiotic qualities of conceptual models (Lindland et al. 1994)

Syntactic quality	... is the degree of correspondence between a conceptual model and its representation.
Semantic quality	... is the degree of correspondence between the conceptual model and the real world.
Pragmatic quality	... is the degree of correspondence between the conceptual model and its (individual) interpretation.

The first connection between models and pragmatics is quite simple, but often underestimated. The standard situation of professional model development consists of a client who has a problem in a real world system which can't be investigated directly and a model development team who is charged with the task to solve this problem within a model. Since the clients view of the real world system generally differs from the view of the model developers, adjustments of both views are essential before starting to create a conceptual model of the real world system. We experienced this well known difficulty within our development teams, too. Therefore, from our experience, these adjustments together with proper model validation are keys to model quality (see Figure 4) (see Hofmann 2002). Generally, the adjusting of the different views of the client and the model developer, respectively, in our case, among the different model component developers is performed via natural language communication. Hence, the conceptual model can seldom be understood without taking into account the pragmatics of the communication.

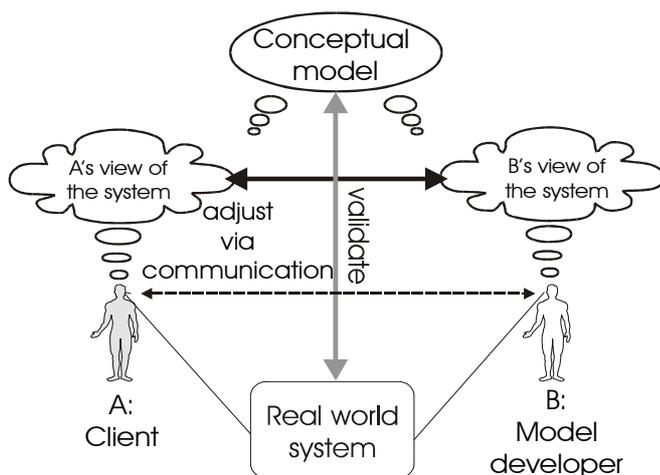


Figure 4: Adjusting personal views and validating a model

One of the central dogmas of modern computer science is the demand for unambiguous programs that **can be used without any additional context information**. Especially for component-based software architectures this requirement is said to be essential. Taking this dogma literally implies that documentation of programs mustn't be essential for model understanding and application, but only (extremely) helpful. Ideally the program/module itself (as a sequence of statements in a programming language) should contain the whole meaning/sense of the underlying (conceptual) model.

I do not doubt that from the perspective of software engineering this dogma is completely justified. There actually is a huge amount of software components that fulfill this black-box criteria. However, as far as I can see, these components are of a very fine granularity, and very often monofunctional. The simplicity of these components in terms of degrees of freedom is the reason why the black-box approach works. However, to base a general hierarchy of domain specific components - that finally would lead to complex multifunctional modules - on a black-box architecture is most probably an illusion of current software engineering. **In complex military, economic or logistic simulation systems the code vastly underdetermines the modeler's ideas and intentions**. Therefore, model documentation in natural language and additional verbal communication, despite all their disadvantages of ambiguity and connotations, are essential parts of the interaction among model developers and users. I am also convinced that the restricting of programming languages to syntax and semantics is an illusion, that has contributed to the software crises. Pragmatics as the part of semiotics that deals with the relation of signs to their interpreters must be included into the theory of programming languages, since reused models (programs) are means of communication between people, too.

## USEFUL COMPONENTS

The by far most useful things in more than twenty years of military simulation experience at the IASFOR have been **concepts to abstract and idealize reality** and algorithms extracted from this concepts - not necessarily implemented algorithms and not necessarily algorithms that could be reused without changes. Hence, what one really appreciates designing or improving a GCSS according to a model development process (Figure 5) are **well described and structured ideas of abstraction and idealization, which balance the model's need for simplification against the constraints of the system context and the imposts of the problem**. Sometimes it is even the documented system analysis preceding the design of a conceptual model which is the most useful thing of an older model. When developing complex simulation systems like GCSS, one starts with the model purpose (or the problem definition), minds the scope of the model and the user mode and chooses a global level of resolution for time and space. Afterwards, one has to find suitable

concepts for the modeling of the six (respectively seven) aspects of combat, fitting the unique combination of purpose, scope and general resolution. The relatively low degree of usefulness of *software* components is caused by the fact that they seldom fit into a new model without modifications. Additionally, if you try to find an appropriate model component for a new purpose, you will get lost with code. It will take you weeks before you grasp the idea of abstraction and idealization of the conceptual model from the executable model.

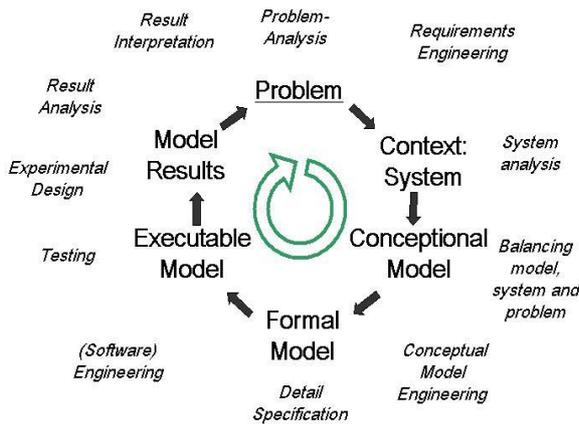


Figure 5: An idealized view of model development

### GRANULARITY

As a result of their work with the French simulations system ESCADRE, (Igarza et. al) have stated that the reuse of military simulations as a whole (lowest granularity) for a new purpose is almost always impossible. These experiences and opinions conform with our own results: As a general rule we have found that model components which include more than one of the seven aspects of combat are too specific to be reused for new purposes. Hence the lowest granularity level successfully applied in our GCSS represents the real world system with components that depict one aspect of combat. From the higher granularities (with a further break down of the one-aspect-components into smaller components (hierarchy of components)), the most successfully one discriminates at the level of entities, and there attributes. Weapon system specifications, cell specifications in grid terrain or priority definitions for target selection, for example, could be reused in a high resolution combat simulation system after 15 years without any major modifications (see (Hofmann et al. 1984) and (Hofmann and Hofmann 2001)). On the other hand there are some components nearly useless in new models. Most of them belong to an intermediate class of granularity that lies between entity level and the “aspect of combat”-level. As an example take support modules for the “command and control components” such as terrain evaluation modules, assessment of the own and enemy situation modules and

other estimation modules. Even slight modification of a model can devalue these modules completely.

### REPOSITORIES

As a consequence of our reasoning, repositories for combat simulations systems, which would be in fact very useful, should not be restricted to software component libraries like the C++ or Java libraries in the net (Repositories 2003). It would be more promising to assemble concepts and algorithms applied in combat simulations together with documentations of system analysis, model experiments and successful model applications. Such a repository takes the whole model development process (figure 4) into account. In order to organize a part of this repository we currently work with the classification scheme showed in table 12. As an illustration some examples are inserted as a catchword. Further explanations can be found in (Hofmann 2000; Hartman 1985, Olsen (ed.) 1994, and Farell 1989).

Table 2: Classification scheme for GCSS-concepts

	Aggregated theater level modeling	Aggregated corps/division and lower echelon modeling	High resolution bataillon/company and platoon modeling	High resolution single weapon system and single person modeling
<b>terrain and environmental representation</b>	vector graphics	large grid terrain representation	narrow grid terrain, Line-of-sight algorithm	3-D virtual reality algorithms
<b>movement</b>	vector optimization	Branch & Bound	Branch & Bound, A*, dynamic programming	hitherto interactive
<b>attrition</b>	Lanchester	Lanchester-differential-equations,	markov-chain based approaches	single shot models based upon hit probabilities
<b>transportation</b>	classic OR-optimization	classic OR-optimization	transport capacities	explicit transport amounts models
<b>communication</b>	connection matrix	extended connection matrix	terrain considering algorithms	line of sight-communication
<b>reconnaissance</b>	simple probability approach	sophisticated probability approach	glimpse, scan, continuous - models	Line of sight algorithm
<b>command and control</b>	Case-based reasoning	rule-based reasoning	OR-optimization, rule-based reasoning	rule-based reasoning

In addition the classification considers for what purposes, scales and user modes the concept of modelling/algorithm has been successfully applied, what resolution in detail (temporal scale, spatial scale, processes, entities, attributes and dependencies) has been used and what kind of implementations are available.

### CONCLUSIONS

The main results of our experience with components in ground combat simulation systems are:

- it is seldom possible to reuse domain specific multifunctional components as black-boxes,
- the successful reuse of domain specific components seems similar to a successful communication on all semiotic levels,
- concepts and algorithms are the key for successful reuse of components,
- the value of components from different granularity levels is very different, too,

- model components which include more than one of the seven aspects of combat are, generally, too specific to be reused for new purposes, hence the maximal amount of useful functionality within one component seems to be limited,
- repositories for GCSS should include products from all phases of the model development process and not only executable code.

Whether these results are transferable to other application domains or not is difficult for me to answer, but I am convinced that similar experiences must have been made in some other domains, too.

## REFERENCES

- Alexander, C.J.W.: *Notes on the Synthesis of Form*, Cambridge, MA, Harvard University Press, 1964
- Bach, K. *The semantics-pragmatics-distinction*, <http://online.sfsu.edu/~kbach/semprag.html> (Feb. 2003)
- Bach K. and Harnish, R. H.: *Linguistic Communication and Speech Acts*. MIT Press, Cambridge, MA, 1979.
- Baldwin C. and Clark, K.: *Design Rules: Volume 1. The Power of Modularity* MIT Press, Cambridge, 1999.
- CAE : Information about SIRA only available with permission: <http://www.cae.com/> or <http://offizierschule.de/hptzh/sira/> (March 2003)
- Chomsky, N.: *On the nature of pragmatics and related issues. (1999)* [http://cogprints.soton.ac.uk/documents/disk0/00/00/01/26/cog00000126-00/chomweb\\_399.html](http://cogprints.soton.ac.uk/documents/disk0/00/00/01/26/cog00000126-00/chomweb_399.html).
- Czarnecki, K. and Eisenecker, U. W.: *Generative Programming*. Addison Wesley, 2000.
- Dahmann, J.S., Kuhl, F. and Weatherly, R.: „Standards for Simulation: As Simple As possible But Not Simpler. The High Level Architecture For Simulation“, SIMULATION 71:6,7 p. 378-387,1998.
- Davis, Paul K., and Reiner K. Huber. 1992. *Variable Resolution Modeling: Issues, Principles and Challenges*, N-3400-DARPA, RAND, Santa Monica, Calif.
- Davis, P. K., Zeigler, B.: “Multi-Resolution Modeling and Integrated Families of Models” in: *Technology for the United States Navy and Marine Corps, 2000-2035: Becoming a 21<sup>st</sup> Century Force*; Volume 9, National Academy of Sciences, 1997
- Davis, P. K., Bigelow, J. (1998). *Multi Resolution Modeling*. RAND Corporation, Santa Monica, USA.
- Duchan, J. F.: *The Pragmatics Revolution 1975-2000*, University at Buffalo. <http://www.acsu.buffalo.edu/~duchan/1975-2000.html> (Feb 2003)
- Farrell, Robert L. 1989. *Remarks on Ground Combat Attrition Modeling*, paper presented at Army Research Office Workshop on Attrition Modeling in Large-Scale Simulations, February 2-4, Washington, D.C
- Flood, R. L. and Carson, E. R.: *Dealing with complexity: an introduction to the theory and application of systems science*, Plenum Press, New York. 1993.
- Green, G. M.: *Pragmatics and Natural Language Understanding*, Hillsdale, NJ: Lawrence Erlbaum. 1989; 2nd edition 1996.
- Hartmann, J.K.: *Lecture Notes in High Resolution Combat Modelling and: Lecture Notes in Aggregated Combat Modelling*. Naval Postgraduate School, Monterey (CA). 1985
- Hofmann, M.: *Zur Abbildung von Führungsprozessen in hochauflösenden Gefechtssimulationssystemen. Dissertation*, Universität der Bundeswehr München. NG Dissertationsverlag, München, 2000.
- Hofmann, H. W. and Hofmann, M.: On the Development of Command & Control Modules for Combat Simulation Models on Battalion down to Single Item Level” in: *New Information Processing Techniques for Military Systems*”, RTO Meeting Proceeding MP-049, Neuilly-sur-Seine, Cedex; Frankreich. 2001.
- Hofmann, M.: “Introducing Pragmatics into VV&A” in: *Proceedings of the 2002 European Simulation Interoperability Workshop (Euro-SIW)*, London, 2002,
- Hofmann, H.W., Litzbarski, S., Rochel, T., Steiger, K.: „Basis - Ein Gefechtsmodell auf Btl/Rgt-Ebene, Band 1: Beschreibung des Gefechtsmodells“. *IASFOR-Bericht S-8401*, Universität der Bundeswehr München, Neubiberg. 1984
- Hofmann, H.W., Rochel, T., Schnurer, R., Tolk, A.: KOSMOS - Ein Gefechtssimulationsmodell auf Korps-/Armee-Ebene, Band 1: Beschreibung des Gefechtsmodells. *IASFOR-Bericht S-9208*, Universität der Bundeswehr München, Neubiberg. 1992
- IABG: Information about HORUS, PAPST and KORA only available with permission: <http://www.iabg.de/> (2003)
- Igarza J.-L. et al.: „Development of a HLA compliant version of the French ESCADRE simulation support environment (SSE): lessons learned and perspectives“, Centre d’Analyse de Défense (DSP/DGA), Frankreich, 1998
- Kuijpers, N. van Gool, P. and Jense, H.: „A Component Architecture for Simulator Development“, TNO Physics and Electronics Laboratory, The Hague, 1998.
- Levinson, S. C.: *Pragmatics*, Cambridge University Press, Cambridge. 1983.
- Lindland, O. I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. *IEEE Software*, Vol 11, No. 2, March 1994, 42-49
- Mey, Jacob L.: *Pragmatics: An introduction.*: Blackwell, Oxford .1993
- Miller, G. A.: “The magical number seven plus minus two: Some limits of our capacity for processing Information”, *Psychological Review* 63, pp. 81-97, 1956.
- MITECS: Abstracts on Pragmatics (<http://cognet.mit.edu/MITECS/Entry/horn2> (2003)

- Olson, W. K. (ed): *Military operations research analyst's handbook, Volume 1: Terrain, unit movement, and Environment*, MORS, Alexandria, VA. 1994.
- Pötzsch, V. *Entwicklung und Implementierung von Modulen für die Prozesse Bewegung, Abnutzung und Aufklärung und Entwurf eines Rahmens für Führungsmodule für das Gefechtssimulationmodell COSIMAC. Diplomarbeit.* Universität der Bundeswehr München, Neubiberg. 1997.
- Repositories: (March 2003)  
<http://www.sei.cmu.edu/publications/documents/98.reports/98tr011/98tr011chap04.htm>  
<http://www.npsnet.com/danf/software/library.html>
- Schwierz K. (1995). Laborbetrieb IRIS. DASA-Dornier, Friedrichshafen.
- Simon, H. A.: "The Architecture of Complexity", in: Proceedings of the American Philosophical Society 106, pp. 467-482, 1962, reprinted in: idem, *The Science of the Artificial*, 2<sup>nd</sup> ed. Cambridge, MIT Press, 1981.
- Sperber: *Pragmatics-modularity-and-mindreading.*:  
<http://www.dan.sperber.com/pragmatics-modularity-and-mindreading.htm>
- STRICOM: (March 2003)  
<http://www.stricom.army.mil/PRODUCTS/JANUS/>
- Szyworski, C.: *Component Software – Beyond Object-Oriented Programming*. Addison-Wesley, 1998.
- Taylor, James G. 1983a. "An Introduction to Lanchester-Type Models of Warfare," in Proceedings of the Workshop on Modeling and Simulation of Land Combat, L.G. Callahan (ed.), Department of Continuing Education, Georgia Institute of Technology, Atlanta, Ga.; Taylor, James G. 1983b. *Lanchester Models of Warfare*, two volumes, Operations Research Society of America, Arlington, Va.
- Turner, Ken (ed.) *The Semantics/Pragmatics Interface from Different Points of View*. (Current Research in the Semantics/Pragmatics Interface, vol. 1). Oxford: Elsevier, 1999.
- Zeigler, B. P., Praehofer, H. and Kim, T. G.: „*Theory of Modelling and Simulation*“, ACADEMIC PRESS, San Diego, USA, 2000
- Zeigler, B. P. : „*Object-Oriented Simulation with Hierarchical, Modular Models*“, ACADEMIC PRESS, Boston, USA, 1993.

responsible for basic research in applied computer science (component based modeling, combat simulation systems, VV&A). He gives lectures at the University of the Federal Armed Forces (operations research).

## Author Biography

**MARKO HOFMANN** is Project Manager at the Institute for Technology of Intelligent Systems (ITIS), Neubiberg, Germany. After his studies of computer science at the University of the Federal Armed Forces in Munich he served two years in an army battalion staff. From 1995 to 2000 he was research assistant at the Institute for Applied System Analysis and Operations Research (IASFOR) at the University of the Federal Armed Forces. Since April 2000 he is

# SUBJECT VARIABILITY AND THE EFFECT OF STRESS IN DISCRETE-EVENT SIMULATION

Dr Andrew Belyavin  
QinetiQ Ltd  
A50 Building, Cody Technology Park  
Ively Road, Farnborough, Hampshire  
GU14 0LX UK  
E-mail: ajbelyavin@qinetiq.com

Anna Fowles-Winkler  
Micro Analysis and Design, Inc.  
4949 Pearl East Circle, Suite 300  
Boulder, Colorado, USA 80301  
E-mail: awinkler@maad.com

## KEYWORDS

Discrete-event simulation, human stressors, subject variability, thermal modelling, traits, states

## ABSTRACT

This paper sets out to address the problem of representing the impact of variability in human characteristics and abilities on overall performance in military systems. It is argued that a key element of the impact is the interaction between individual characteristics and environmental stress. An approach to modelling the variation of multiple characteristics is put forward using factor analysis. An example is calculated using anthropometric data, and the application of the approach is demonstrated using the Integrated Performance Modelling Environment (IPME) to model the variation in performance of a Surface-to-Air Missile (SAM) operator subject to different levels of thermal stress. It is concluded that under stressful conditions up to one third of the subject population may find the task too demanding.

## INTRODUCTION

This paper outlines the demands of modelling changes in human performance and variability in performance degradation due to both environmental factors and individual characteristics. The focus of the paper is on the problems of modelling human operators rather than modelling the behaviour of system components. A popular human factors approach to modelling performance is to use task analysis for the dissection of what the operator has to do, and then to simulate the system and operator elements together using task network modelling (Graine 1984; Hood et al. 1993). Examples of tools that use this approach are Micro Saint, a commercially available general-purpose discrete-event simulation tool, and the Improved Performance Research Integration Tool (IMPRINT) developed for the United States Army Research Laboratory for specialised workload and staffing analysis.

In parallel with the development of frameworks for analysing operator performance in systems, there has been an attempt to systematise understanding of cognitive and physical performance by defining

taxonomies and other models (Farina and Wheaton 1971; Roth 1991). By relating the effect of a stressor to particular task types, it is possible to construct a concise mapping from environmental stress to task performance, and thus model performance degradation. This is the approach used in both IMPRINT and the Integrated Performance Modelling Environment (IPME), a Unix-based discrete-event simulation tool. The effects of stressors have been examined with respect to overall system performance, but variability of system performance due to individuals has not been typically included in an analysis. This paper argues that stressors and variability of individuals *combined* are an important component of the variability of system performance.

To establish a strategy for the representation of the effect of stressors and variability on human performance in a broad range of model frameworks, the following three issues need to be considered:

1. The phenomena we are trying to represent
2. How stressor effects and human performance are currently represented
3. How present methods can be developed in the future

## THE NATURE OF THE PHENOMENA

The military context is noteworthy for the wide range of potential stressors to which personnel are exposed. A basic list of stressors, which aims to focus on the most important, contains 10 potential sources that should be considered (Belyavin 1999):

- Sleep loss fatigue/circadian effects and time on task
- Physical fatigue
- Thermal effects (thermal strain/dehydration/discomfort)
- Visual environment
- Fear/Anxiety/Morale
- Task demand – workload
- Noise (continuous and impulse)
- Vibration
- Hypoxia (loss of oxygen in high flying fast jets)
- High G (fast jets only)

Before the effect of environmental stress or variability can be defined, it is necessary to define what the operator(s) have to do, and specify metrics through which performance can be quantified. In observational

work, a task has customarily been defined as the smallest unit of operator activity with an observable output, and this definition has generally been retained in task network modelling. The metrics of task performance have then been defined usually as the time taken to undertake the specified task, and the accuracy with which the task has been executed.

This general approach has been employed in the modelling of human performance by identifying the *degradation factor* associated with the particular stressor, and applying it to the time taken to do the task. Variability has been modelled by making *time-to-perform* a stochastic variable. If suitable data are available, a similar approach has been used to estimate the effect on error, although the latter has proved more difficult in practice.

The stressors in the above primary list can be divided into two groups: those arising directly from the environment, such as heat, noise, vibration; and those arising from the context of the task, such as sleep loss, physical exertion or fear. The first group describe a direct change to the environmental conditions that influences human performance through a change in operator state. The second group includes stressors that are modified in a less direct manner, although their performance effects are also clearly mediated by a change in the state of the operator and may be moderated by individual traits.

Any description of the effect of both groups of stressors should at least recognise the change in the state of the operator implicit in the exposure to the stressful condition. Figure 1 shows how the sequence of cause and effect between environment change and performance change can be represented. Any change in the environment may be modified through the impact of individual characteristics at both stages of the process.

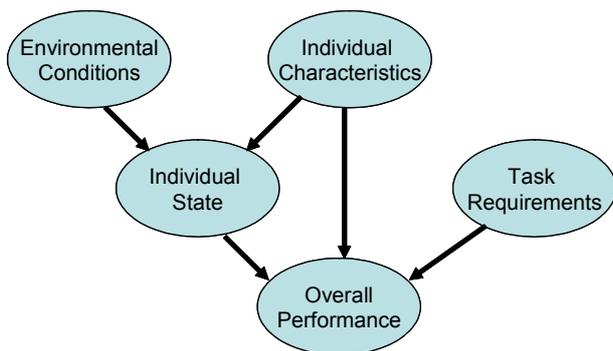


Figure 1: Sequence of Cause and Effect in Performance Changes

In the psychological literature, a similar approach to the analysis of the effect of stress on performance has been employed, although there is a tendency to identify a single state measure – arousal – rather than a multiplicity of state dimensions as the previous outline

suggests. Whether a single state can be employed to cover all possible stressors remains to be tested rigorously. What the scientific evidence indicates is that a sound predictive model of human performance under stress should be considered as comprising two stages: first a model of operator state, and then a model relating state to task performance (Belyavin 1999). Both stages of this model may be subject to variability in individual characteristics. For example, change in body temperature in response to a change in thermal environment is affected both by clothing and individual body characteristics. Additionally, the operator’s task performance as a result of having a high body temperature may be mediated by personality.

## THE IMPLEMENTATION IN IPME

The approach represented diagrammatically in Figure 1 has been implemented in IPME. IPME is a discrete-event simulation system with a graphical modelling interface used to predict human performance. It is a Linux-based integrated environment of simulation and modelling tools for answering questions about systems that rely on human performance to succeed. IPME focuses on the human, the tasks that the human performs in support of a goal, the environment in which the human operates, the stressors that affect human performance, and interfacing with external simulations.

An IPME system model is composed of four component models: Environment, Crew, Performance Shaping and Task Network. The Environment, Crew, and Task Network models formally represent the cascade displayed in Figure 1. The Environment model represents the factors that control the physical, mission, threat, and crew environment. The Crew model represents the human operators in the system, including the operator characteristics, which include states, traits, and properties. States are changing operator characteristics such as temperature or fatigue. Traits are non-physical operator characteristics that remain constant during a simulation execution, such as fitness, cognitive ability, or personality. Properties describe an operator’s physical characteristics, such as hands or eyes.

The Performance Shaping model contains functions that represent an operator’s ability to perform a task, based on operator states and traits, and impact task time or probability of failure. The Task Network model represents the system processes, including those performed by human operators, and relates the environmental factors, performance shaping functions, and operator characteristics.

It has long been recognised that human performance of an individual task is subject to stochastic variation, and this has been captured in task network modelling frameworks through the variability of individual task performance. In practice this variability comprises two components: *intra*-individual variability (variability in

individual performance from occasion to occasion) and *inter*-individual variability (variability between the performance of individuals within the target population) determined by individual characteristics. The second source of variability induces correlation between task performance for an individual for different tasks. Representing overall variability through independent variation of performance of different tasks does not reflect reality. The induced correlation is captured in the repeated measures model employed in statistical analysis.

A repeated measures model has been implemented in IPME to enable variability in system performance to be described. The Crew model has been modified to allow a sample of individual operator traits to be selected using a chosen joint distribution of operator traits. The remainder of the performance-shaping model has then been exploited to represent the impact of variation in traits on operator performance. As an example of the overall process, a preliminary model of the effect of variation in operator characteristics on performance under thermal stress has been implemented.

### THE STATISTICAL MODEL

Before implementing the repeated measures model in IPME, an appropriate model of the distribution of the characteristics in the target population was first constructed. It is clearly necessary to consider multiple characteristics simultaneously, so a multivariate approach is essential. Although it is relatively simple to develop a sampling methodology for a single variable, it is considerably more difficult to achieve the same goal for a multivariate population, since potentially complex interdependencies between the variables must be accommodated.

A well-established approach to the problem of describing a complex multivariate population is to reduce the dimension of the relevant space to a small set of fundamental factors and to derive the values of all the measures from the reduced set of factors through simple functional relationships. If the fundamental factors are distributed independently, the sampling problem is reduced to that of sampling from a set of independent univariate populations. A key step in the argument is the construction of independent factors from a larger set of interdependent variables. This is achieved through linear transformation only in the case of the multivariate normal distribution, where all linear combinations of the variates are normally distributed.

A procedure that achieves the goal for unimodal distributions of characteristics is as follows:

1. Test a sample for multivariate normality.
2. If normality is rejected, seek power transformations of the variables to normality, using the maximum likelihood procedure of Box and Cox (1964).

3. Confirm multivariate normality for the transformed variables.
4. Calculate principal components of the transformed variables
5. Retain a minimum sufficient set of principal components
6. Confirm multivariate normality for the retained principal components
7. Derive the best relationship between the original variables and the principal components by inverting the transformations

### ANTHROPOMETRIC CHARACTERISTICS

The procedure outlined in the previous section was applied to a set of three anthropometric variables used in the prediction of thermal strain: Body Weight (Wt), Height (Ht) and Mean Weighted Skinfold Thickness (MWST). These three measures are interrelated since both Ht and MWST affect Wt, and an approach to sampling the population must take account of the interdependencies.

The sample of data used in the analysis was drawn from the anthropometric study of aircrew characteristics conducted in the UK in 1973 (Bolton et al. 1973). As a first step, descriptive statistics were calculated for all three measures as shown in Table 1, and multivariate normality was tested using Mardia's (1970) measures of kurtosis and skewness. It was concluded that the population was non-normal and the initial variables should be transformed.

Table 1: Descriptive Statistics for Original Measures

Measure	Mean	Std. Dev.	Skew.	Kurt.
Ht (mm)	1774.0	62.34	0.050	3.159
Wt (Kg)	75.0	8.75	0.253	2.983
MWST (mm)	11.1	3.63	0.848	4.010

Power transformations of the measures were selected using the maximum likelihood procedure of Box and Cox (1964), and the following transformations were determined:

Ht No transform  
Wt Square root transform  
MWST Logarithmic transform

After confirming the normality of the transformed variables, principal components of the correlation matrix were calculated for the three-dimensional space. It was concluded that two components accounted for 94.1% of the variance. An orthogonal varimax rotation of the two components was calculated and the resultant components were standardised to unit standard deviation. The correlation matrix is displayed in Table

2, and the principal components are displayed in Table 3.

Table 2: Correlation Matrix Between Transformed Measures

	Ht	Wt
Wt	0.525	
MWST	-0.031	0.618

Table 3: Rotated Principal Components

	Factor 1 Loadings	Factor 2 Loadings
Ht	0.001	0.981
Wt	0.752	0.592
MWST	0.971	-0.078

The multivariate normality of the two-dimensional space was again checked using Mardia's skewness and kurtosis measures, and normality was not rejected. The best linear predictors of the transformed variables were then calculated and the formulae for reconstructing the original measures from the principal components derived. Constructing a sample from the tri-variate population is in this way reduced to generating a sample from a pair of independent normal variates in standard measure and applying the calculated relationships displayed in Table 4.

Table 4: Generation of Anthropometric data

Measure	Function
Ht	$1774.5 + 0.08237 * \text{Fac1} + 61.138 * \text{Fac2}$
Wt	$0.1 * (27.73 + 1.1987 * \text{Fac1} + 0.9434 * \text{Fac2})^2$
MWST	$\text{Exp}(2.354 + 0.3135 * \text{Fac1} - 0.02517 * \text{Fac2})$
Fac1 and Fac2 are independent normal variates	

In addition to the anthropometric data, there is evidence of individual variability in the metabolic cost of load carriage while walking. A good indication of the *expected* metabolic cost of movement while carrying load has been derived by Pandolf et al (1977). Recent work at QinetiQ indicates that there is individual variability about the expected value that can be described by a multiplier that is normally distributed about 1.0 with a standard deviation of approximately 0.09.

This model was implemented in IPME by generating two operator traits that were sampled independently, and deriving the values of the key traits from them using the relationships provided in Table 4. The scaling multiplier for metabolic cost of movement while carrying load was sampled as a third independent variable, and used to scale the metabolic cost of movement.

## EXAMPLE SYSTEM

The system that was used to investigate this prototype statistical model was a model of a Surface-to-Air Missile (SAM) system that depends on optical detection and recognition of an incoming target. In the SAM model, it is assumed that there is a command centre that detects targets and passes them to a system operator in the field. The system operator then must detect and identify the incoming target before the target is engaged. The physical work rate associated with the system was assumed to be relatively low, so an artificial scenario was constructed in which it was assumed that the operator had to march briskly carrying a 20 Kg load for 30 minutes before a rapid series of engagements was commenced. It was then assumed that, if the system operator's core temperature exceeded 38.5° C, the operator would rest. Neither the system nor the scenario is based on real systems, but they provide a basis for illustrating how the variability of anthropometric data can be applied in IPME.

The complete system model was constructed from three components:

1. Target Client: a simple application that generates a series of targets for the SAM system
2. Thermal Client: a whole body thermophysiological model (Higenbottam and Belyavin 1998)
3. IPME System Model: a task network model describing the operation of the system.

The task network model includes the model of operator variability. Both the Target and Thermal applications are client applications to IPME, communicating with IPME via a TCP/IP sockets protocol.

The target client supplies target position and speed to the IPME simulation. IPME informs the target client of missile launch, enabling the target client to manage potential interception. The thermal model client supplies the current thermal state to IPME, and calculates water loss due to sweating. IPME supplies environmental conditions, the operator's clothing state, and the operator's current physical work rate to the thermal model. A diagram of this relationship is displayed in Figure 2.

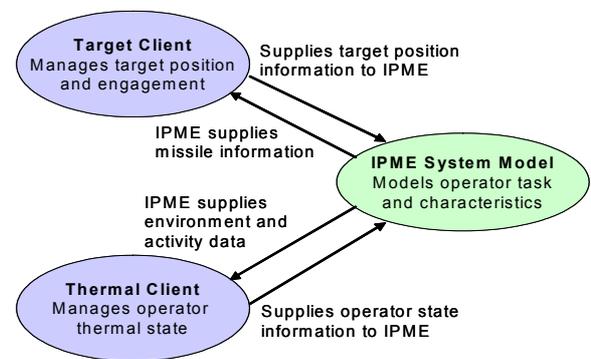


Figure 2: Relationship Between IPME and Client Applications

The task network model is displayed in Figure 3.

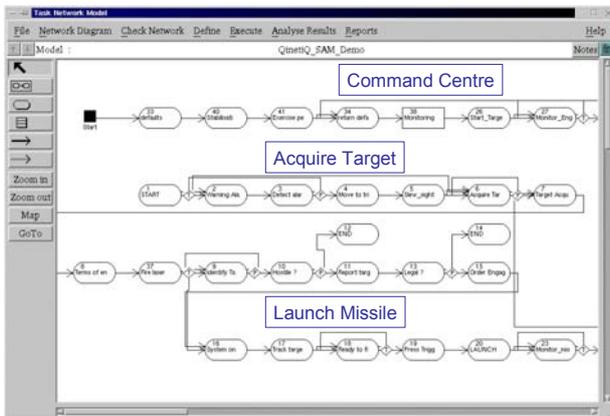


Figure 3: The SAM System Task Network Model

The system operator has to perform some physical activities to perform his or her task, as well as detecting and identifying incoming targets using an optical sight. To demonstrate the variability in system performance due to operator characteristics, a sample of 12 operators was generated based on the characteristics outlined in the previous section. Each operator was exposed to two conditions:

1. Dry bulb temp.: 26° C, Relative Humidity: 50%
2. Dry bulb temp.: 32° C, Relative Humidity: 50%

In both conditions the operator wore clothing with a thermal insulation of 1 clo. For the 30 minutes preceding the arrival of the targets, the operator walked briskly on level ground (1.75 metres sec<sup>-1</sup>), carrying a load of 20Kg. While engaging the targets the operator was assumed to be working at a steady 50 Watts.

There are two routes through which the operator traits can impact performance. The direct effects of the operator traits will be on thermal strain in response to the environmental and clothing conditions. The main consequence will be the need for the operator to rest if his core temperature reaches 38.5°C. A number of smaller effects of thermal strain and dehydration on task performance were included as Performance Shaping Factors in the model. These were based on those described in Belyavin (2000), and it was anticipated that they would influence the precise timings of interceptions if and when they occurred.

The performance of the operator in successfully engaging targets was taken as the overall measure of effectiveness. In addition, core temperature, skin temperature and sweat loss were measured. Preliminary analysis of the results from this prototype model indicated that, under the less stressful condition, all engagements that could be achieved were successful.

Under the more stressful conditions, 4 of the 12 subjects failed to complete the task, and thus failed to engage the last one or two targets. These subjects were the fattest and heaviest of the sample, and it would be anticipated that they would experience the largest thermal strain.

Under the less stressful condition, the mean core temperature reached 37.99°C at maximum, whereas, under the more demanding condition, the mean core temperature reached 38.45°C at maximum. This is consistent with the observation that all the operators continued to work in the first condition, but four stopped work in the second.

The precise timing of the interceptions varied from occasion to occasion but there was no evidence of an effect due to subject traits. It was concluded that in this particular example the “indirect” effects of thermal stress embodied in the Performance Shaping factors had relatively little impact on overall system performance.

## FUTURE DEVELOPMENT

The model described in the previous section is simple and the effect of the varying characteristics on system performance is direct. However, even this relatively simple model embodies a number of critical concepts: the use of clients to describe additional processes, the cascade from environment to task performance, and the variability of individual characteristics. These underlying principles are reflected in the IPME architecture by design; however, these design principles may be applied in a broad range of architectures, such as those used in the construction of Computer Generated Forces or other discrete-event simulation engines.

Although the example discussed in this paper directly relates to dismounted soldiers and thermal stress, these same concepts may be applied to other stressors, such as physical fatigue, and other operator traits due to the general and extensible implementation in IPME.

## ACKNOWLEDGEMENT

This work was funded by the Chemical and Biological Defence and Human Sciences Domain of the United Kingdom Ministry of Defence Corporate Research Programme.

## REFERENCES

- Belyavin, A.J. 1999. “Modelling the effect of stress on human behaviour.” In *Proceedings of the 8th conference on Computer Generated Forces and Behavioral Representation* (May), 481-487.
- Belyavin, A.J. 2000 “Modelling the effect of thermal stress on human performance.” In *Proceedings of the Military, Government and Aerospace Symposium. ASTC '00.* (April) 182-187.
- Bolton, C.B., et al. 1973. “An Anthropometric Survey of 2000 Royal Air Force Aircrew 1970/71.” FPRC/1327.
- Box, G.E.P., and D.R. Cox. 1964. “An analysis of transformations.” *J. Roy. Statist. Soc. B*, 26, 211-252.

- Farina, A.J., and G.R. Wheaton. 1971. "Development of a taxonomy of human performance: The task characteristics approach to performance prediction." American Institutes for Research Inc AIR-726/2035-2/71-TR7 (February).
- Graine, G.N. 1984. "HARDMAN: U.S. Navy's Answer to an Integrated Personnel Subsystem. Workshop on Applications of Systems Ergonomics to Weapon Systems Development." Volume 1, NATO, (April).
- Higenbottam, C., and A.J. Belyavin. 1998. "2-D Thermoregulatory model – description and validation." DERA/CHS/PPD/WP980204.
- Hood L.; K.R. Laughery; S. Dahl. 1993. "Fundamentals of Simulation Using Micro Saint." Micro Analysis and Design.
- Mardia, K.V. 1970. "Measures of multivariate skewness and kurtosis with applications." *Biometrika*, 57(3):519-530.
- Pandolf, K.B.; B. Givoni; and R.F. Goldman. 1977. "Predicting energy expenditure with loads while standing or walking very slowly." *J. Appl. Physiol.* 43:577-581.
- Roth, J.T. 1991. "A preliminary taxonomy for predicting the magnitude of stressor effects on human task performance." Applied Science Associates, (September).

## **AUTHOR BIOGRAPHIES**

**ANDREW BELYAVIN** is a Principal consultant in the QinetiQ Centre for Human Sciences, specialising in statistical analysis and human performance modelling. He moved to CHS on its formation in 1994, leading the Biometrics and Ergonomics group, providing consultancy on statistical problems of surveys and experiments, and contributing to the development of models of aspects of human performance, including occupational stress, fatigue, whole-body thermal response, and workload. In 1995, he became project manager for the Integrated Performance Modelling Environment (IPME). His current work includes further development of workload models, new methods for modelling and assessing the impact of command decisions within an organisational framework, and the development of manpower models. His e-mail address is: [ajbelyavin@qinetiq.com](mailto:ajbelyavin@qinetiq.com).

**ANNA FOWLES-WINKLER** is a Principal Software Developer at Micro Analysis and Design, where she manages the IPME project. She has a Bachelor of Science in Computer Science from the University of Maryland. Ms. Fowles-Winkler started working on the IPME project in 1999 as a software developer, and then became project manager in 2001. Her e-mail address is: [awinkler@maad.com](mailto:awinkler@maad.com).

# The Implementation of CGF-Oriented Helicopter Dynamic Model

Jianxiang Liu

Haozhi Li

Beijing Institute of System Engineering

P. O. Box 9702-19#

Beijing, 100101 China

## KEYWORDS

CGF(Computer Generated Forces) helicopter Dynamic model

## ABSTRACT

CGF(Computer Generated Forces) means a set of virtual entities which can be run autonomously in the distributed virtual battle field. The physical behaviors incarnating their motion are the basis of other higher intelligent behaviors, so the entities' dynamic model should be built. In this paper a generally simplified helicopter dynamic model and control system model is implemented, it has been applied to the simulation of CGF to change the entities' motion status, which can reduce the calculation complexity when the precision is not affected and prevent singularity from occurring by quaternion.

## 1 INTRODUCTION

Distributed Virtual Battle Field Environment (DVBF) is becoming the next generation infrastructure of military training. In DVBF, the operators of all kinds of weapon system simulators can collaborate and perform tactics rivalry through the network. In order to enrich the battle field environment and improve its verisimilitude, a lot of computer-generated virtual weapon entities called CGF run autonomously in distributed virtual battle field to provide rivals for the people involved in the rivalry, CGF entities' physical behaviors that incarnate their motion are the basis of other higher intelligent behaviors. So it is needed to build a model for their physical behaviors. Modeling for the virtual weapon system entities is mainly related to their dynamics and their control systems.

According to the characteristic of CGF, a generally simplified helicopter dynamic model is implemented in this paper, and an integral method is explored for the model when it is discretized. In addition, during the calculation, the occurrence of singularity is prevented by quaternion.

## 2 MODELING OF HELICOPTER DYNAMICS

To simplify the model of the helicopter dynamics, this paper only reckons in the air dynamic power and moment affecting the rotors, airframe and empennage of helicopter, and ignores the dynamic power affecting the other vanes

and the ground effect when the helicopter flights in low altitude. As shown in figure 1, Earth Axes  $O_d X_d Y_d Z_d$  and Aircraft Axes  $O_b X_b Y_b Z_b$  are defined, where  $O_d$  is a random point on the ground,  $X_d$  points to the east,  $Y_d$  points to the north, according to the right-hand rule,  $Z_d$  points to the vertical down;  $O_b$  is barycenter of the airframe,  $X_b$  points to the nose of the helicopter,  $Y_b$  points to the right of the airframe,  $Z_b$  points down.

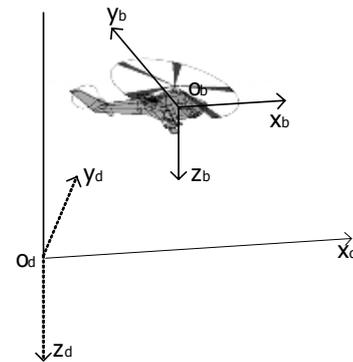


Figure 1 The Definition of Earth Axes and Aircraft Axes

In Aircraft Axes, the effects of the air dynamic power and the moment on the helicopter are:

$$\vec{F}_b = \vec{F}_b^R + \vec{F}_b^f + \vec{F}_b^{TR} \quad (1)$$

and

$$\vec{M}_b = \vec{M}_b^R + \vec{M}_b^f + \vec{M}_b^{TR} \quad (2)$$

where,  $R$ ,  $f$  and  $TR$  represent the rotors, airframe and empennage respectively. Thus the power and the moment of the rotors and empennage can be calculated by Blade Element Approach. In addition, the power affecting the airframe  $\vec{F}_b^f$  is simplified to a function of the velocity  $\vec{V}$ , i.e.  $\vec{F}_b^f = G(\vec{V})$ . The inputs of dynamics system are the collective pitch, the rotor longitudinal feathering, the rotor lateral feathering and the tail rotor pitch. (Please refer to [2] for the concrete calculation).

Assuming that  $u$ ,  $v$  and  $w$  are the speed values of the helicopter velocity  $\vec{V}$  in three axes respectively, we can get:

$$\begin{cases} \dot{u} = rv - qw - g \sin \theta + \frac{F_x}{m} \\ \dot{v} = pw - ru + g \sin \phi \cos \theta + \frac{F_y}{m} \\ \dot{w} = qu - pv + g \cos \phi \cos \theta + \frac{F_z}{m} \end{cases} \quad (3)$$

where,  $F_x$ ,  $F_y$  and  $F_z$  are the power value of the three axes respectively,  $\theta$ ,  $\psi$  and  $\phi$  are pitch, yaw and bank respectively.

When calculating the attitude of the helicopter, we only reckon in the moment value of inertia in the three axes. Assuming that  $p$ ,  $q$  and  $r$  are the angular rates of  $X_b$ ,  $Y_b$ , and  $Z_b$  respectively, then it can be obtained:

$$\begin{cases} \dot{p} = \frac{M_x}{I_x} \\ \dot{q} = \frac{M_y}{I_y} \\ \dot{r} = \frac{M_z}{I_z} \end{cases} \quad (4)$$

where,  $I_x$ ,  $I_y$  and  $I_z$  are the moment values of inertia in the three airframe axes respectively, and  $M_x$ ,  $M_y$  and  $M_z$  are the moment values of the three axes respectively.

According to the Euler equation, the three attitude angles  $\theta$ ,  $\psi$  and  $\phi$  can be calculated by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & -\cos\phi\sin\theta/\cos\theta & \sin\phi\sin\theta/\cos\theta \\ 0 & \cos\phi/\cos\theta & -\sin\phi/\cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5)$$

As shown in the equation (5), when the pitch value  $\theta$  is  $90^\circ$ , i.e.  $\cos \theta = 0$ , the angles cannot be calculated due to the occurrence of singularity. To avoid this case, a coordinate transition through quaternion is made as follow.

Assume  $B_{db}$  is the transition matrix from Aircraft Axes to Earth Axes, that is:

$$B_{db} = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} l_{11} &= q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ l_{12} &= 2(q_1q_2 - q_0q_3) \\ l_{13} &= 2(q_1q_3 + q_0q_2) \\ l_{21} &= 2(q_1q_2 + q_0q_3) \\ l_{22} &= q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ l_{23} &= 2(q_2q_3 - q_0q_1) \\ l_{31} &= 2(q_1q_3 - q_0q_2) \\ l_{32} &= 2(q_2q_3 + q_0q_1) \\ l_{33} &= q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{aligned}$$

$$\begin{aligned} q_0 &= \cos \frac{1}{2} \psi \cos \frac{1}{2} \theta \cos \frac{1}{2} \phi + \sin \frac{1}{2} \psi \sin \frac{1}{2} \theta \sin \frac{1}{2} \phi \\ q_1 &= \cos \frac{1}{2} \psi \cos \frac{1}{2} \theta \sin \frac{1}{2} \phi - \sin \frac{1}{2} \psi \sin \frac{1}{2} \theta \cos \frac{1}{2} \phi \\ q_2 &= \cos \frac{1}{2} \psi \sin \frac{1}{2} \theta \cos \frac{1}{2} \phi + \sin \frac{1}{2} \psi \cos \frac{1}{2} \theta \sin \frac{1}{2} \phi \\ q_3 &= -\cos \frac{1}{2} \psi \sin \frac{1}{2} \theta \sin \frac{1}{2} \phi + \sin \frac{1}{2} \psi \cos \frac{1}{2} \theta \cos \frac{1}{2} \phi \end{aligned}$$

The speed values in the three axes of the helicopter can be calculated by

$$\begin{bmatrix} \dot{y}_d \\ \dot{x}_d \\ \dot{z}_d \end{bmatrix} = B_{db} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (7)$$

where  $x_d$ ,  $y_d$  and  $z_d$  are the displacements of helicopter in  $X_d$ ,  $Y_d$  and  $Z_d$  respectively.

The three attitudes are given by:

$$\begin{cases} \theta = \arcsin(2(q_1q_2 - q_0q_3)) \\ \phi = \arccos\left(\frac{1 - 2(q_1^2 + q_3^2)}{\cos \theta}\right) \\ \psi = \arccos\left(\frac{1 - 2(q_2^2 + q_3^2)}{\cos \theta}\right) \end{cases} \quad (8)$$

In our implemented system, the ranges of the attitudes are limited as:

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}, \quad -\pi \leq \phi \leq \pi, \quad -\pi \leq \psi \leq \pi \quad (9)$$

In addition, in order to calculate the quaternion, we define a middle matrix  $\Omega$  as:

$$\Omega = \begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix} \quad (10)$$

Thus the changing rate of the quaternion can be calculated by:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \Omega \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (11)$$

and the quaternion can be calculated through integral when the equation is discretized.

### 3 CONTROL SYSTEM MODEL

Based on section 2 of this paper, the inputs of the dynamic model are the collective pitch, the rotor longitudinal feathering, the rotor lateral feathering and the tail rotor pitch, the power and the moment can be changed by these inputs. However, in the real operation, we cannot always get the

expected values immediately. These values must experience a transient process. In our system, we simulated the process through 1-order system, assume that the expected input increment is  $\Delta U_e$ , and the factual input increment is  $\Delta U$ , their relationship is:

$$\frac{\Delta U}{\Delta U_e} = \frac{1}{\tau s + 1} \quad (12)$$

Then  $\Delta U$  can be calculated by:

$$\Delta U = \Delta U_e \cdot (1 - e^{-\frac{\Delta t}{\tau}}) \quad (13)$$

#### 4 NUMERICAL ALGORITHM OF THE MODEL

Based on the previous analysis, the output of the dynamics system can be expressed as six 1-order differential equations as follows:

$$\begin{aligned} \dot{u} &= f_1(v, w, q, r) \\ \dot{v} &= f_2(u, w, q, r) \\ \dot{w} &= f_3(u, v, q, r) \\ \dot{p} &= f_4(q, r) \\ \dot{q} &= f_5(p, r) \\ \dot{r} &= f_6(p, q) \end{aligned} \quad (14)$$

In order to achieve high precision, the equation (14) are usually solved by the Runge-Kutta method, but it need much resources to calculate. In this paper, the dynamic equation of helicopter is based on the dynamic model used in the real-time simulator, because the CGF entities is different from the man-in-loop simulator, i.e. main-in-loop simulator requires high precision and perfect real-time characteristics, on the contrast, CGF requires lower precision and lower real-time characteristic. In order to simulate more than one entities in one computer, they should be built as "approximate real-time" simulators. The precision of the Runge-Kutta method is so high that the time cost is also very high. In fact, we found that the three attitudes  $p$ ,  $q$  and  $r$  are independent from each other through our experience, so the Runge-Kutta method is combined with Euler method in this paper, which can keep high precision and reduce the time cost.

Our method is: the differential equations of  $u$ ,  $v$  and  $w$  are discretized at first, then the Euler method is used for calculating the integral, and the 4-order Runge-Kutta method [7] is used for  $p$ ,  $q$  and  $r$ .

The concrete algorithm is:

1. Access the integral step  $\Delta t$ ;
2. Access the values of  $u$ ,  $v$ ,  $w$  and  $p$ ,  $q$ ,  $r$  of the last loop;
3. Calculate  $p$ ,  $q$  and  $r$  by the Runge-Kutta method;
4. Calculate  $u$ ,  $v$ , and  $w$  by the Euler method;
5. Calculate the transition matrix by the Euler method;

6. Calculate the three speed values ( $u$ ,  $v$ ,  $w$ ) and the three displacements ( $x_d$ ,  $y_d$  and  $z_d$ );
7. Calculate the three attitude angles  $\theta$ ,  $\varphi$  and  $\psi$ ;
8. Jump to step 1.

#### 5 IMPLEMENTATION / PERFORMANCE

The above-mentioned model has been applied to a system we developed in the past, Through a series of experiments, it was tested to be able to:

- i. Simulate the attitudes change of the helicopter correctly when it is flying. In the experiment, the attitudes of helicopter changes correctly according to the operator's manipulation, and it adapt to the disturb of the wind from the simulated atmosphere to keep its balance.
- ii. Satisfy the real-time requirements of the system, and collaborates properly with other entities. In the experiment, based on the model, the frame rates of the system can achieve 25 to 28 frames in one minute on PIII-450M, 38 to 42 frames on P4-800M, about 60 frames on P4-2G (all computers have been installed professional graphic cards). In contrast, when solving all equations by the Runge-Kutta method, the frame rates of the system can only achieve 7 to 10 frames in one minute on PIII-450M, 22 to 25 frames on P4-800M, less than 45 frames on P4-2G.
- iii. Satisfy the precision requirements in rivalry training, the precision of the three attitude angle  $\theta$ ,  $\psi$  and  $\varphi$  of the helicopter can be controlled to 0.1 degree, which make the virtual helicopter in the battle environment change its attitude very smoothly on the monitor.

#### 6 CONCLUSION

Aiming at the characteristic of CGF, one type of dynamic model for helicopter has been implemented in this paper, the Runge-Kutta method is combined with the Euler method in numerical algorithm to simplify the calculation as well as not to lower the precision. At the same time, it prevents the occurrence of singularity by using quaternion.

#### REFERENCES

- [1] Johnson, 1997. W. "CAMRAD II, Comprehensive Analytical Model of rotorcraft Aerodynamics and dynamics" Johnson Aeronautics, Palo Alto, California.
- [2] ShiChun Wang .The Dynamics of Helicopter. Beijing University of Aeronautic and Astronautic Science and Technology
- [3] Yamauchi. G.K.; Heffernan, R.M.; and Gaubert, M. "Correlation of SA349/2 Helicopter Flight Test Data with a comprehensive Rotorcraft Model." European Rotorcraft Forum, Germany, September 1986; *Journal of the American Helicopter Society*, Volume 33, Number 2, April 1988.

- [4] Chao Yang, XiaoGu Zhang. 1998. Analysis on the character of flight mechanics model of helicopter., *flight mechanics*.
- [5] KuYu Wang. 1991. The Flight Control System of Helicopter. The *LanTian Press*.
- [6] Harris, F.D.; Tarzanin, F.J., Jr.; and Fisher, R.K., Jr. 1970. "Rotor High Speed Performance, Theory vs. Test." *Journal of the American Helicopter Society*, Volume 15, Number 3.
- [7] Jilin Shi, GuiZhen Liu. 1996. Numerical Algorithm. *GaoJiao Press*.
- [8] GuoFong Pan, XinPing Zhao. 1999. The CGF in DIS . *Computer Science*.

#### **AUTHOR BIOGRAPHIES**

**Jianxiang Liu** is associate research fellow of Beijing Institute of System Engineering , located in FengTai district, Beijing, China. His research focuses on HLA, CGF and computer distribute computing. He can be contacted by e-mail at [13366093801@m165.com](mailto:13366093801@m165.com)

**Haozhi Li** is associate research fellow of Beijing Institute of System Engineering, His research focuses on Distributed Virtual Battle Field Environment and modeling, He can be contacted by e-mail at [mave\\_rick@21cn.com](mailto:mave_rick@21cn.com)

# TUTORIAL



# ACTUAL AND FUTURE OPTIONS OF SIMULATION AND OPTIMIZATION IN MANUFACTURING, ORGANIZATION AND LOGISTICS

Thomas Wiedemann  
Hightschool for Business and Technology Dresden  
Friedrich-List-Platz 1  
Dresden, 01069 GERMANY  
wiedem@informatik.htw-dresden.de

Wilfried Krug  
DUALIS GmbH IT Solution  
Tiergartenstrasse 32  
01219 Dresden, Germany  
wkrug@dualis.net

## Abstract

The paper is divided in a basic introduction and a discussion of new and future methods of optimization and simulation. The introduction discusses actual architectures of complex information systems, which include optimization systems or modules. The optimization system is presented as an example for a practical solution of optimization tasks. The second part discusses mainly Web-based applications.

## INTRODUCTION

Computer-based optimization has become increasingly important in the last few years. This demand is caused by a more difficult economical situation, where efficiency of all processes is more and more necessary. Optimization tools can improve the effect of other analysis tools like simulation systems or decision support systems some times. Although the whole task of analysis and optimization is more complex than before, the connection of both tool classes reduces the needed amount of time and personal staff dramatically. Instead of having a high-qualified simulation expert, a optimization tool can automatically evaluate simulation results and will try new scenarios. The results of optimization tools, like the number of the best strategy for the next week of operation, could be presented to traditional working staff without any additional teaching or difficult explanations. The fast development of computer power allows more practical and more complex tasks.

## BASIC METHODS OF OPTIMIZATION

There is a very large bandwidth of available methods and tools. Optimization methods can be divided into two large groups (see figure 1). On the one hand is the large group of continuous parameter optimisation methods, while on the other are the discrete optimisation methods (Krug ARENA/ISSOP Handbook 1997).

Within the broad spectrum of continuous methods, the first to be considered (and this is a point worthy of criticism) are the deterministic methods. These are also called discrete mathematical parameter optimisation methods, which can be applied to static, non-discrete, non-stochastic optimisation problems.

In the literature, these deterministic methods are often called hill-climbing strategies, because they method of searching for the optimum (maximum) is similar to a blind mountain climber, who tries to climb from a valley to the highest peak. For minimisation problems, the direction is reversed accordingly.

These methods currently dominate in the solution of technical optimisation problems. In the second column of Figure 1 the random methods are shown, which are becoming increasingly important in computer-based optimisation. They are used when the deterministic algorithms of column 1 are unsuccessful or unusable. These methods vary the values of the variables according to random, rather than deterministic rules.

Many deterministic optimisation methods, in particular those that require the gradient of the target function, can have convergence difficulties at points where the parameters have discontinuous derivatives. A narrow valley leads to the same problem when the finite step sizes are larger than the width of the valley. In this case, all attempts to obtain improvements in coordinate directions or by using local test steps to determine a new direction will fail.

Evolutionary algorithms are oriented towards results from observations of the natural evolution of living organisms. Using natural evolution as a basis for optimisation is justified, since it has been repeatedly shown that plants and animals have adapted themselves optimally to their environments. By contrast to deterministic or pure random methods evolutionary algorithms considers a set of solutions called a population. Each solution is correspondingly called an individual. In genetic algorithms, solutions are also called chromosomes. By analogy with biology, each component of a chromosome is called a gene. According to the nature, the values of this chromosomes are change like in a mutation and new population are generated by a recombination of chromosomes. The advantage of evolutionary algorithms consists in the combination of randomness and a stepwise selection of better population, were bad population can live also some specific time interval.

According to the endless number of optimization tasks, **there is no best algorithm**. For practical applications there should be a number of different applicable strategies !

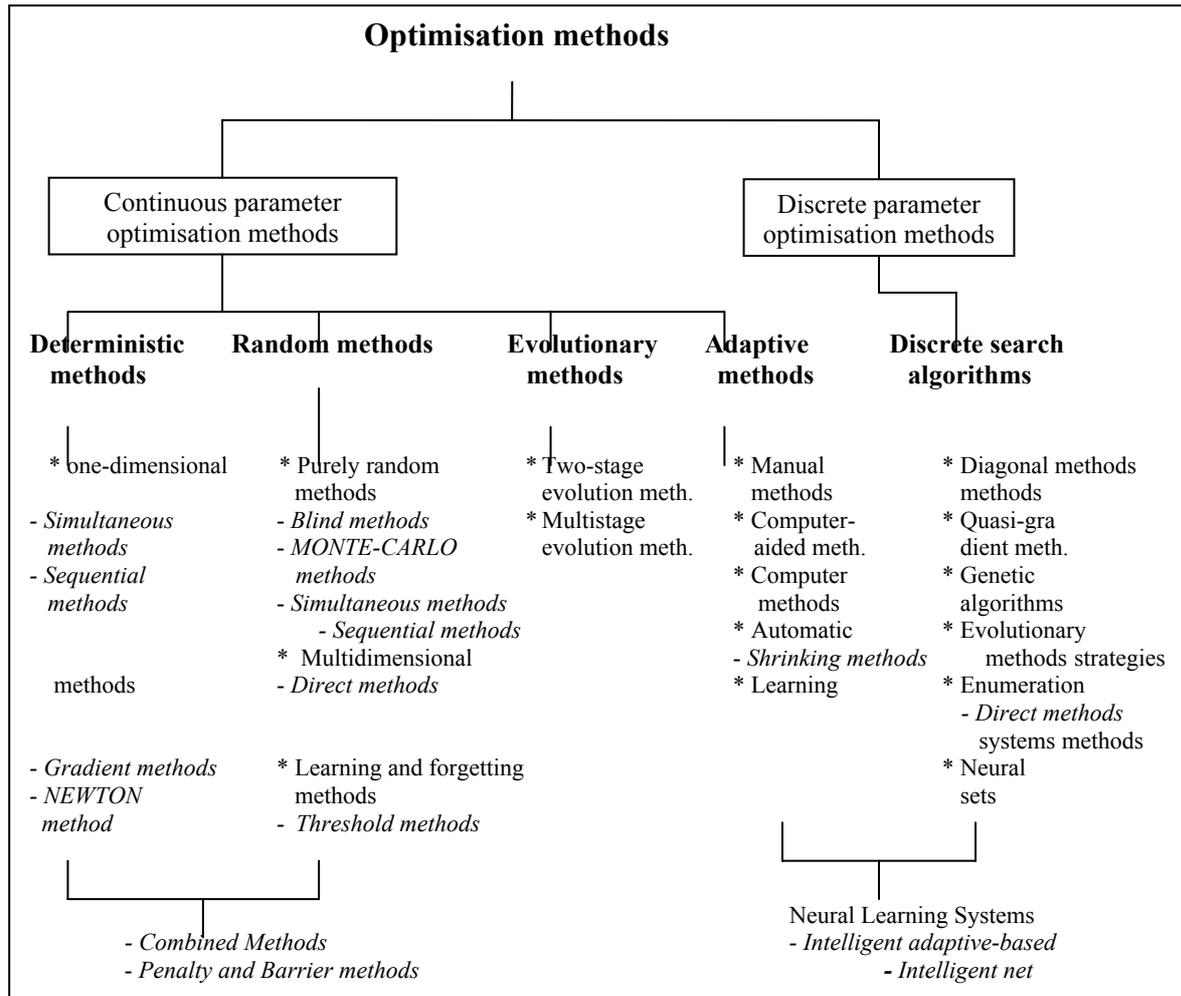


Figure 1: Classification of optimisation methods

Finding an optimal solution for a problem by building a model and running a simulation has always been the goal. Until now, however, it was usual to define the model a priori with certain parameters, and then to simulate, in order to see "what comes out", i.e. a human performs optimization by hand. He or she compares and evaluates results, fixes new parameters, and re-starts the simulation. This approach is very time-consuming, and the probability of finding an optimal solution in this way is relatively low.

With complex processes, which contain a large number of possible combinations of parameters and several mutually contradictory target criteria (costs, utilization, throughput etc.), it is practically impossible to perform a manual optimization. This is also true for manufacturing sequencing.

One solution is to use software with powerful optimization strategies which is coupled with the simulator. This optimization software must be able to access the model and to modify the values of model variables, read the simulation results that are relevant to the goals, and to determine the optimum (or a compromise).

## INDUSTRIAL APPLICATIONS OF OPTIMIZATION AND SIMULATION

Europe's Producing companies and especially Small and Medium-size Enterprises (SME) have to participate in dynamic networks and virtual factories in order to exploit (within alliances which are limited in time but not in distance) market chances that are hardly accessible for a single enterprise. Within such constantly self modifying environments the processes of organization, production and logistics have to be evaluated and optimized continuously. This requires the integration of comprehensive process models, efficient simulation and optimization tools as well as systems for Workflow Management, PPC, etc. via standardized interface. Various surveys have identified problems with the decision making process which are related to using methods that do not reflect the dynamic aspects of the manufacturing environment. The Paper aims to overcome these problems by producing integrated methods and tools, based on simulation, and optimization, to support effective decision making at all levels, from strategic to operational, of the company production, planning and control and business reengineering. The main theme is to support decision making

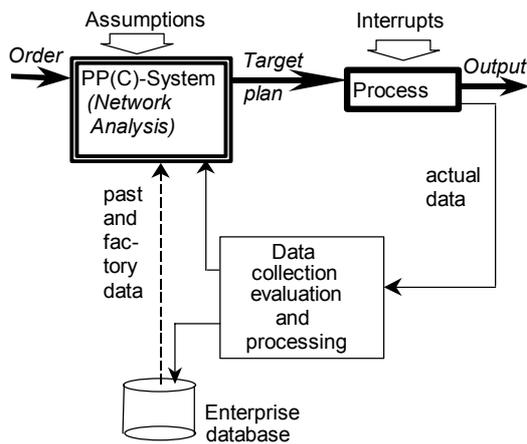
associated with the whole life cycle of products, but also included is evaluation of the impact of these decision making support tools on the personal using them, the organizational structure, and the Logistics management aspects of the company.

The Performance by integration of intelligent tools will be implemented in a software package. Existing state-of-the-art tools for PPC, analysis, simulation and optimization will be integrated into this software. The integrated systems SAP-R3 and ARENA/ ISSOP will be validated by solving actual problems in industry under virtual enterprise conditions.

### Typical software architectures

Today, a number of powerful software systems individual are available for most of the a afore-mentioned tasks, e.g. tools for

- Order processing
- Material management
- Production planning and control



**Figure 2:**  
Conventional production planning cycle without simulation support

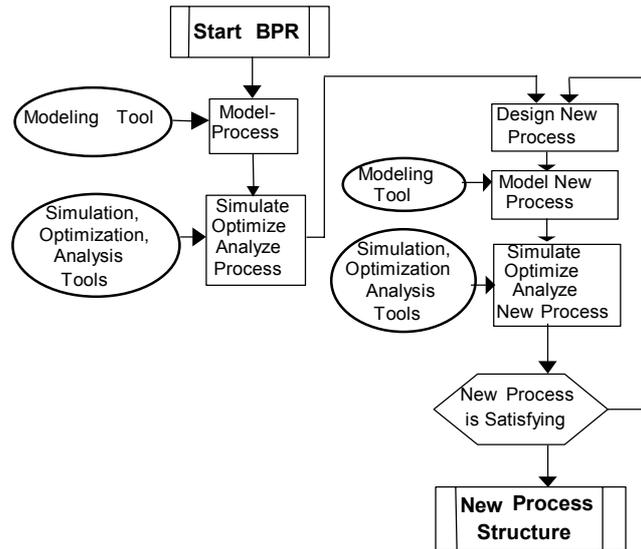
Complete product models, descriptions and related data may be exchanged using PDDI (Product Definition Data Interface), PDES (Product Data Exchange using), STEP (Standard for the Exchange of Product Model Data), EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport). In particular, STEP will be relevant for standard interfaces.

Actual PPC systems mainly use network analysis as a planning mechanism. Thus, temporal data like transportation time, machine preparation time, processing time, fault time etc. are modeled b

- Enterprise Modeling and analysis
- Workflow management and Standards
- Statistical evaluation of processes (e.g. average processing times, frequency of changes between different media or organizational units, process-oriented cost calculation etc.)
- Dynamic analysis (simulation and optimization) and visualization of processes.

Well known examples for such systems are SAP-R/3, ARIS, BAAN-DEM, STEP, CIM-OSA, EXPRESS, CRIMP, BIASS, ARENA, AUTOMOD, CIMPLE, ISSOP etc..

However, all these systems, though providing more or less elaborated interfaces for information exchange, are not systematically integrated so that they do not form a homogeneous platform for the user. Consequently, planning and business process reengineering usually is performed as shown in figures 2 and 3 .



**Figure 3:**  
Conventional BP improvement without integrated tool support

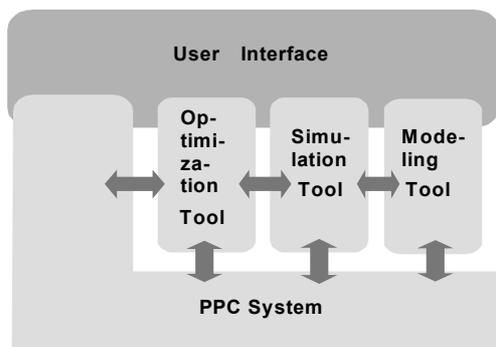
input values that enter the calculation as fixed entities, although they are, in reality, subjected to variations.

Network analysis tries to cope with that restriction by calculating the values each from an optimistic value, a pessimistic value and from the expectation (Erkollar and Mayr 1997). For reasons of simplicity, however, often only average values are considered. Thus, what actually happens in reality and the danger emerging from deviations are considered only insufficiently so that bottlenecks and delays are not always transparent to the planner.

To sum up, producing companies actually face the following problems in Manufacturing and Logistics:

- There is no common proceeding model for computerized modeling, simulation and optimization.
- There is only few flexibility in the available planning and controlling mechanisms.
- There are no integrated means for evaluation, analysis, simulation and optimization.
- There are no means for automated re-design and re-structuring of models based on optimization results.
- There are rarely means for the iterative improvement of process dynamics.
- The interfaces of the different tools are poorly coordinated and therefore do not allow for an effective integrated use.

Consequently, comprehensive process models are needed as well as a coupling of powerful tools for simulation and optimization in connection with automated mechanisms for Workflow Management, PPC, CAD, CAE etc.. A standardized interface might guarantee the integration that is necessary for efficient computer supported cooperative work within a dynamic network of SME's. Moreover, in order to achieve the goal of capturing and sharing the knowledge of the partners within a virtual enterprise reference models have to be used. Such reference models are available as results of fore-going projects.



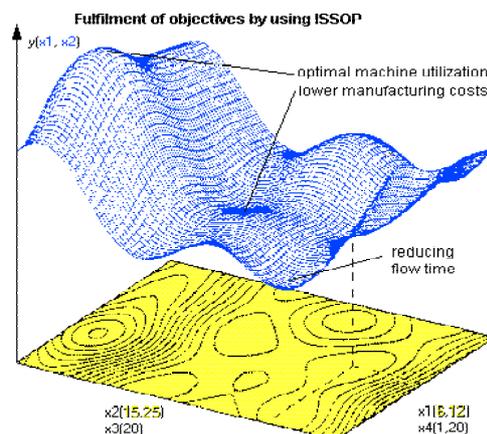
**Figure 4:**  
Interconnection and integration

Objectives (or aims) are variable process parameters of the simulation model that result from a simulation run, e.g. mean flow time of orders, machine utilization, manufacturing costs etc. After the objective are weighted, they are used as a substitute objective function.

## THE ISSOP-OPTIMIZATION TOOL

To achieve the level of system integration that SME's need for an successful participation in virtual enterprises/networks on a global market, the optimization tools ISSOP was developed under the following goals:

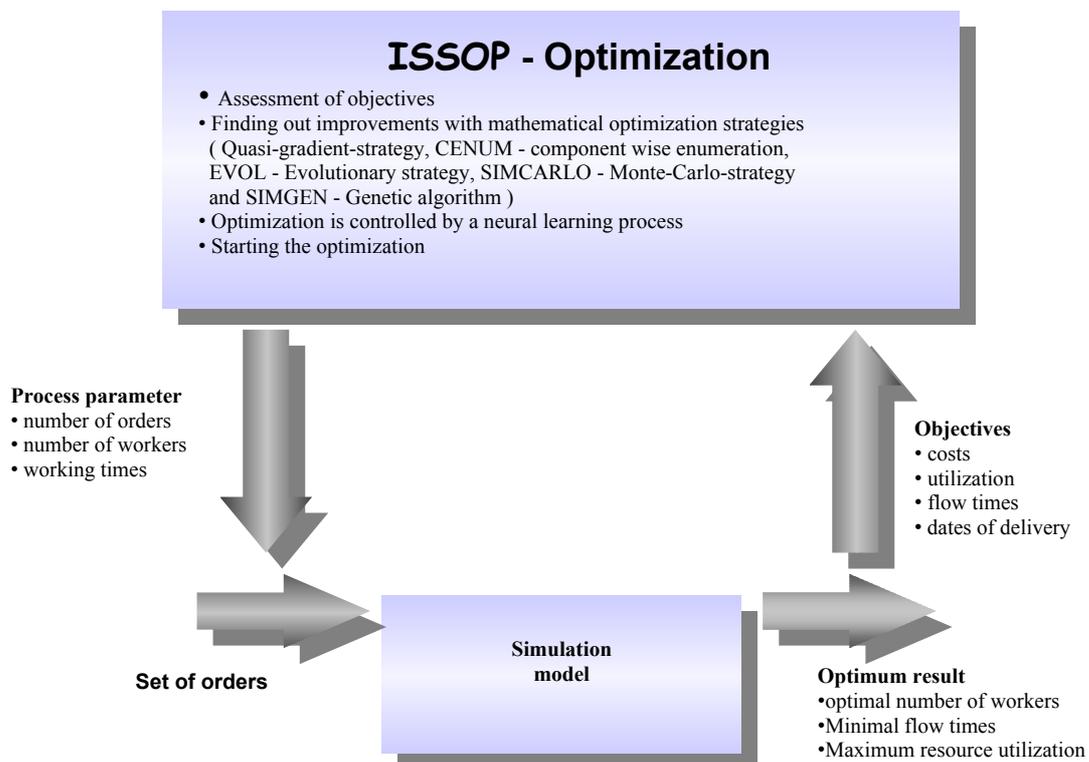
- (1) Integration of the reference models for distribution, procurement, order flow and production control into the context of the usage within an SME network.
- (2) Development of a Toolset which allows the coupling and exchange of data between existing subsystems for PPC, modeling, simulation and optimization (see figure 4). This interface will enable, among others, a (virtual) manufacturer to consider and exploit deviations that might occur during production in order to detect and implement an optimal solution for a given planning task. A particular step of this approach is the automatic extraction and transformation of the PPC's network into a simulation model in the format that is expected by the resp. system. The simulation results again are transferred to an optimization tool. The optimization results then are fed back into the PPC system (see figures 3 and 4).



**Figure 5:** Optimization Results in the Production Planing

- (3) Test and validation of our approach within a real live environment, i.e., by an application case study within an SME network. The improvements in speeding up production planning and control, in raising the quality of its results and in reducing costs will be measured and evaluated.
- (4) Development of new mechanisms for process optimization: The change of industrial business by the globalization of markets inside and outside of Europe requires a comprehensive reor-

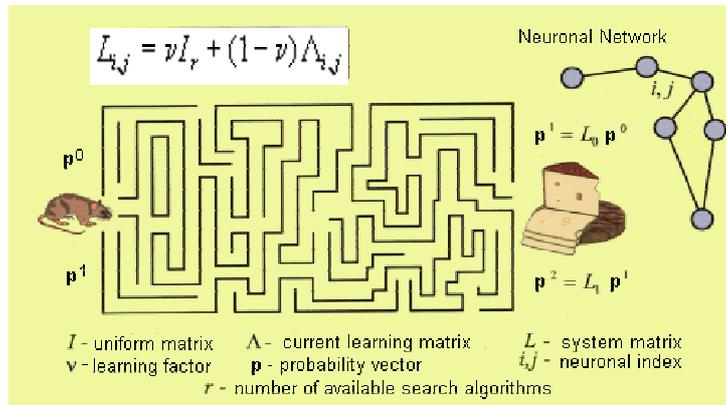
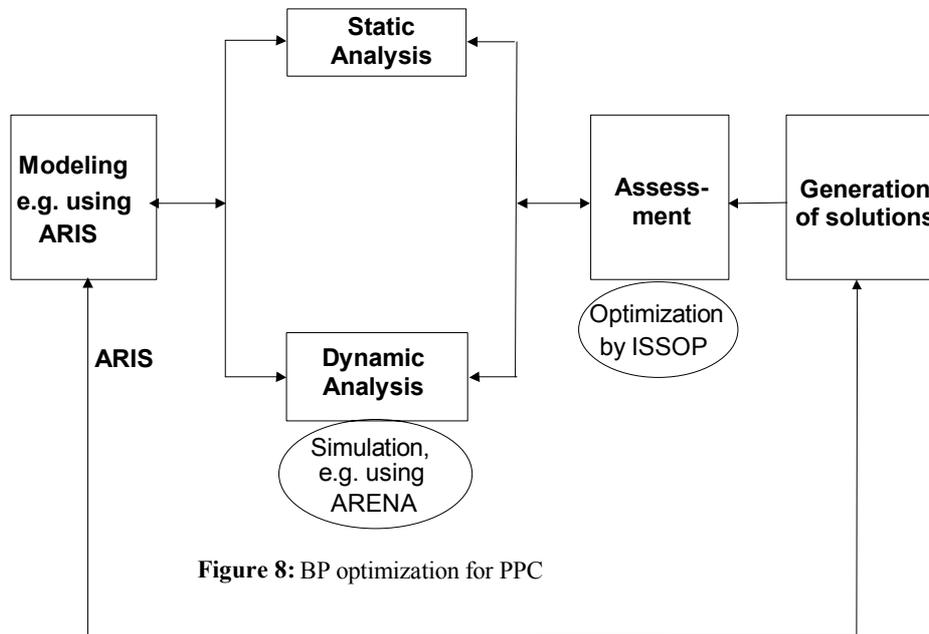
ganization of enterprise structures. In this context, an important aspect is virtual enterprise reengineering based on a static process analysis as well as on a dynamic analysis that consists in validating and optimizing the (virtual) order and business processes to the highest achievable efficiency (see figure 6, 7). The results gained here will be important for industry branches like metal-working industry, processing engineering, electrical engineering/electronics, etc..



**Figure 7:** Simulation and Optimization in Order Processing

The Simulation and Optimization of order processes in Manufacturing, organization and Logistics in focused of variation by number of orders, number of workers, working time etc. is seen in Figure

6. In the end on the optimization in coupling with simulation will be solved the new set of orders in connection with objectives of cost, utilization, flow time and dates of delivery.



The ISSOP learning process is based on a labyrinth problem similar to the biological systems that can be found in nature. Therefore the neuronal network will be solved on a mathematical basis solving the system matrix  $L$  illustrated on the picture below. From optimization problem to optimization problem in SME's better optimization results will be obtained by means of different optimization strategies. These strategies are implemented in ISSOP (see fig.7).

In addition to the business process models as such, the following aspects have to be considered:

- Process goals and objectives as well as evaluation criteria,

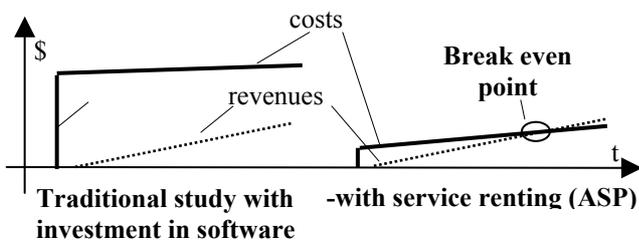
- Strategies for directed search and optimization in connection with a neuronal learning algorithm, seen in Figure 9,
- Representation of possible changes and variations within a process model (e.g. change of sequence, paralleling, change of resources assignment etc.),
- Reference models and module libraries to be used for generating new process structures,
- Representation of restrictions and logical dependencies (e.g. concerning the exchangeability of functions).

It is not intended or expected to reach a complete automation of business process development. The process designer rather will be provided with a means to select the most appropriate process out of a number of promising and tested alternatives.

## FUTURE OPTIONS BY WEB BASED SYSTEMS FOR OPTIMIZATION

Nearly all different requirements of simulation users can be transformed into basic terms of **profit** and **time**.

The profit is calculated as the difference between development costs of a simulation study and the expected revenues from the study. Development costs are influenced by the cost of the simulation environment and the modeling philosophy and comfort. Unfortunately the starting investments are on a high level between 5000\$ and 50.000\$ for typical simulation environments or external consultants. The revenue of a simulation study is unknown in the beginning. The risk of losing money rises with increasing investment costs. Web based information technologies allow new business models of using simulation services. Instead of a high starting investment in software, simulation services can be rent for a interval of time.



**Figure 10:** Costs of traditional and ASP-simulation

In traditional studies it takes a long time to reach the break even point. If the software is rented as a service, the starting investment is very small and equals the efforts for setting up the data interfaces and short teaching lessons.

The second term "Time" is important in real decision scenarios. Often a decision must be made under special circumstances like disturbance or external, unpredictable factors. The "**time to decision**", this means the time for finding a solution, is limited to some minutes or hours. This requires a very powerful simulation environment in terms of performance and optimization capabilities. More than one license and computer could be necessary. After the decision process is done, this environment runs idle until the next decision occurs. By using modern web technologies such free simulation capabilities can be used by other decision makers. The costs for simulation studies can be reduced significantly, if the peaks of required simulation power are averaged by a common pool of simulation resources, which are shared between different customers.

"**Portability**" and "**interoperability**" are often called the most important benefits of web based simulation. This is particularly correct for the current state of different computer environments. Depending on the existing hardware there are only two options - the simulation study is impossible due

to incompatible technologies or the systems allow data exchange and control. For the customer, portability plays the role of a "killer" question. If this question is answered positively, costs and time for decision making again become the most important benefits of web based simulation for the customer.

### Simulation areas with high efficiency of web based systems

The costs of developing complex simulation models are always very high. Although model generators and highly sophisticated modeling techniques could be used, the efforts for basic system research, data acquisition, model verification and validation actually are still connected with human resources. As it was mentioned before, the web supports operations with information distributing characteristics very well. Tasks with a high degree of creative work, resulting in synthesis of new web objects are still executed with external programs like HTML-editors or layout programs. This means - the internal structure and available functions of the current web are not ready for a real creative developing process ! This implies the following conclusions:

- The web can not simplify the modeling process in the near future. Simulation models should be developed by using traditional simulation tools.
- The web can support the reuse of existing simulation models very well by its distributed nature and the content management function like search machines and common data access protocols.

If we combine these conclusions, web based simulation systems will be of high efficiency, if the models remain nearly unchanged and only the simulation control variables and the input data are changed. Models of this kind are based on real systems with a fixed structure and dynamic working conditions, like

- flexible manufacturing cells with N machines from a set of M machine types, where the load is defined by external ERM systems,
- computer network systems with static network layout and dynamic routing strategies and random loads,
- fixed railway networks with changing time tables,
- nearly all serving processes with fixed stations and changing customer requirements.

In traditional simulation analysis such systems are modeled as "black box" models and the load is defined by parameters or in various data files. Even GPSS was used 30 years ago for defining such models. The power of this approach is determined by the quality and flexibility of the implemented interfaces for data exchange.

## REQUIREMENTS FOR SUCCESSFUL SIMULATION IN THE WEB

Concerning the actual deficits of tools for development of web based user interfaces, the efforts for model design and test should be minimized at the current time. The idea of Plug & play from computer hardware architectures will work also in web based simulation. One possible solution is a three level model repository and handling system:

- The first level provides very common, fixed models. Only external data files will change the behaviour of the models. The models are defined in the language of the used simulation system.
- The second level provides a library of predefined components. The client can define specific parameters of the components.

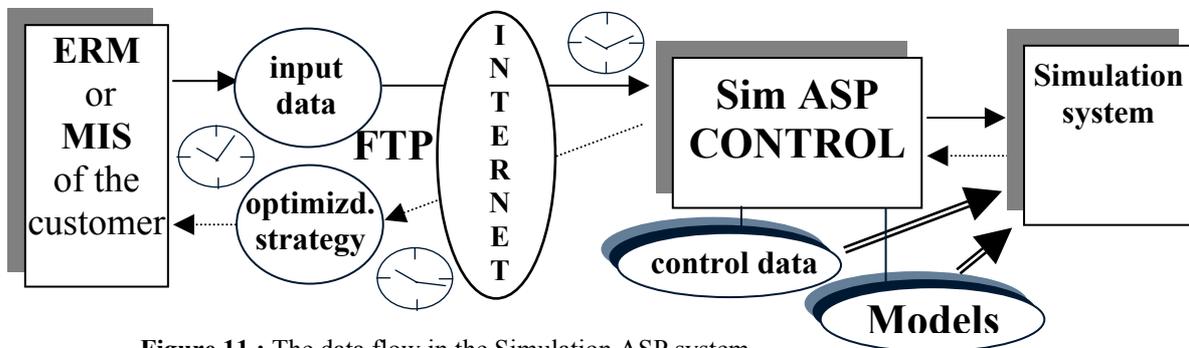


Figure 11 : The data flow in the Simulation ASP system

- This method is similar to well known component based systems like Arena or TAYLOR. Only the user interface and the number of forms and parameters are simplified.
- The third level allows a free definition of source code for the used simulation systems. The service of the ASP-system is limited to the execution and result analysis of the simulation.

### Automatic data exchange

In current web based simulation environments data exchange is often reduced to manual operations, like copying text into the source code of the model or extracting results from long trace lists. Compared to professional methods of data handling in data bases or data warehouses, this level of data exchange is not acceptable for professional customers. A efficient usage of web based simulation system requires a full integration in the common data flow of the enterprise. This integration can be made by time scheduled export and import routines in ERP-systems and the simulation environment. The FTP protocol can be used for the physical transport of the data files (see figure 11). Other protocols like HLA or CORBA are also possible, but require more development efforts.

### Result analysis with database functions

If the client is provided with a ERP, Data Mining or decision support system, result analysis of simulation runs is possible by importing the simulation trace files and using the integrated functions of these systems. Clients without powerful analysis tools depend on the functions provided by the web based simulation system. As demonstrated by the VisualSLX system (Wiedemann 2000) (Wiedemann 1998), this task can be solved by using databases for storing the results and calculating all aggregated values. The actual power of client-server databases also supports multi-model and multi-run comparisons. Visualization of graphical diagrams is supported by small Java applets.

### Fast and permanent access to simulation control functions

The first web based systems were often realized with CGI programs. The main disadvantages of this approach are low performance and a non-permanent connection to the simulation system. In result of the used batch mode, it was very difficult to control or interrupt a running simulation from outside. Information about the progress of the simulation was also hard to catch by CGI interfaces. A web based simulation environment should provide a permanent connection between the client and the simulation kernel. State information and control functions must be available during the whole time of a simulation run.

### A Simulation ASP-SYSTEM

Concerning the discussed requirements, an Application Service Providing (ASP) System for simulation was developed. Basic elements of this system are:

- a database for all model and simulation data,
- an object oriented modeling philosophy based on model entities and attributes,
- an universal code generator for converting the model description into a simulation program.

Details of the modeling philosophy and the code generator are presented in (Wiedemann 2000). The

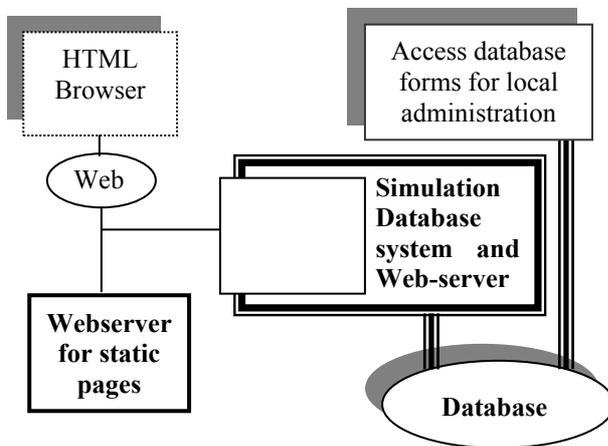
most important feature is the interface of the simulation environment to the web. In result of existing powerful software components for internet applications this interface is realized as combination of the VisualSLX database system and a web-server-component (see fig. 12). Here we have no CGI-interface or similar technology. Data-base related requests from the web are received by the Winsock-component in the VisualSLX/WEB-application and are answered immediately. Advantages of this web-server integration are:

- a very high performance in result of direct data-exchange and always open database tables,
- a long-time connection between the client and the server with continuous data flow during simulation or result processing .
- Actually the SLX simulation system is used as a simulation kernel. For all code templates are stored in the database, the same code generator is used for HTML files and simulation programs.

The system can work in three modes:

- as a traditional, local stand-alone system,
- as a multi-user database in a local network,
- and as a real client server system in Intranet or Internet environments.

The first two modes are realized by traditional database forms. Application specific forms can be developed in some minutes by using latest technologies of assistant supported database design (e.g. in Microsoft Access).



**Figure 12** : The architecture of the system

The system supports all three levels of Plug & Play models according to the discussed requirements. The supported simulation systems are SLX and SIMPLEX III. Other systems can be integrated without large efforts. Currently a integration of the Enterprise Dynamics is under construction.

## WEB SPECIFIC PROBLEMS

In result of specific characteristics of the actual Internet technologies we meet some typical problems.

## Web performance and bandwidth

Outside of the intranet the bandwidth is very often low and rapidly changing. In order to solve this problem we see a solution in parallel editing of more than one entry. For example all objects and their parameters could be offered in form fields at the same time. Checking operations are done with only one connection to the web server and all problems are reported at the same time to the user. A second option is the usage of more than one browser window and a interleave interaction mode of the user. The best solution would be a Java-based user-interface which performs all major operations at the client side. For time reasons this solution is planned for a later version.

## Multi-user lock problems

In the multi-user local database mode a database record is locked, when a user edits the content. During this time other users are able to see the record but they can not edit them. The Web is a system without defined sessions. Thus a user can switch off the computer or close the browser during an edit operation and the database receive no information about the loss of the connection. Similar to modern client server system this problem can be solved by a timeout of the lock mode. In the current database this lock operation and the check for timeout is done by VisualSLX.

## Run-Time interaction

Due to the delay of information transfer from the server to the Web, the state of the simulation system and the visualization at the web based client user interface may differ for some seconds. Thus the feedback for user interactions like Break or Stopping the simulation will take two times the interchange time (from the client to the server and back), which could be sum up to 10 seconds. The main solution is revealed in the further technical improvement of the Web or a usage inside a Intranet with guaranteed quality of service. A simulation at the client side is not very useful due to the bad performance of Java.

## License and security constraints

Web-based simulation also creates new requirements for software licensing and project management. Traditional software licenses of simulation packages only allow a single place usage. A web-based system must have the same license model like a network license. The payment of the simulation customers can be done per project or time. Another very critical fact is also data security. If a company uses a web-based simulation system possibly sensible data will be stored in an external database. In order to provide a safe simulation study some secret data could be decrypted with a public key of the company. A special encrypt DLL will be included in the import routines of the simulation model. The private key for decryption is

directly transferred between the customer and the decryption module. If the source code of the decryption DLL is validated by an external institution, the security level of the private input data for simulation will be very high.

A further improvement of security is possible by a **content scrambling** at the side of the customer. Sensitive data like customer names or product brands are replaced by random values. The conversion table is stored only at the customers computer. For the simulator there is no difference about working on a order from BMW or on data of F3234. After the simulation is finished, the results are decoded by using the stored conversion table at the customer side. Even if the network connection or the simulator is hacked by external intruders, there is no risk of loosing information, because all important data does not leave the customer system.

All security issues should be seen as very important decision factors for or against a ASP-system. If a potential customer only feels some possible security risk of giving his data in external hands, this could be a "killer fact" against the ASP-idea, where all technical and financial benefits will be without any relevance.

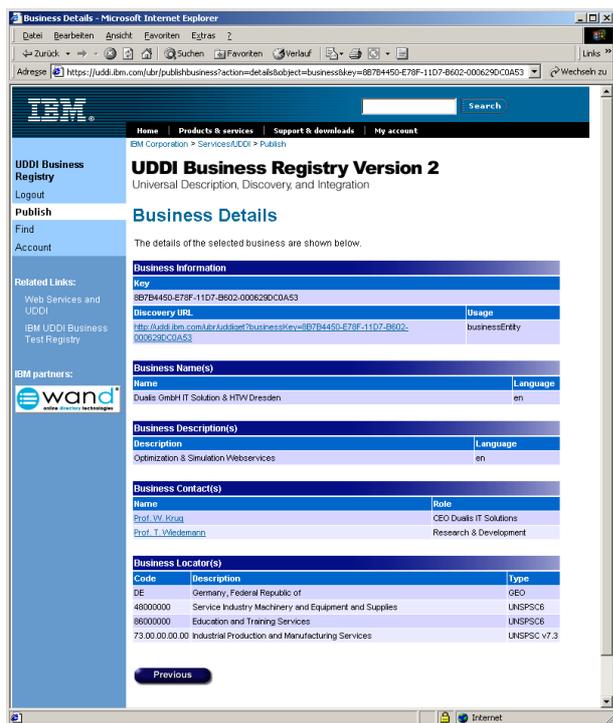


Figure 13: The official registration data of the optimization web service at **uddi.ibm.com**

## Web Service for Optimization & Simulation

Based on new web technologies like SOAP, direct connections between applications are now possible. Compared to traditional web interfaces with human oriented HTML-forms there are no GUI-elements. The applications communicate directly by transferring short XML-messages with the SOAP-protocol on different communication channels like TCP/IP socket connections, HTTP, FTP or Email (not recommended, but possible).

In the case of Enterprise Dynamics models, the network interface is realized by an special optimization ED-Atom. The atom includes an TCP/IP-socket and communicates with the remote optimization server. For the end user this atom acts like a traditional, directly included optimization module. The difference comes only in the pricing scheme – there is no need of an expansive investment for the optimizer, but only the effective time used for optimization is billed with 5 to 100 Euros per hour. Related to optimization effects of more than 1000 Euros per successful experiments this seems very interesting for smaller firms.

The above described WEB SERVICE was registered at uddi.ibm.com for world-wide-usage (see fig. 13 and 14).



Figure 14: The XML-based optimizer task description

## CONCLUSIONS

The success of optimization technologies depends on the combination of optimization tools with other Information Technologies for PPC, and Workflow Management Systems. Simulation- and Optimization Tools and Standard Interfaces will be changed continuously. This is the best strategy for getting a maximal synergy and a wider usage of optimization and simulation tools.

The perspectives of web based simulation will improve, if the current web technologies are used for a maximum of user friendliness instead of copying existing simulation systems to the web. Actually this approach will limit the capabilities of modeling large and complex systems. But the ease of use and the fast return of investments will turn this user driven approach into a very interesting way for improving and increasing the usage of simulation in commercial decision making.

## REFERENCES

- Erkollar, A., Mayr, H. C.: Simulation Aided Network Analysis in Production Planning and Control, In (Teo, Y.M. et al. Eds): Proc. WCSS'97, World Congress on Systems Simulation, Singapore, 1997, pp. 88-92.
- Fishwick, P.A. 1996. Web-Based Simulation: Some Personal Observations. In *Proceedings of the 1996 Winter Simulation Conference*, 772-779.
- Henriksen, J.O., 1995. An Introduction to SLX. In *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos. 502-509. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.
- Healy, K.J. and R.A. Kilgore, 1997. Silk: A Java based process simulation language. In *Proceedings of the 1997 Winter Simulation Conference*, 475-482.
- Krug W., Modeling-, Simulation – and Optimization – based Intelligence Business Process Engineering, in *Proceedings: Facilitating Development of Information and Communications Technologies for Competitive Manufacturing IiM 97*, Dresden, 1997, Edited by D. Fichtner and R. Mackay, Page 329-333.
- Krug 1997 : Intelligent Simulation- and Optimization system for Manufacturing, Organization and Logistic ARENA/ISSOP Handbook, 1997, Edited by SCS International San Diego U.S. 200 Pages and 100 Figures.
- Kuljis, Ray J. Paul, 2000: A Review of web based simulation: whiter we wander?, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, page 1872-1881
- Wiedemann. T., 1998. Sim-Mining and SimSQL - A database oriented approach for component-based and distributed simulation, *Summer Simulation Conference 1998*, Reno Nevada,
- Wiedemann. T., 2000. VisualSLX – an open user shell for high-performance modeling and simulation, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, 1865-1871

