

ESM 2004

SCIENTIFIC PROGRAM

PLENARY PAPER

USING SIMULATION TO PREDICT QUALITY AND COST IN THE AUTOMOTIVE BUSINESS

Dr. Stefan Greiner
DaimlerChrysler Research
Dept. RIC/AS
70546 Stuttgart, Germany
HPC T 728
E-mail: stefan.greiner@daimlerchrysler.com

KEYWORDS

System modeling, parameter estimation, cost-and quality analysis.

ABSTRACT

Simulation in industry has reached a very high popularity. Especially in the automotive industry, simulation is used very heavily when developing new vehicle concepts (e.g. crash test simulation). Typically, continuous simulation is used for these types of applications.

On the other hand, discrete simulation is becoming more and more popular, because many problems can be mapped to discrete models. Consider, for example, the question of how many breakdowns for a fleet of vehicles can be expected during their warranty/courtesy period. In this case, we have a continuous time frame and discrete events, namely the breakdown of a vehicle. Before we can analyze a system, we have to set up a model for this system/scenario. Therefore, we need a modeling paradigm that is both easy to understand and well suited for numerical analysis. At DaimlerChrysler, we use the modeling paradigm of stochastic Petri Nets, which we have extended in order to fit our specific modeling needs. These extensions result in models that do no longer have the Markovian property and therefore a closed system solution was not longer easily computable. On the other hand, discrete system simulation was perfectly suited to solve these types of extended Petri Net models. Before we can analyze the model, using simulation, we have to parameterize the corresponding model. In industrial applications, we have very often field data available which need to be analyzed statistically to obtain the corresponding model parameters. The parameterized model can then finally be analyzed to obtain the results.

Based on different research activities at DaimlerChrysler, the Petri net modeling and analysis tool *Expect* was developed.

During the talk, the modeling paradigm Petri nets is explained. Then, it is shown how to obtain model parameters from field data, and finally, an exemplary analysis is performed, using the analysis tool *Expect*. Different modeling and analysis examples are discussed and demonstrated. Together with numerical examples, an introduction to the analysis tool *Expect* is provided. To obtain the model parameters, the tool *ParEs* is used.

PETRI NET MODEL (GSPN)

A Generalized Stochastic Petri Net (GSPN) is a state-transition system, where the transitions are assigned firing times. The set of all possible states in the Petri net is called the state space of the Petri net. Only in the case where the Petri net consists only of exponential- and timeless transitions, the state space can be solved in a closed form by setting up the balance equations and solving them. In the case of a transient analysis we obtain a system of linear differential equations to be solved and in the case of a steady state solution, the system of differential equations is reduced to a linear system of equations.

Especially in industrial applications, we have very often non-exponential distributions assigned to the transitions. Consider for example the mileage behavior of a vehicle. In this case it turns out that the underlying distribution has a log-normal nature. Or, in the case of modeling the failure behavior of mechanical components, the underlying distribution usually shows some type of Weibull nature. We have either an infant mortality failure behavior (high failure rate at the beginning and then a decreasing failure rate over time), a wear-out behavior (low failure rate at the beginning and then an increasing failure rate over time), or a mixture of both types. Electronic components usually show a random failure behavior. Taking this into account, a numerical solution of the Petri net by solving the flow equations is usually not possible. Therefore, we turn very often to system simulation to

solve a Petri net to obtain the corresponding performance, reliability, quality and cost measures. In the following, we will introduce common types of distribution functions and how to obtain the parameters of a distribution function from field data in order to parameterize the corresponding Petri net models.

PARAMETER ESTIMATION

While stochastic Petri nets have become a widely used means for modeling complex systems, problems already arise in practical applications when the transitions in a net have to be parameterized. The goal of this paragraph is to introduce techniques, that allow the estimation of the parameters of several lifetime-distributions from field-data. These parameters serve then as input for the Petri nets. Besides the two- and tree-parameter Weibull distribution that are used in traditional reliability analysis, the exponential distribution, normal and logarithmic normal distribution and a new distribution-type, referred to as bathtub distribution, can be handled. The algorithms used for parameter estimation rely on traditional methods as regression and Maximum-Likelihood-Estimation, employing local and global optimization techniques.

THE WEIBULL FAMILY

The Weibull distribution, introduced by Waloddi Weibull in 1937 [Abe94] is the most frequently used distribution in reliability engineering. The original Weibull distribution has the distribution function (CDF):

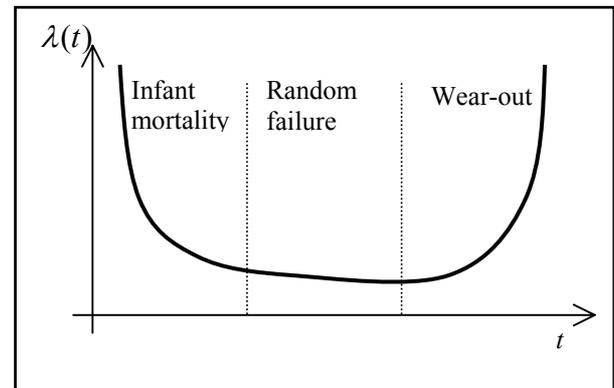
$$F(x) = 1 - e^{-\left(\frac{x}{\alpha}\right)^\beta}$$

with the two parameters α and β (which we therefore refer to as the two-parameter Weibull distribution). α is called the characteristic lifetime, which is defined as the point, where $F(\alpha) \approx 0.632$ (i.e. 63.2% of all units have failed). The parameter β is referred to as the shape parameter, because it defines, whether the distribution models infant-mortality failures ($\beta < 1$) or wear-out failures ($\beta > 1$). In the case of $\beta = 1$ we have the exponential case, i.e. random failure.

THE BATHTUB DISTRIBUTION

This distribution models the whole lifetime of a component with infant-mortality- random-and wear-out-failures. It is obvious from the previous paragraph, that this distribution can be built from the superposition (weighted sum) of two Weibull and one exponential

distributions. The name bathtub distribution is derived from the curve of the hazard rate of this distribution which gives the well known bathtub shape as shown in the following picture:



But doing this, one can see that, with one exception, the exponential distribution, which should explicitly model the stable lifetime, has nearly no influence on the shape of the distribution. The reason for this is that the exponential part is implicitly contained in the superposition of the two Weibulls. Therefore, for practical reasons we decided to leave the exponential part out to reduce the number of parameters to be estimated from eight to six (for clarity: this is not because we consider the exponential part as unimportant but because its explicit representation is superfluous). The CDF of the bathtub distribution is therefore given by the following equation (please note that the parameter p determines to what extent the current distribution function is subject to infant mortality or subject to wear out):

$$F(x) = p * \left(1 - e^{-\left(\frac{x}{\alpha_1}\right)^{\beta_1}}\right) + (1 - p) * \begin{cases} 1 - e^{-\left(\frac{x-x_0}{\alpha_3-x_0}\right)^{\beta_3}} & x > x_0 \\ 0 & x \leq x_0 \end{cases}$$

The two parameter Weibull is used to model infant mortality failures and the three-parameter Weibull models wear-out-failures and p ($0 \leq p \leq 1$) defines the weight of the terms (note: the index 2 has been omitted intentionally to remember the reader, that the explicit representation of the exponential part is missing). The exceptional case mentioned above in which the influence of the exponential part would be visible is the case if there are no wear-out-failures. But in this case, the second Weibull distribution can be used with $\beta_3 = 1$ to model the exponential part, such that this special case of a constant failure is also covered within the bathtub distribution.

THE NORMAL FAMILY

The distributions of this family, namely the normal and logarithmic normal (lognormal) distribution, are in general not used for the modeling of failure behaviour. The typical use of these distributions is the use as a mileage distribution, whereas the lognormal distribution has become the most commonly used distribution in this application area. The CDF of the normal distribution is given as:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$$

and that of the lognormal distribution as

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\ln(x)} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$$

respectively. Hereby, μ is the mean value and σ the standard deviation.

DIFFERENT TYPES OF FIELD DATA

In this paragraph we describe the different modes of data that have to be treated by the estimation algorithms. Most presentations of the algorithms shown later implicitly assume a so-called *full sample*, i.e. all parts fail till the end of the test and the exact times of the failures are known. In reality, this is very rarely the case. One has to deal with clustered and suspended data. Clustered means, that the exact failure times of an error are unknown but one only knows that there were for example k_i errors for parts with a lifetime between

x_i and x_{i+1} hours / miles. Suspended means, that some parts survived the end of the test, each of which with a certain lifetime / milage. Certainly, also the information about the suspended elements can be clustered, for example there were l_i parts with an operating time between x_i and x_{i+1} hours / miles, that did not fail. To put it in a nutshell, combining all possible modes of single / clustered data with failures and / or suspensions, there are six combinations that have to be considered in practice (the cases that contain only suspensions and no information about failures are not considered as all following estimation procedures need at least one failure for performing an estimation). The main advantage of using field data for analysis is, that one can be sure that the data reflect the behaviour of the part under investigation under real conditions. Thus, by using field data, the most critical point of a test rig series -- does the simulated stress reflect the real usage stress -- is circumvented. But this

convenience comes at a price: Usually, field data have very poor quality, such that the estimation algorithms must be able to work with this low quality, and commonly contains no information about the milage of the suspended elements, simply because one doesn't go to the garage if there is no error. Thus, in practice, the milage of the suspended elements has to be estimated, too.

LINEAR REGRESSION

In the following, the linear regression method for determining parameters of a distribution function is explained. Since the linear regression method is the simplest type of analysis techniques, we will focus on it. Other techniques as for example the maximum likelihood technique are not explained in detail. For more information on these techniques, the reader is referred to [BGdT98].

The basic idea of linear regression is to transform the CDF of the distribution into a linear form $y = a*x + b$, find a least-squares fit through the failures and finally calculate the distribution parameters from the straight line parameters. This is also the idea behind the well known probability papers. For example, in the case of the two-parameter Weibull distribution with CDF

$$F(x) = 1 - e^{-\left(\frac{x}{\alpha}\right)^\beta}$$

the linear form can be achieved by taking the natural logarithm twice. This leads to the linear form

$$\ln\left(\ln\left(\frac{1}{1-F(x)}\right)\right) = \beta * \ln(x) - \beta * \ln(\alpha)$$

Now, we define

$$\begin{aligned} y &= \ln\left(\ln\left(\frac{1}{1-F(x)}\right)\right) \\ a &= \beta \\ x &= \ln(x) \\ b &= -\beta * \ln(\alpha) \end{aligned}$$

and obtain the linear form $y=ax+b$. One of the problems in performing a least-squares fit to this form is, that one has only the x-values, i. e. the times of failure or suspension, but no y-values. Therefore one has to find an estimation for the correct plotting positions. According to [Abe94] median ranks, which have to be adjusted for the handling of suspended elements, are used for that. For the least-squares solution it is important that there is usually a large error in the time of failure, such that x should be taken as the

dependent variable in the fit. Finally, the back-transformation of the straight parameters to the distribution parameters in the two-parameter Weibull case is done by the formulas:

$$\alpha = e^{-\left(\frac{b}{a}\right)}$$

$$\beta = a$$

For the treatment of clustered failures and suspensions, there are different possibilities: For example, in [Lawl82] an algorithm for calculating the ranks in the clustered case is given. In practice, a simpler solution is possible: the failures or suspensions in a cluster are distributed either uniform or normal over the cluster (both versions have been implemented and tested but the differences are neglectable) and then the formulas for the single-data-case are used. Surely, this enlarges the number of data points that have to be considered in the least-squares estimation, but in practice the runtime of the estimation even with thousands of failures is within a few seconds.

The application of regression to the lognormal distribution leads to a numerical problem, as the linear transformation is

$$ierf(2 * F(x) - 1) = \frac{1}{\sqrt{2\sigma}} \ln(x) - \frac{\mu}{\sqrt{2\sigma}}$$

and therefore requires the efficient calculation of the inverse of the Gaussian error function. This can be done starting with the equality

$$erf(ierf(y)) = y \Rightarrow y - erf(ierf(y)) = 0$$

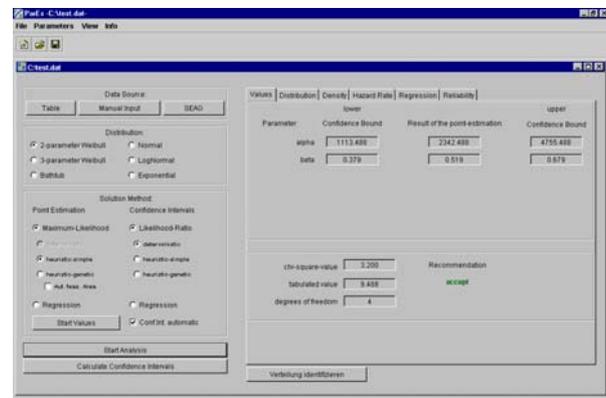
Defining $f(x_n) = y - erf(x_n) = 0$ we must find the zeros of $f(x_n)$ where x_0 can be approximated by one of the known total formulas for $ierf$. The solution follows the idea of Newton's Method [BuFa01] but uses a third-degree Taylor-polynomial instead of one of first order, which leads to faster convergence, and $erfc$ instead of erf for numerical stability.

Another problem is the use of the three-parameter Weibull distribution with regression as one has to estimate three distribution parameters from two straight-parameters. As the parameters for α and β can be easily estimated by the method described above, it is obvious to separate the estimation of x_0 . When using a probability paper it is a hint that there is an x_0 if the plotted points don't lie on a straight line. Hence it is adjacent to use the correlation coefficient, which indicates how well the points fit onto a straight line, for the estimation of x_0 , i.e. the failure times have to be shifted such that the correlation coefficient is maximized. A golden-section-search is used for

performing the optimization. The shift needed for the maximization is the estimation for x_0 and the estimation of the remaining parameters works on the transformed values.

Parameter Estimation Tool ParEs

This paragraph introduces ParEs, a tool for parameter estimation based on field data. ParEs (Parameter Estimation) is a GUI-based pure java-tool that incorporates all of the algorithms given above to give even the statistically non-experienced user the possibility to estimated parameters from field data and therewith to parameterize the transitions in his Petri net model. In the following, a screenshot of the tool is shown.



As datasources, the user has the choice between an ODBC-datasource, which is usually a database in the background, the manual input of failures and suspensions in a spreadsheet-like form or the selection of special car-components from a tree-structure. The user has the choice to pre-define a distribution function to be fitted or he can select the automatic detection option. In this case the system determines automatically which distribution function fits the given field data best. After the selection of the estimation-algorithm, the user may insert start values for the estimation, which are by default computed automatically, if someone wishes to fix some parameters to a special value. Finally, one can start either the computation of the point estimation and confidence-interval-calculation altogether or just the point estimation and the confidence-interval-calculation manually afterwards. The reason for this possibility is, that the calculation of the point-estimation is usually very fast while the computation of the confidence-intervals takes a lot of time, such that one should first be able to decide, if the distribution he guessed is correct before the lengthy part of the computation is started. To help the user in the decision if the supposed distribution is right, a Chi-Square goodness-of-fit test is performed, too. The final estimation results are presented to the user as the pure parameter values with the confidence bounds but also

the typical plots of CDF, PDF, hazard-rate or the regression-straight line (if regression has been used) are provided. Finally, the resulting parameters are also provided as tagged data to facilitate the use of the component for example by the Petri net tool Expect.

The idea behind this data export facility is, to make the whole parameter estimation process completely invisible for the user in the case of input modeling. Because in this case, users are usually not interested in and are not familiar with the parameter estimation process and its special details but just need appropriate parameters for their model. For example, if one wishes to use a transition in a Petri net to model the failure behavior of a special part, one just has to select this part and need not to know anything about the parameters. All the rest (parameter estimation, data handling and model parameterization) is handled by the tool. For this reason, a whole web based system environment has been set up and ParEs is part of this environment. This environment has been developed in Java and is constantly extended.

In the following, we will demonstrate the accuracy of the methods given above. For a better judgement of the results, no real data were used but data that were generated by a random number generator for the given distribution. For all examples, the confidence level is set to 95%. In the Weibull example, 50 failures have been generated while in the bathtub case 500 failures were used. At a first glance, this seems to be only a few input data if one considers the fact, that in the case of real field data thousands of real vehicles are considered. But, on the other side, one has to consider the fact that the data used in this example are single data while in the case of real field data we have data clusters. Each of these clusters can contain thousands of data points. Thus, even with a small cluster size of only 1000 miles, 500 clusters would define a range from 0 to 500.000 miles which will surely cover the whole lifetime of a vehicle.

The time for the evaluation of the likelihood function depends linear on the number of failures and suspensions in the case of single data and on the number of clusters, otherwise. Thus, one would expect that enlarging the size of the sample will always result in an enlarged runtime in the same order of magnitude. Practically, this can be taken as a worst case estimation because a larger sample usually contains also more information. Thus, fewer iteration steps will be needed if MLE is used such that the increase in runtime will be sub-linear in the average case. As in the regression case, there is a closed form solution where the runtime is within a few seconds even with thousands of failures.

One question that is often asked is, how many failures are needed to perform a reasonable estimation. If using MLE, theoretically one typical failure is enough if there are suspensions. But, if x_0 of the tree parameter Weibull distribution has to be estimated, too, at least 15 failures are needed. Practical applications show that one can expect reasonable estimates if there are at least

about 20 failures. But using these values, one has to keep in mind that this always means *representative* failures for the failure mode under investigation and not only random failures.

The first example to be demonstrated is a three parameter Weibull distribution. The following table shows the result of the estimation with the confidence intervals for local optimization with heuristic confidence interval calculation and regression. Hereby, α_l denotes the lower confidence bound for the parameter α while α_u denotes the corresponding upper confidence bound.

Parameter	True Value	MLE	Regression
α_l	---	137407	95435
α	150000	151029	150969
α_u	---	167381	279169
β_l	---	1.572	2.620
β	3.2	3.092	2.710
β_u	---	4.542	2.801
x_{0l}	---	0.736	44859
x_0	40000	40494	44859
x_{0u}	---	69144	44860

The computation of the above results took 18 seconds (point estimation: 3 seconds, confidence interval calculation: 15 seconds), while the regression needed only 4 seconds. There is only a slight difference in the upper confidence bound of β if an evolutionary strategy or the deterministic version is used for calculation. But, in this case the calculation time increases to about 5 minutes (compared to 15 seconds as before).

In the second example we apply our analysis algorithms to a bathtub distribution. The corresponding results are shown in the following table.

Parameter	True value	Lower Bound	Point Estimation	Upper Bound
α_1	3000	981	2026	4707
β_1	0.5	0.38	0.52	0.7
α_3	80000	79378	79944	80548
β_3	4.2	3.64	4.16	4.789
x_0	65000	62562	64645	66032
p	0.15	0.11	0.16	0.22

The computation time for the point-estimation was 29 seconds with the penalty method, and 4 seconds with the simplex method. The confidence intervall calculation took 241 seconds.

THE ANALYSIS TOOL EXPECT

The analysis tool Expect has been developed at DaimlerChrysler [HeGrHo02] and is designed for a broad range of applications, including safety, quality and cost analysis. Safety analysis is concerned with the design of safe automotive systems, including their interactions with humans. In the area of safety analysis, system measures as for example reliability, availability, load and throughput of system components and probabilities for critical states, are determined. Quality analysis is concerned with building models at the component, system and vehicle levels for obtaining information on product quality. One example of this is comparing the behavior of systems built using different components. Cost analysis allows, for example, the calculation of price models for maintenance and service packages, which depends on the level of service offered, as well as the age and mileage of the vehicle. In addition, statistical methods are used to predict the expected quality and cost into the future. In all cases, the goal is that these analyses can be performed as fast and as comfortably as possible, while covering a large number of variants, in order to study and compare different real-life scenarios.

The analysis tool Expect was designed with the goal of offering the modeler an easy to use and understand modeling environment and on the other side, allowing for an integrated system simulation, using discrete time event simulation. In the following, a short excerpt of the capabilities of Expect is described.

TOOL FUNCTIONALITY

The modeling and analysis tool Expect consists of a graphical editor, a simulator and a visualization component. It is implemented in Java in order to ensure platform independence, and also to make use of its special language features. One example is the dynamic loading of classes, which makes it possible to include functions in the Petri net model which are formulated in Java syntax, which can then be compiled and executed at run-time. This means that such functions execute with the same level of performance as the tool itself, since expensive parsing routines are no longer necessary. Furthermore, the Java compiler and interpreter are available everywhere free of charge.

The Expect graphical editor is designed with multi-document capability, allowing several nets to be edited simultaneously. This in turn allows sub-nets to be copied between models and for these to be compared quickly. Since Petri nets can quickly become very complex and unwieldy, Expect also allows hierarchical modeling. A net may be divided into sub-nets, which may be edited separately and linked together via transition or place interfaces. This supports both logical model development on the one hand, and clear graphical presentation of large models on the other. Expect contains a large number of configurable

parameters; model parameterization is supported by dialog windows for each net component, which include plausibility checks for parameter values. All net components, with the exception of arcs, have unique names.

The visualization component of Expect can be started directly from the editor. This allows the user to study the behavior of the net by watching the token game. This is important for demonstration and debugging purposes. The visualization module contains continuous speed settings as well as a step-by-step mode. The enabling state and enabling times of timed transitions are visualized in order to further enhance understanding of the net's dynamic behavior.

The simulator may also be started directly from the editor. Both simulation up to a specified point in time or up to an absorbing state are permissible. Furthermore, a number of replications can be specified, and the simulator will provide appropriate statistical results. These include the values of the user-specified rewards in addition to the standard measures for places (probability of being non-empty, average number of tokens) and transitions (throughput and probability of being enabled.) In addition to this functionality, it is also possible to perform transient analyses by parameterizing the range and step size of solution time points. The simulator will then compute statistical values for each of these.

CLASS OF NETS SUPPORTED

Expect supports a very general class of stochastic Petri nets. These include many additional features which greatly enhance the usefulness of the tool to the modeler.

Places can be assigned an initial marking and a maximum capacity, which may either have a constant value or be defined as a function of the current state of the net. An otherwise enabled transition is disabled, if, by firing, the maximum place capacity would be exceeded.

Both timed and immediate transitions are supported. Immediate transitions can be assigned a weight, whereas timed transitions can be assigned a firing time distribution chosen from a large set of alternatives. Firing time distributions range from exponential and phase-type distributions to Weibull and Normal distributions. In addition, the parameters for the firing probabilities and distributions may be defined as marking-dependent functions. Expect also allows the use of marking-dependent guard functions and priorities to control the enabling of transitions.

Transitions may be of single, multi- or infinite server type; multi-server and infinite-server transitions are treated by the simulator by assigning multiple firing times to the transition in accordance with the current enabling degree. Transitions may also be assigned a

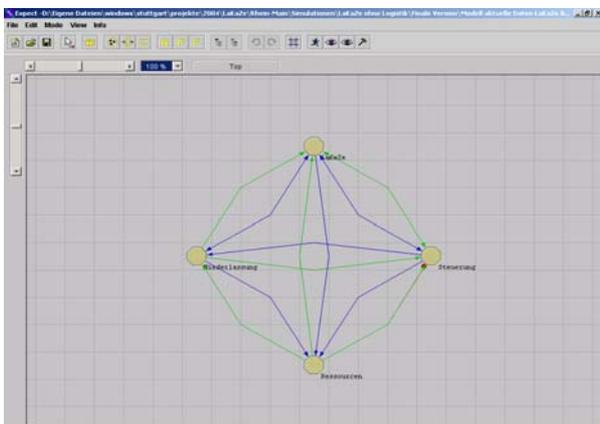
memory policy of type age or enabling. Memory policies define how the enabling time of a transition is treated when the transition becomes disabled for any reason other than itself firing. In the enabling case, a transition will "forget" that it has been enabled for a certain period; when it once again becomes enabled, a new firing time will be computed. In the case of the age policy, the transition "remembers" its enabling time, which shortens its remaining firing time when it once again becomes enabled. These memory policies are very important for modeling purposes.

Expect also supports multiple arcs, whose multiplicity may be specified as a constant or with a function.

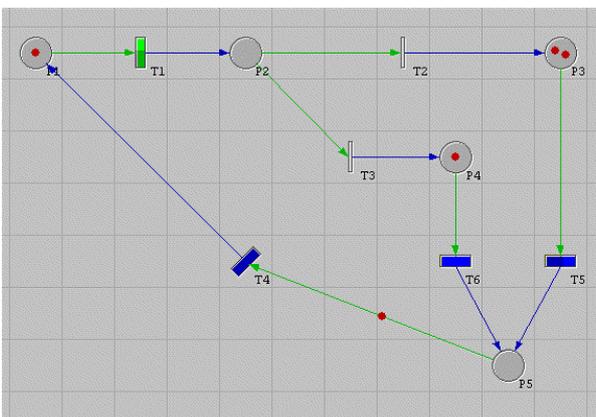
Various types of rewards are also available, including accumulated, non-accumulated and impulse rewards, which are associated with the firing of a transition. In addition to the net components themselves, independent parameters and functions may be defined, which can be referenced by the functions used to parameterize the net components. These facilitate the fast and safe modification of model parameters.

In the following, three screenshots of Expect demonstrate, how a system can be modeled in a hierarchical manner, how a detailed block looks like and how analysis results are finally presented to the user.

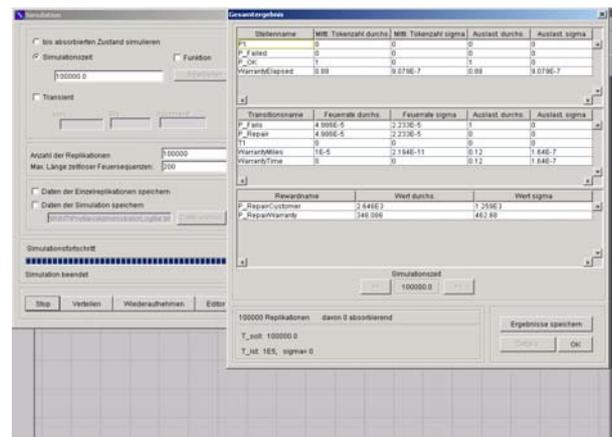
- Hierarchical top layer of a Petri net



- Detailed Petri net model within a hierarchical layer



- Representation of the analysis results



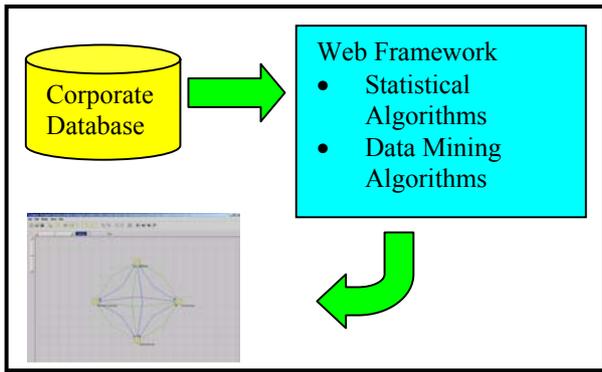
NEXT DEVELOPMENT STEPS

Expect is used very heavily at DaimlerChrysler Research in the form described here. In addition to the current functionality, various extensions are also planned, which will be briefly described.

One important application of Expect is the simulative prediction of the availability and reliability of current and future on-board vehicle systems. In addition, the tool is used to analyze quality measures and predict warranty costs. These applications include features for automatic report generation and integration with a corporate database, which contains up-to-the-minute reliability data. In order to access this database, an interface to Expect has been developed and is currently under test. Another very important topic while providing an interface between the analysis tool Expect and our corporate database is the pre-calculation of model parameters, based on our warranty data. Therefore, right now techniques are included into the tool Expect that allow the user to set up a model. Once this is done, the user can link every transition in the model with a component in our corporate database. As soon as this link is established, the system automatically extracts the field data from the corporate database that are associated with the link, performs a statistical analysis on these data and feeds the corresponding parameters into the model. Every time, the model is started and information in the corporate database has changed (e.g. new field data arrived at the corporate database), then the system automatically initiates a re-calculation of the parameters and starts a new simulation.

In this sense we are able to set up a quality-and cost tool suite, consisting of

- A Web-framework that contains statistical analysis algorithms for pre-processing the field data
- The analysis tool Expect that uses the pre-processed field data to parameterize and analyze the model



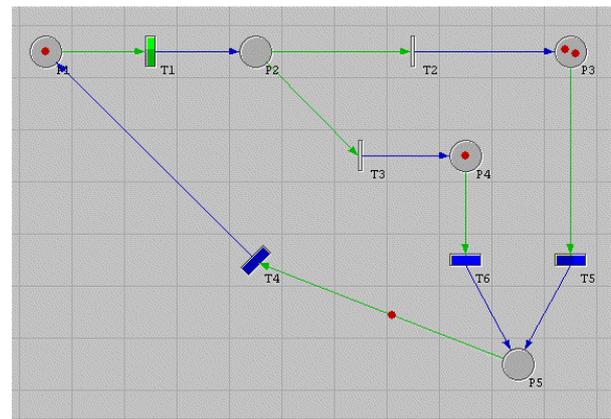
Many questions of interest can be answered by analyzing the state space of a net. These include qualitative properties such as deadlocks and livelocks in addition to the usual quantitative simulation results such as average reward values and marking probabilities. In particular, when all timed transitions in the net are exponentially distributed (or phase-type), then the state space can be converted into a continuous-time Markov chain, for which very efficient numerical transient and steady-state analysis methods are available. For these reasons, state space generation and analysis techniques will also be included in the tool.

A further issue of interest is to increase the performance of the discrete-event simulation itself. This is motivated by the high accuracy requirements for safety analyses, requiring a large number of replications to achieve statistically significant results, and the large degree of stiffness of many models, which require a long time to reach steady-state. In both cases, very expensive computations can result. Our approach to accelerating simulations by aggregation of the state space [HeHoLu98] will be studied in the Petri net context. In addition, an automatic parallelization of the net has been incorporated into the tool. Hereby, not the Petri net is split into pieces and each being analyzed on a different computer but we just parallelize the replications across a network of workstations. This means that we distribute the overall Petri net to different computers. Each computer starts the simulation with a different starting value. In this case we avoid the communication overhead involved in a

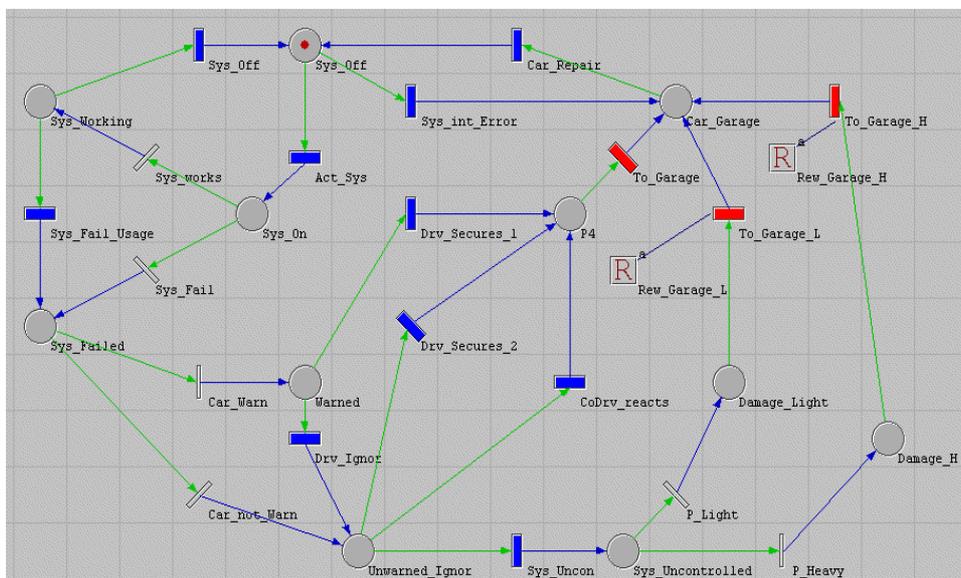
standard parallelization (distributing pieces of the Petri net to different computers). This type of parallelization proved to be very effective, stable and provided a linear speed-up. Therefore, this type of parallelization has been incorporated into the tool as a standard feature

MODELING EXAMPLES

Example 1: The following picture shows a screenshot taken during the visualization of the simulation of a small example net. The distribution functions of the transitions can be identified by the transition color. The expired enabling time of each transition is shown by the darker portion of the transition. Tokens are drawn in red. When a transition fires, tokens move symbolically along the input arcs from the input places to the transition and along the output arcs from the transition to the output places, according the various arc multiplicities. Viewing the simulation visualization in this manner significantly enhances the user's understanding of the net's dynamic behavior, and is very useful in presentations to non-specialists and of course for verifying (i.e debugging) the model.



Example 2: The next Expert screenshot shows a small Petri net, which is part of a model developed at DaimlerChrysler Research. The goal of this project was to study the reliability of a system over the entire life



cycle of a vehicle, including the interaction with the driver, in particular the manual deactivation of faulty systems. Rewards are used to compute the probabilities of light and serious damage occurring due to inappropriate behavior by the driver. The screenshot shows a Petri net with its initial state, in which the system is off. This is characterized by the token in the place *sys_off*. In this state it is possible to activate the system by the user or the system can fail unused. When the system is not used and it fails, it is detected by the system and it changes the state in defect. This behavior is modeled by the transition *Sys_intError*, which changes the state of the Petri net from *Sys_off* in *Car_Garage*. From this state, the system will be repaired and changes into the initial state of the net. If the system has not failed, the user can activate it and the token in place *Sys_off* walks to place *Sys_on*. By the activation of the system it is possible that the system is defect. In the case of no defect, the state of the system will change to working and the token walks to the place *Sys_working* by firing of the transition *Sys_works*. In this state the system can be deactivated by the user and the system changes in the state deactivated. This way from system deactivated to activated and back is the normal usage of the system, but while using the system, the system can fail and the token in place *Sys_Working* moves to the place *Sys_Failed*. This state of the system is the same state which will be reached if the system is down and will be activated. In the case of a system failure while the system is in use, either the driver can be warned by the vehicle or, in the case of a failure, the warning does not occur. When the user is warned, the token moves to place *Warned*. In this state, the driver can ignore the warning or he will secure the vehicle. Ignoring of the warning will lead to the same state as when the vehicle cannot warn the driver. This state is characterized by a token in place *unwarned_ignor*. At this point, the question arose, whether it is necessary to improve the availability of the warning or not. In the original model it was not obvious if it is or not. By analyzing the model, it was possible to prove that the warning must not be redesigned, because the most probable cause for damage was the ignorance of the driver. If the driver does not ignore the warning, he secures the vehicle (by braking, for example) and the system will be repaired at a garage. This case is modeled by the places *P4* and *Car_Garage*, and the transitions *Drv_Secures_1*, *To_Garage* and *Car_Repair*. In the case where the driver ignores the warning, or there is a warning failure, we have three different possibilities. The first way is the securing by the driver, if he is fast enough to react; the second is securing by the front-seat passenger if he can react fast enough. In both cases, the token will go to place *P4* and the system will be repaired in the garage. The last possibility is the worst one, in which the system has failed and the vehicle is in an uncontrolled state, in which there is no way to return to a secure state. In this situation, damage is unavoidable and the question is how heavy the damage will be. In

the model, the damage is divided in two classes, light and heavy. These are modeled in the Petri net with the transitions *p_Light* and *p_Heavy*. In both cases, the vehicle is repaired after the crash and the vehicle with the system returns to the initial state. A very interesting question for DaimlerChrysler was the amount of light and heavy damage. For measuring this, the impulse rewards *Rew_Garage_L* and *Rew_Garage_H* are used. The model is a simplified model, because it abstracts from system functions and different failure modes of the system. Nevertheless, it shows the usage of Petri nets for analyzing systems and the behavior of systems with interactions with humans. With the model shown in this example, it was possible to prove that a failure in the warning system was not the main danger; the main problem was the driver ignoring the warning, which lead to new concepts for increasing the safety of the overall system.

REFERENCES

- [ABE94] R.B. Abernethy, "The New Weibull Handbook", 536 Oyster Road, North Palm Beach, Florida, 1994.
- [BGdT98] G. Bolch, S. Greiner, H. deMeer, K. Trivedi, "Queueing Networks and Markov Chains – Modeling and Performance Evaluation with Computer Science Applications", Jogn Wiley&Sons, New York, 1998.
- [BuFa01] R.L. Burden, J.D. Faires, "Numerical Analysis", 7th Edition, Pacific Grove: Brooks/Coole, 2001.
- [HeGrHo92] S. Heller, S. Greiner, G. Horton, "PeNeTo: A Petri Net Simulator for Fast Safety and Quality Analysis and Cost Prediction", Proceedings ESM 2002.
- [HeHoLu01] S. Heller, G. Horton, M. Luber, "Accelerating Discrete-Event Simulation via State Space Reduction", ESM 2001, Prague, June 2001, Society for Computer Simulation and Modeling.
- [Lawl92] J. Lawless, "Statistical Models and Methods for Lifetime Data", John Wiley, New York, 1982.

BIOGRAPHY

Dr. Stefan Greiner studied Computer Science at the University of Erlangen. After two years at Duke University (Prof. Trivedi) he received his PhD degree in 2000 from the University of Erlangen. Since 1997 he is working for DaimlerChrysler Research in Stuttgart. His research area is the analysis and prediction of quality and cost. In this context, a tool environment for quality, cost and safety analysis is developed and contantly extended.

**HIGH-PERFORMANCE
COMPUTING
AND
SIMULATION**

SIMULATION STUDY OF A SIGNALLING PROTOCOL EFFICIENCY IN A COMPOSITE RADIO ENVIRONMENT

Charalabos Skianis
Kimon Kontovasilis
George Kormentzas

National Center for Scientific Research "DEMOKRITOS"
Institute for Informatics & Telecommunications
GR-15310 Ag. Paraskevi, POB 60228, Greece
E-mail: {kkont, skianis, gkorm}@iit.demokritos.gr

Giordana Lisa
Motorola Electronics S.P.A - GSG Italy
Via Cardinal Massaia 83, 10147 Torino, Italy
E-mail: giordana.lisa@motorola.com

KEYWORDS

Heterogeneous Systems, Signalling, Simulation.

ABSTRACT

Key aspects of a signalling protocol governing the interactions between terminals and network entities within a composite radio environment are presented in this paper. On the basis of these aspects, simple simulation models for capturing the dynamics of the message exchanges are discussed. The simulation study allows the derivation of both qualitative and quantitative results, that provide insight on key characteristics of the environment.

INTRODUCTION

Composite radio is directly relevant to the actively researched area of Fourth Generation (4G) wireless systems (4GVision&Technologies 2004). The concept refers to the joint employment of co-operating heterogeneous wireless technologies (Mobile 2.5G or 3G telecommunications systems, such as GPRS (3GPP TS 23.060 2004) and UMTS (3GPP TS 22.101 2004); Broadband Radio Access Networks, such as IEEE 802.11 (Groups 802.11 2004) or HIPERLAN (Varshney and Vetter 2000); and wireless broadcasting technologies, such as DVB-T (Digital Video Broadcasting 2002)) towards a versatile infrastructure that can support flexible wireless access to quality-aware information services.

Taking the same approach as the CREDO (Composite Radio and Enhanced service Delivery for the Olympics) (CREDO consortium 2001) project, this paper does not regard composite radio simply as a system where terminals switch to alternative access networks through a vertical handover merely upon loss of coverage, but rather as a system where its constituent components coordinate intelligently, towards exploiting the increased potential for optimisation that becomes possible when these constituents are jointly operated. Operation at this level of intelligence presupposes the existence of appropriate management functionality; such functionality is assumed to be available at both the composite network and the wireless terminals. In the

context just outlined, the following key features of the composite radio system may be identified:

- Diverse radio segments interconnected via a backbone.
- A Network Management System (NeMS) for the management of the composite radio network. It enables the joint optimisation of the alternate radio network segments, so as to deliver services efficiently, in terms of QoS.
- A Terminal Management System (TeMS) for the management of the wireless multimode terminals capable of operating in the composite radio environment and comprising functionality for interfacing to NeMS and for conducting decisions (NeMS-driven or independent) about the most appropriate radio technologies to be used for the efficient (in terms of QoS) reception of services through this terminal, under the each time applicable circumstances.
- The NeMS and TeMS exchange information towards beneficially combining the terminal's 'local view' (e.g., radio conditions in the area, services requested/received over the terminal, QoS levels associated with those services, etc) and the 'global view' of the composite network (e.g., traffic load over the various segments, QoS preservation via congestion avoidance, etc)
- Quality-aware applications and demanding service access.

It is clear that in order to fully develop a composite radio system comprised of the aforementioned key features it is imperative to adopt a signalling protocol governing the interactions between the terminal's TeMS and the network (through NeMS) towards the efficient selection (or reselection) of the most appropriate access network for the terminal to use. In this context, the paper presents a study on the efficiency of such a signalling protocol, implemented for the purposes of the IST project CREDO. The study contributes to the understanding of the protocol's dynamics and provides insight towards further improvements and modification. The concepts employed in the study are of a nature more general than the specific protocol and thus of value in other similar interaction mechanisms as well. Early work found in (Kontovasilis et al. 2003) on the model where simple analytic approximations are

employed to allow the computation of response times guided the presented work.

The rest of the paper is organised as follows: the second section outlines the technical requirements and the main structure of the TeMS-NeMS communication protocol developed in the context of the CREDO project. The third section argues on the necessity of an efficiency study and sets its context. Fourth section introduces the proposed model and exemplifies its use through the simulation of representative case studies. Finally, the last section concludes the paper.

SIGNALING PROTOCOL REQUIREMENTS

The main technical requirements for the communication protocol are outlined as follows:

- The interaction mechanism should allow terminals to
 - Ask from the network (through NeMS) the engagement of new/additional services.
 - Notify the network about service termination and candidate access segments within the terminal area.
 - Report to the network status parameters relating to the quality with which services are received over the terminal (e.g., status values at the radio-/ IP-, and application-level).
- The interaction mechanism should enable the network (through NeMS) to
 - Advise a terminal about the access segments that should be selected.
 - Ask from terminals to send status reports for assessing the conditions in specific radio segments.
 - Instruct terminals to switch to a different access network.
- The message protocol governing the interaction mechanism should not depend on the high-level IP means used for its implementation. In particular, the implementation should not assume that the order of messages sent from any communication end is preserved. Given the signalling nature of the protocol, low-overhead implementations requiring few packet exchanges may be preferable, but should not be imposed by the protocol's structure.
- The message protocol should be independent from the specific radio access technologies integrated into the composite network.

Following the above requirements, the messages of the signalling protocol between the terminals and the network can then be categorised as follows:

- Initial terminal registration/initialisation to the composite network.
- A core pair of messages (a Service Request/Reply pair) used from a terminal (the request) in order to report in a periodic manner the services running and the reachable network or to ask for new services, to report a change in the currently running services (e.g., a stopped service) or in the networks reachable to the terminal, and from NeMS when replying with the appropriate network to be selected.

- Messages for reporting quality status information from terminals to NeMS.

- A message from NeMS to a terminal for triggering a message exchange leading to reselection of the network segment used by the terminal.

The main advantages for the signalling protocol implemented within CREDO are its simplicity (easy to implement), its low overhead (appropriate to carry critical messages), and finally that it is specifically designed for responding to the challenges posed on a composite radio environment. An outline of the protocol in the context of the multimode terminal's architecture can be found in 0.

SIGNALING PROTOCOL EFFICIENCY STUDY

In light of the protocol's outline in the previous section, the bulk of the signalling traffic are Service Requests (sent from TeMS to NeMS), which trigger corresponding Service Replies (from NeMS back to the TeMS). If reply to a Service Request hasn't been received within time equal to half a Lifetime threshold, the request is considered to have been lost and is retransmitted by the TeMS.

Customarily each terminal should periodically send Service Request messages every T time units, even if there is no updated information to report. These periodic messages act also as "keep-alive" indicators, assuring NeMS that the terminal is "still there". Note that whenever $T \leq \text{Lifetime}/2$, there is no point in retransmitting timed-out requests, as a more recent message has been issued before deciding to repeat the old one. Retransmissions are of value when $T \gg \text{Lifetime}/2$.

Since other messages occur rarely compared to the ones just mentioned, they are not so important when studying the efficiency of the signalling protocol.

Exchange of messages between TeMS and NeMS occur in a network setting like that depicted in Figure 1. The configuration includes a GPRS network, a WLAN segment and a DVB-T segment. Note that the latter is unidirectional and must use one of WLAN or GPRS as return channels. Due to the presence of Mobile IP, each message from TeMS to NeMS must traverse the uplink radio segment, reach the system's Home Agent, and be forwarded to NeMS for processing. A corresponding sequence is involved in the backwards direction.

In light of this multi-step process, it is important to ensure that the exchange of messages occurs efficiently, i.e., that excessive delays in the delivery of messages are not very probable, considering even large-scale contexts.

Towards this goal, suitable modelling is employed to capture key properties of the signalling dynamics. Focus is on a 'steady-state' setting, where terminals regularly send Service Requests to NeMS and receive back the corresponding replies from it. The quantity of primary interest is the total roundtrip time from the issuance of the Service Request to the reception of the Service Reply. This should be kept sufficiently low to ensure

that the TeMS may take benefit of the advice from NeMS as soon as possible, in case this is required.

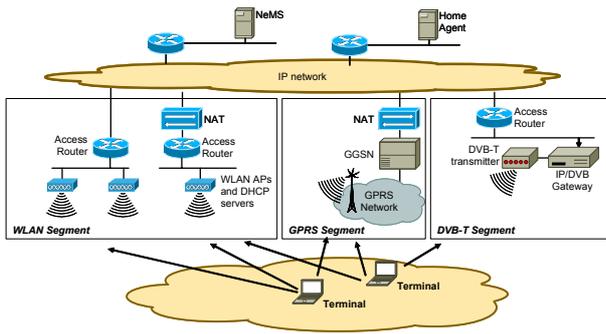


Figure 1: Composite radio network architecture

MODEL SPECIFICATION

This section introduces, in a detailed manner, the model developed for the study of the efficient exchange of signalling messages between network and terminal entities followed by the simulation study.

The Model

Figure 2 identifies the various components contributing to the roundtrip time between the emission of a TeMS-initiated Service Request message and the reception of the Service Reply message returned by NeMS in response. These components include network traversal (one-way delay) times and queueing (i.e., waiting plus service) times.

Random variables of potentially distinct probability distribution function (PDF) are used for the traversal times of different radio networks. Without further modelling complexity, it is also possible to assign different PDFs for uplink and downlink traversal of the same radio segment (this issue being void in the case of DVB-T, as it is unidirectional). The main simplifying assumption adopted by the model is that the times for different traversals along a given radio segment are assumed independent, for messages associated either to the same or to different terminals. This is an assumption typically employed in end-to-end networking studies and has been proven quite reasonable to adopt.

The traversal times over the IP backbone are associated with a PDF different from the previous ones. An independence assumption like the one just mentioned is assumed for this case too. It should be pointed out that although IP traversal does not always correspond to one-way delay between the same endpoints (e.g., there is traversal from the radio network to the Home Agent and from the Home Agent to the machine hosting NeMS) nevertheless, the model follows the assumption that all such times are distributed according to the same PDF. This assumption, supported by measurements, appears a reasonable one to make for high-speed backbones that employ optimised routing between their major nodes. These characteristics are quite representative of the setting being modelled.

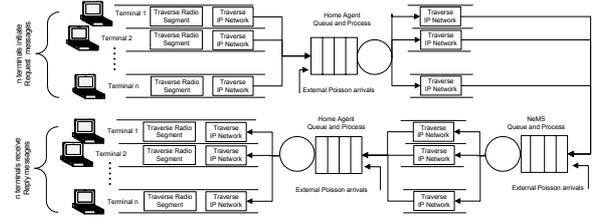


Figure 2: Flow of messages in Service Request/Reply chains

For the functional form of the traversal time PDFs, a rather conservative choice (i.e., one that is likely to yield estimates worse than the actual values) can be provided by the distribution corresponding to the sum of a constant time plus one term with a truncated exponential PDF. The second term signifies an additional occasional delay, due to congestion. Measured statistics for the mean, variance and percentile of exceeding a preset threshold could serve matching purposes. For a simpler choice, a plain exponential distribution may also be used. As already noted, different parameters apply to each of the three radio segments and to the IP backbone.

Besides network traversals, the time components in Figure 2 include two queueing times, at the Home Agent and at the NeMS, where messages pool for processing. Both queues are assumed to operate according to the FIFO policy. The service time at the Home Agent queue is the time required for processing/forwarding the packet carrying the message. The service time at the NeMS queue is the amount required for the internal bookkeeping (including possibly the execution of short-term optimisation algorithms) and for creating the Service Reply message. The traffic load to these queues primarily consists of the periodically generated Service Request messages (for both queues) and the corresponding Service Replies (only for the Home Agent queue). The model assumes that each active terminal in the composite network generates a service request every T time units. Consequently, the average rate of message arrivals due to these periodic messages is n/T at the NeMS queue and $2n/T$ at the Home Agent (because for each request there is a corresponding reply). It is conservatively assumed that the packets carrying the requests and replies do not get lost along the way and thus the queues are presented with the full 'nominal' load.

Besides the explicitly considered messages (and possibly retransmitted Service Requests, discussed shortly), external Poisson load is introduced by the model to each of the two queues. For the NeMS queue, the Poisson load models packets carrying other messages (e.g., corresponding to non regular Service Requests for invocation of a new service, or messages of a type not considered in this study). This load is typically low, compared to the main one. For the Home Agent queue, the Poisson load, besides implicit messages also represents actual data packets (i.e., carrying content from servers to terminals) and

(possibly) packets coming from other parts of the Internet, external to the composite network infrastructure.

The choice of Poisson process is a reasonable assumption when modelling a large aggregation of independent packet arrivals. In fact, when modelling a composite network with a large number of terminals, all traffic arriving at the two queues can be reasonably assumed of Poisson type. This further simplification is not employed in the simulation study. However, comparisons of mean value results between the simulation and an approximate analytic technique (c.f., (Kontovasilis et al. 2003)) support the validity of the simpler model in a context associated with many terminals.

In following the functionality of the signalling protocol, the model uses a Lifetime constant for determining ‘expired’ Service Requests (retransmitted only if the model’s input parameters satisfy $T > \text{Lifetime}/2$).

The model treats the events relevant to disjoint service request/replies (associated to either the same or different terminals) through independent “process threads” (meaning that events corresponding to drawing a time sample for e.g., traversing the radio segment, occur in parallel and independently between different threads) as depicted in Figure 2. These “threads” coordinate (become dependent) at the two common queues, which handle all packets arriving at them.

It is clear that the roundtrip delay involved in a request/reply pair is equal to the sum of all the random variables associated with the time components of Figure 2, distinguishing when applicable between uplink and downlink radio traversal (e.g., a required distinction for a terminal in DVB-T, as the return path is through another radio segment). However, when the parameters of the model suggest that retransmissions be modelled, the computation of the roundtrip is a bit more complicated when timeouts occur. Indeed, assume that a request timeouts k times; this means that $k+1$ requests were ultimately generated and only the last one received a reply before time equal to $\text{Lifetime}/2$ passed from its issuance. In that case, the roundtrip delay should be computed by subtracting the time when the first request was generated from the time when the $(k+1)$ -th reply was received at the terminal.

Given this model description the following list provides suitable Input/Output parameters for use when the model is implemented on a simulator:

— Input Parameters

- Number of terminals active (in each of the three radio segments; for each of those in DVB-T, the uplink segment must also be specified).
- Lifetime.
- Time interval T between the periodic issuance of Service Request messages from each terminal (not including those generated due to timeouts, should the notion apply).
- Delay PDFs for the IP traversal and the traversal of radio segment.

- Service time PDFs for the Home Agent and the NeMS queues.
- Poisson load at the NeMS and the Home Agent queues.
- Output parameters:
 - Statistics for the roundtrip time for each segment hosting active terminals.
 - Probability of timeout (per segment).
 - Average arrival rate at each of the two queues (non-redundant only if retransmissions are enabled).

A Typical Case Study.

Table 1 summarises typical input parameters for an application of the methodology just described. Some parameters (like the f factor and the mean service times) have been intentionally kept greater than the values one would normally expect them to have, in order to come up with conservative predictions.

Table 1: Typical input for the study of system’s reaction to terminals’ requests

Network Parameters			
Network Size		240 terminals	
Terminal distribution among radio segments			
GPRS	WLAN	DVB/GPRS	DVB/WLAN
110	110	15	5
Traversal Times (ms)			
GPRS	WLAN	DVB-T	IP
400	50	60	100
Home Agent Parameters			
average service time		25 ms (i.e., 40 messages/sec)	
Sq. coefficient of variation for service time		1	
<i>Arrival rate HA</i>		9.6 messages/sec	
Utilisation, ρ		0.24	
NeMS Parameters			
average service time		50 ms (i.e., 20 messages/sec)	
Sq. coefficient of variation for service time		1	
<i>Arrival rate NeMS</i>		3.47 messages/sec	
Utilisation, ρ		0.174	
Additional Parameters			
Lifetime		180 sec	
f -factor for additional load		30%	
$r_{external}$		100% of n/T	

Nevertheless, traversal times are typical, as is the value of the Lifetime. The parameter T is half the Lifetime. The composite radio hosts a large number of terminals, so that the Poisson assumption is realistic.

The model has been simulated on the OPNET simulation platform. In all cases simulation time was large enough to guarantee reaching steady state conditions.

Figure 3 displays the mean roundtrip delay for each radio segment as a function of the number of terminals, with values ranging from 200 up to 400 terminals. With magnification of the size of the system, appropriate dimensioning is presumably applied to the radio segments, so that the traversal times do not vary. In the graph, different traversal and queuing time components are indicated by different colour (tones of grey in black-and-white printouts). It can be observed

that most of the delay is due to traversing networks, while queueing is negligible, even for higher number of terminals, resulting in higher traffic at the queues. The most dominant part of the delay is due to IP traversal and (where applicable) to the traversal of the ‘slow’ GPRS segment.

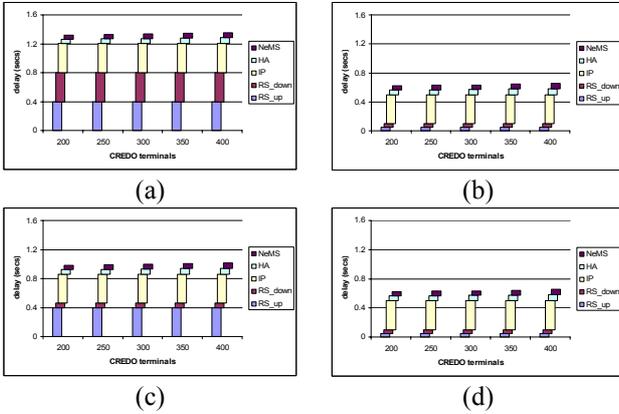


Figure 3: Mean roundtrip delay (and parts of it) for exchange of service request/reply messages vs number of terminals, for terminals in GPRS (a), WLAN (b), DVB/GPRS (c) and DVB/WLAN (d)

Additionally Figure 4 and Figure 5 display the roundtrip delay for each radio segment as a function of the Home Agent (rates from 25 up to 50 packets/sec) and the NeMS (rates from 10 up to 35 packets/sec) service times respectively. In each case a trend is exhibited in respective delays within the queues with the service times modified. However, the overall effect on the total round-trip time remains small in all cases due to the significantly larger delays imposed by the IP and the radio segment traversals, which are independent of the service times within the queues.

In all cases, in a qualitative sense the higher delays are found at the ‘slow’ GPRS segment, the faster segment is the WLAN, while DVB-T is affected by the choice of the uplink route with WLAN being the faster choice.

It is also notable that even in the most stringent case, the mean roundtrip delay remains below 2sec, i.e., about 50 times smaller than the timeout threshold. This illustrates that timeouts are highly improbable and would have remained so for considerably smaller values of T. Chebyshev’s inequality (see e.g., (Feller 1968)) gives a conservative upper bound for this probability, through $\Pr\{T > 50 \cdot \text{mean_delay}\} \leq \text{SCV_delay} / 50^2 = \text{SCV_delay} \cdot 4 \cdot 10^{-4}$, which is less than 1% even when the squared coefficient of variation for the delay is as high as 25.

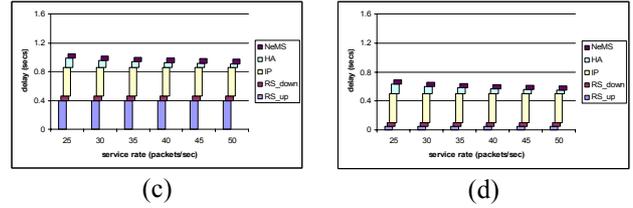
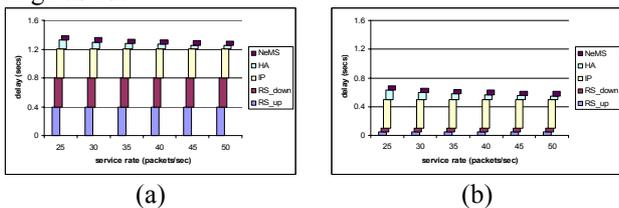


Figure 4: Mean roundtrip delay (and parts of it) for exchange of service request/reply messages vs Home Agent service rate, for terminals in GPRS (a), WLAN (b), DVB/GPRS (c) and DVB/WLAN (d)

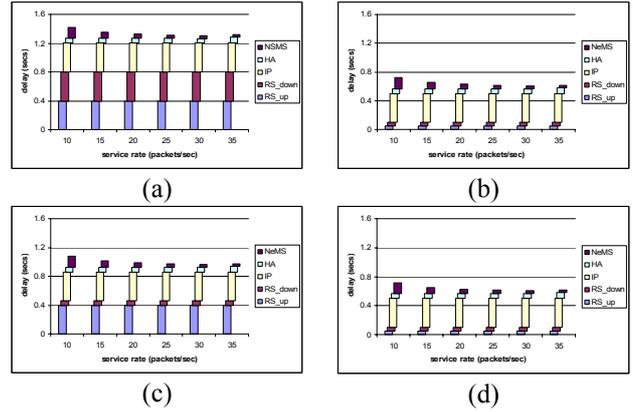


Figure 5: Mean roundtrip delay (and parts of it) for exchange of service request/reply messages vs NeMS service rate, for terminals in GPRS (a), WLAN (b), DVB/GPRS (c) and DVB/WLAN (d)

As an indicative additional test, Figure 6 displays the mean roundtrip delay for DVB/WLAN, this time with unreasonably high variability (SCV=10) in the service times (at both the Home Agent and NeMS). The impact of queueing in the total delay has increased, although it has remained much smaller than the IP traversal part (see subfigure (a), when T has the previous value). The right subfigure (b) maintains the high value in the SCV parameter, doubling though the value of T (effectively halving the load at the queues) with the end-result of bringing the queueing times back to the previous very small levels. This illustrates that the ability of the signalling protocol to negotiate the Lifetime parameter provides a versatile mechanism for controlling signalling congestion.

Note that in the case where the squared coefficient of variation is higher than one the Generalised Distribution (see e.g., (Kouvatsos 1994)) was adopted which is completely specified by just the mean and the variance.

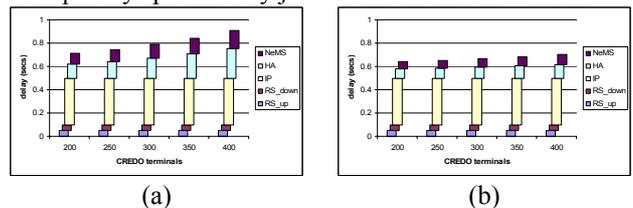


Figure 6: Mean roundtrip delay (and parts of it) for exchange of service request/reply messages vs number of terminals in DVB/WLAN and SCV=10, with T=1/90sec (a) and T=1/180sec (b)

CONCLUSIONS

This paper outlined key aspects of composite radio environments, focusing on a signalling protocol that governs the interactions between the network management system and intelligent multi-mode terminals. A framework for studying the impact of the signalling messages exchanged between the terminals and the Network and Service Management system, in a context of large scale was reported. Models for capturing the message exchanges were discussed, along with simulation studies for representative cases. In all cases the implemented protocol was shown to work efficiently. The model and the simulation allow the display of response times broken into separate time components for network traversal and queueing. Results from case studies showed that IP traversals contribute significantly to the overall time, while queueing times are less important. Future work would involve exhaustive comparison studies between simple analytic approximations and simulation additionally expanded to other types of exchanged signalling messages. Further degrees of complexity are envisaged towards a finer set of approximation and simulation tools for the study of the behaviour of the signalling protocol.

ACKNOWLEDGEMENT

This work has been performed in the framework of the project IST-2001-33093 CREDO, partly funded by the European Union under the IST programme.

The work outlined herein has benefited from contributions by all partners in the CREDO consortium, which consists of: Motorola Labs (MOTLABS)—France, National Center for Scientific Research (NCSR) “Demokritos”—Greece, Motorola Technology Center of Italy (MTCI)—Italy, Thales Broadcast & Multimedia (TBM)—France, National Technical University of Athens (NTUA)—Greece, Vodafone Hellas—Greece, IBM Hellas—Greece.

REFERENCES

- 3GPP TS 22.101, 2004. 3GPP TS 22.101: *UMTS Principles*, <http://www.3gpp.org>.
- 3GPP TS 23.060, 2004. 3GPP TS 23.060: *General Packet Radio Service (GPRS)*, <http://www.3gpp.org>.
- 4GVision&Technologies, 2004. Electronically available information in <http://www.mobileinfo.com/3G/4GVision&Technologies.htm>
- M. Catalina-Gallego and P. Stathopoulos, 2003. “Terminal Station Management System for Optimised Service Delivery in Composite Radio Environments”, *IST Mobile & Wireless Communications Summit 2003*, 15-18 June 2003, Aveiro, Portugal.
- CREDO consortium, 2001. *Technical Annex* (September).
- Digital Video Broadcasting (DVB), 2002. <http://www.dvb.org> (Jan).
- W. Feller, 1968. “An Introduction to Probability Theory and its Applications, Volume 1”, John Wiley and Sons.
- Groups 802.11, 2004. Electronically available information in <http://grouper.ieee.org/groups/802/11/>
- K. Kontovasilis, C. Skianis, G. Kormentzas, 2003. ‘Estimating Signalling Efficiency in a Composite Radio Environment’, Proc. Of 1st International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks, D. Kouvatsos (ed.), Ilkley pp. 65/1-65/12.
- D.D. Kouvatsos, 1994. “Entropy maximisation and queueing network models” in *Annals Oper. Res.* vol. 48, 63-126.
- U. Varshney and R. Vetter, 2000. “Emerging Mobile and Broadband Wireless Networks”, *Commun. of the ACM*, Vol. 43, No 6 (June).

A NEW FORM OF EFFICIENT TREE-BASED PRIORITY QUEUES FOR DISCRETE EVENT SIMULATION

Rick Siow Mong Goh* • Ian Li- Jin Thng⁺ • Wai Teng Tang • Marie Therese Quieta
Department of Electrical and Computer Engineering, National University of Singapore
3 Engineering Drive 3, CCN Laboratory, Singapore 117576
Email: <*engp1815@nus.edu.sg, ⁺elet1j@nus.edu.sg>

KEYWORDS

Priority queue, splay tree, skew heap, calendar queue.

ABSTRACT

A priority queue plays an important role in stochastic discrete event simulations for as much as 40% of a simulation execution time is consumed by the *pending event set* management. This article describes a new form of tree-based priority queues which employs the *demarcation* procedure to systematically split a single tree-based priority queue into many smaller trees, each divided by a logical time boundary. These new *Demarcate Construction* priority queues offer an average speedup of more than twice over the single tree-based counterparts and outperform the current expected $O(1)$ Calendar Queue in many scenarios. Their superior performance renders them suitable for many applications such as discrete event simulators.

INTRODUCTION

In stochastic discrete event simulation (DES), we often observe that the known kinds of efficient tree-based priority queues such as the Splay Tree (Sleator and Tarjan 1985) and Skew Heap (Sleator and Tarjan 1986) only have at best an *amortized* time bound of $O(\log(n))$ per operation, where by amortized time is meant the time per operation averaged over a worst-case sequence of operations (Tarjan 1985). Comparatively, multilist-based priority queues such as the Calendar Queue (CQ) (Brown 1988) and its variant Dynamic CQ (DCQ) (Oh and Ahn 1998) offer an “expected” $O(1)$ average time bound per operation, where by “expected” is meant that the CQs are not theoretically proven to be $O(1)$ but rather displays an $O(1)$ performance in numerous scenarios. However, the drawback of employing the CQs is that the worst-case time bound per operation can be as poor as $O(n)$ (Rönngren and Ayani 1997). That said, the CQ has been chosen as the *pending event set* (PES) structure in various simulators such as the popular Network Simulator v2 (Fall and Varadhan 2002).

In DES, the PES is defined as the set of all events generated during a DES and of which the events have not been simulated yet. The basic operations of the enqueue and dequeue of events define the PES as a priority queue of events with the minimum time-stamp having the highest priority and maximum time-stamp having the least priority. Comfort (Comfort 1984) has revealed that up to 40% of the computational effort in a

simulation may be devoted on the management of the PES alone, where the *enqueue* and *dequeue* operations account for as much as 98% of all operations on the PES. A DES frequently operates in a three-step cycle: *dequeue* – removal of an event with the highest priority from the PES; *execute* – processing this dequeued event; *enqueue* – insertion of new event/s resulting from the execution into the PES. The two basic operations, enqueue and dequeue, have run-time complexity closely dependent on the total number of events in the PES. Therefore, a PES structure should be efficient especially for large-scale simulations that involve large number of events during simulation jobs.

In most applications the metric of interest for a priority queue is often the time required to perform the most common operations. This metric is referred to as *access time*. In DES, the total run-time of the simulation job is by far more important than the individual times of the operations, except for real-time applications. Therefore, the amortized (or average) access time per operation is by far more important than the worst-case access time for each individual operation. Fine-grain simulations, such as but not limited to ATM network simulations, are time-consuming due to the huge number of events to process (Oh and Ahn 1998). The faster and the larger the networks, the higher the number of events would be in the PES and the longer run-times these network simulations would require, which may take days or weeks to yield results with an acceptable level of statistical error. For example, experiments conducted in Tcpsim (Dupuy et al. 1990) for a three-minute simulated time over Sun Ultra 1 took more than one day execution time on average (Oh and Ahn 1998). Therefore, to speed up simulation jobs, one approach is to develop high-performance priority queue structures for the PES.

In this article we develop the *Demarcate Construction* (*Demarco*) priority queue, a multilist-based structure which is made up of two building blocks. The name *Demarco* arises from the word “demarcate” which means to divide and separate clearly as if by boundaries. The primary structure is an array of buckets, where each bucket may contain a tree holding *near-future* events. The secondary structure is made up of a simple unsorted linked list to hold *far-future* events. *Demarcation* refers to the process of constructing the primary structure and transferring events from the secondary structure to the primary. In an amortized sense, this demarcation process ensures that a tree-based priority queue has

comparable performance or better, than one which does not undergo demarcation.

DEMARCATÉ CONSTRUCTION

The *Demarcate Construction (Demarco)* has four essential principles. First and foremost, the concept of demarcation is to have many trees each containing a small number of events. In contrast, a tree-based priority queue manages only a single tree containing all the nodes or events. Upon applying demarcation, an array of logical buckets is constructed. Each bucket spans equal time-interval and these buckets systematically enable the events to be demarcated and distributed in the buckets. Thus on the average, the tree in each bucket will have a smaller number of events leading to a much reduced height as compared to a single tree priority queue.

Secondly, *Demarco* defers the sorting of events until necessary. At the onset, all enqueued events are appended in the secondary tier (*SecT*) of *Demarco*. These events are not sorted according to their timestamps. During the first dequeue operation, the primary tier (*PriT*) is constructed and the events are inserted into the corresponding buckets in *PriT* where they are sorted according to the tree-based priority queue's native enqueue algorithm.

Thirdly, unlike other multilist-based priority queues (e.g. the CQs), *Demarco* does not rely on sampling heuristics to obtain structure parameters. The parameters used when constructing *PriT* are obtained from the events distribution in the *SecT*.

Lastly, the algorithm of *Demarco* proceeds in demarcation *cycles* where by a *cycle* is defined as the duration when: the events in *SecT* are transferred to the *PriT*, more events are enqueued in *PriT* and *SecT*, and all the events in the *PriT* are dequeued.

Basic Structure of Demarco

The main building blocks of the *Demarcate Construction (Demarco)* consist of:

1. Primary Tier (*PriT*) – an array of buckets where each bucket may contain a tree. Each tree-node contains an event holding a near-future (i.e. soon to be dequeued) timestamp. Within each bucket, the events are sorted according to the algorithm of the tree-based priority queue. The parameters used in creating the *PriT* are obtained from the events distribution in *SecT*.
2. Secondary Tier (*SecT*) – an unsorted singly linked list. Acting as an overflow list to contain far-future events, *SecT* buffers events that do not affect the *PriT*. This reduces the number of events in the *PriT* and thus, on the average, the number of events in each bucket decreases as simulation time progresses. Since the performance of tree-based priority queues depends on the height (or number of levels), reducing the number of events in *PriT* will

eventually lead to a reduction in the height of the tree in the buckets in *PriT*. This leads to a superior overall performance.

The Demarco Algorithm

Though the *Demarco* is a multilist-based structure alike the CQs, *Demarco* marks the first departure from the CQs' resize triggers and sampling heuristics to obtain structure parameters such as the number of buckets and the bucketwidth. Instead of the static methodologies used in the CQs, *Demarco* employs a dynamic approach of updating its structure parameters by making *PriT* structure parameters (i.e. bucketwidth and number of buckets) to be dependent on the events distribution in *SecT*. Since the *Demarco* proceeds in cycles, the structure parameters of *PriT* gets renewed according to the most current events and are not affected by the past events. This process removes the need to have costly resize operations found in the CQs. This becomes more vivid when the enqueue and dequeue operations are described.

The *Demarco* structure keeps a set of variables to function and they are defined as follow:

PriT_Start – Used for calculating the bucket-index of the event which is to be enqueued in *PriT*. It is set to *SecT_Min* during each Demarcation process, where by events are transferred from *SecT* to *PriT*.

PriT_Num – Number of events in *PriT*.

PriT_Bw – Bucketwidth of *PriT*.

PriT_Index – Bucket-index of the first non-empty bucket in *PriT*.

SecT_Cur – Minimum timestamp of an event that can be enqueued in *SecT*. This value will be set equal to *SecT_Max* at each transfer of events from *SecT* to *PriT*.

SecT_Min – Minimum timestamp in *SecT*.

SecT_Max – Maximum timestamp in *SecT*.

SecT_Num – Number of events in *SecT*.

The Demarco Algorithm – Dequeue Operation

At the onset, all enqueued events are placed in *SecT* in a FIFO manner without time-order thus leaving *PriT* being empty. On the first dequeue operation, *PriT* is constructed and thereafter, all the events are transferred from *SecT* to *PriT*. The bucketwidth of *PriT*, an important structure parameter, is dynamically assigned using equation (1).

$$PriT_Bw = \text{Bucketwidth} = \frac{SecT_Max - SecT_Min}{SecT_Num} \quad (1)$$

The number of buckets to be created in *PriT* is set to be *SecT_Num*, giving an average of one event per bucket on the assumption that the event distribution is a uniform distribution. Though in practical scenarios this may not be true, the *Demarco* will still perform well because the enqueue of events into *PriT* is $O(\log(n_B))$ per event whereby n_B is the number of events in a bucket. For most scenarios, $n_B \ll N$, where N is the total number of events in the *Demarco* structure.

After the construction of $PriT$, the events in $SecT$ are transferred to $PriT$. Transferring of an event into $PriT$ is alike enqueueing an event into $PriT$ which utilizes the tree-based priority queue's native enqueue algorithm. Thereafter, the highest priority event would be in the first bucket in $PriT$ (where $PriT_Index = 0$ and that $PriT_Start = SecT_Min$ have been initialized). On each dequeue, the highest priority event would be removed from the first bucket in $PriT$ by employing the tree-based priority queue's native dequeue algorithm. Subsequently, when the first bucket is empty, it is *invalidated* and the second bucket is then considered, where at the same time, parameter $PriT_Index$ is incremented by one. If the second bucket is empty, $PriT_Index$ is incremented again until a non-empty bucket is found and the current highest priority event is dequeued. After all the events in $PriT$ are dequeued, i.e. all the buckets are empty, the demarcation cycle repeats itself with $SecT$ treating the next dequeue to be alike the first dequeue as mentioned.

The Demarco Algorithm – Enqueue Operation

For each enqueue operation, *Demarco* checks if that event timestamp is greater than $SecT_Cur$. If so, the event is simply placed at the end of the linked list in $SecT$. If the event is not inserted in $SecT$, then the event is enqueue in $PriT$. On enqueueing in $PriT$, the bucket-index of the bucket where this event is to be inserted in $PriT$ is:

$$Bucket_index = \left\lfloor \frac{timestamp - PriT_Start}{PriT_Bw} \right\rfloor \quad (2)$$

and the event is enqueue according to the tree's native enqueue algorithm.

PERFORMANCE MEASUREMENT TECHNIQUES

The performance of priority queues are often measured by the average access time of the enqueue and dequeue operations under different load conditions. The parameters to be varied for each priority queue performance benchmark are: the access pattern, the priority increment distribution and the queue size. The access pattern models used are the Classic Hold and Up/Down. They emulate the steady-state and the transient phase of a typical simulation respectively. The various priority increment distributions tested are as shown in Figure 1 and the queue size ranges from 100 to 1 million. These benchmark scenarios had also been commonly applied in (Jones 1986; Rönngren et al. 1993; Rönngren and Ayani 1997; Oh and Ahn 1998). The experiments were carried out on an AMD Athlon MP 1.2GHz dual-processor server running the priority queues sequentially. Required memory was pre-allocated. All code was written in the C programming language. 10 runs of each experiment were done.

The Camel(x,y) distribution is used to model bursty traffic in computer and communication networks which represents a highly-skewed distribution. The parameters

used for Camel(x,y) result in two humps with x probability mass being concentrated in the two humps. The duration of the humps makes up y of an interval, where x and y are (0,1). Change(A,B,x) is a compound distribution that combines priority distribution A and B , with x priority increments being alternately drawn by A and B . In our experiments, Camel(0.999,0.001) and Change(Exp(1),Triangle,2000) are used.

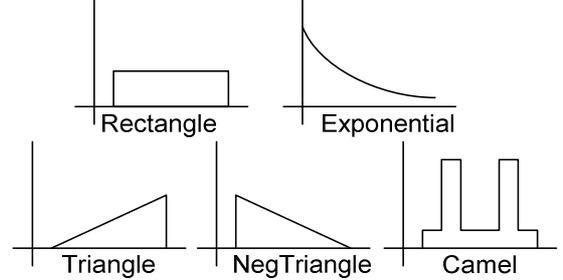


Figure 1: Priority Increment Distributions

EXPERIMENTAL RESULTS

The objectives of this section are firstly to present the performance of tree-based priority queues with and without *Demarco*. Secondly, we compare *Demarco* priority queues with the current fastest multilist-based queues – CQ and DCQ. Lastly, we would like to determine *Demarco* priority queues' generality and sensitivity in the six priority increment distributions using the Classic Hold and Up/Down models, as well as when the queue size increases from 100 to 1 million. Note that a logarithmic scale has been used for the queue-size axis which leads to logarithmic complexity for linear plots.

Steady-State Phase Experiments

Figures 2(a) to 2(f) show the results obtained under the Classic Hold experiment which is commonly employed to test the steady-state performance of the priority queues. Note that the obvious knee seen in the graphs is due to the declining cache performance and occurs when the queue size is about 10,000. This phenomenon is also observed in the graphs in (Rönngren and Ayani 1997) where the experiments were done on SUN and Intel architectures.

Figures 2 show vividly that the performance of *Demarco* structures, i.e. *Demarco-Skews* and *Demarco-Splays*, outperform the tree-based priority queues; Skew Heap and Splay Tree, where by *Demarco-Skews/Splays* is made up of a *Demarco* structure where each bucket in $PriT$ of *Demarco* may contain a Skew Heap/Splay Tree. At larger queue sizes, the performance speedup that *Demarco* offers is more than three times. Figures 2(a) to 2(d) show that the performance of the *Demarco* structures are comparable to the expected $O(1)$ complexity multilist-based priority queues, i.e. CQ and DCQ. Furthermore, Figures 2(e) and 2(f) demonstrate clearly that the *Demarco* structures outperform the CQs which have erratic performance for skewed distributions such as the Camel and Change.

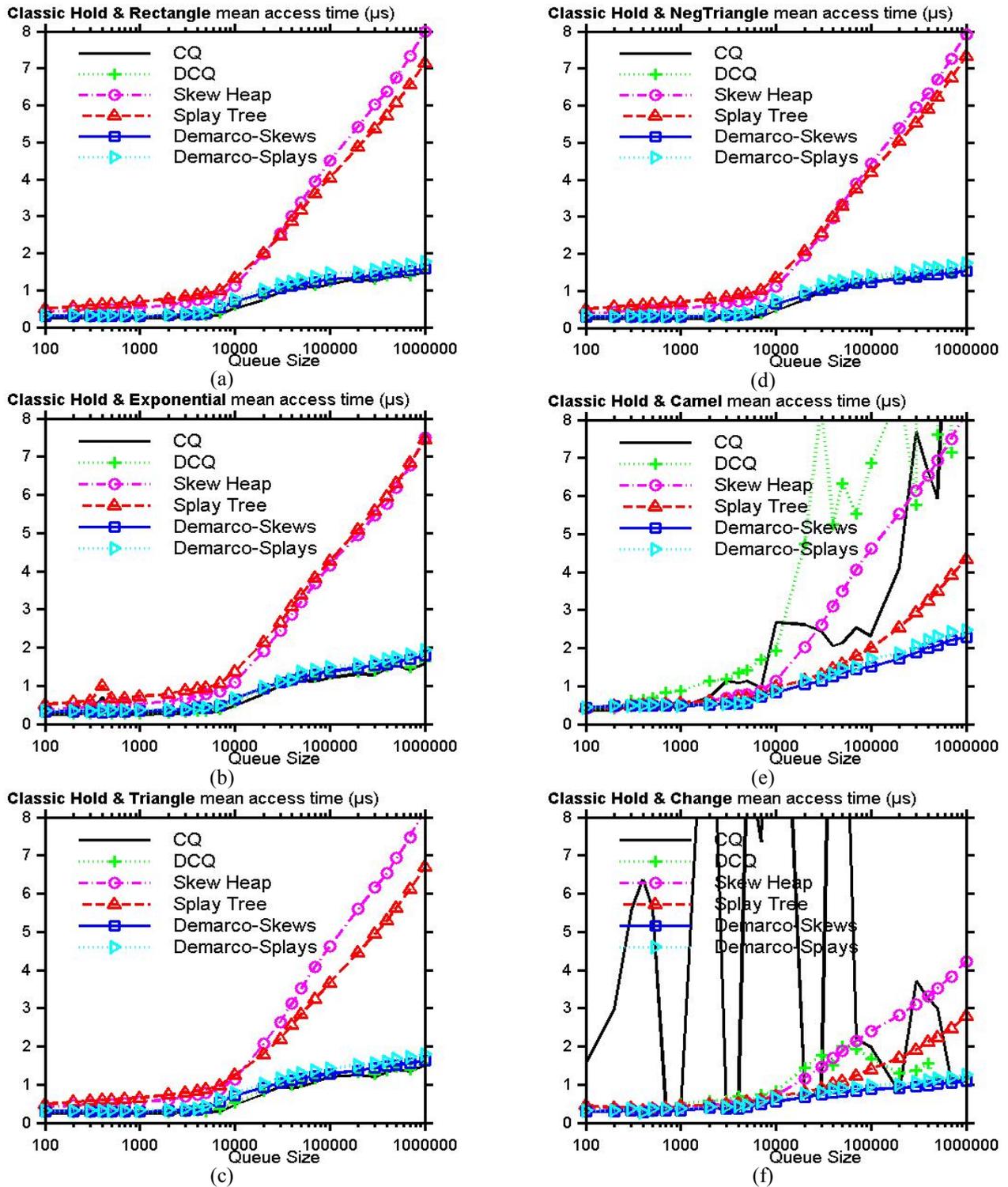


Figure 2: Performance Graphs for Classic Hold Model Experiments.

Transient Phase Experiments

The Up/Down model which tests the performance of priority queue structures during transient periods when the queue size fluctuates frequently, reveals the weaknesses of the CQs. Figures 3(a) to 3(d) show that the CQs experience several peaks and these suggest strongly that the resize operations found in the CQs can be costly since the CQs resize whenever the queue size fluctuates by factors of two. The form of triggers found

in the CQs are clearly inflexible because even though the CQs can be performing well with its existing operating parameters, but because of their static triggers, they still have to resize whenever the queue size fluctuates by factors of two. Figures 3(e) and 3(f) again demonstrate that the CQs are sensitive to skewed distributions. The *Demarco* structures outperform all the priority queues in all these experiments.

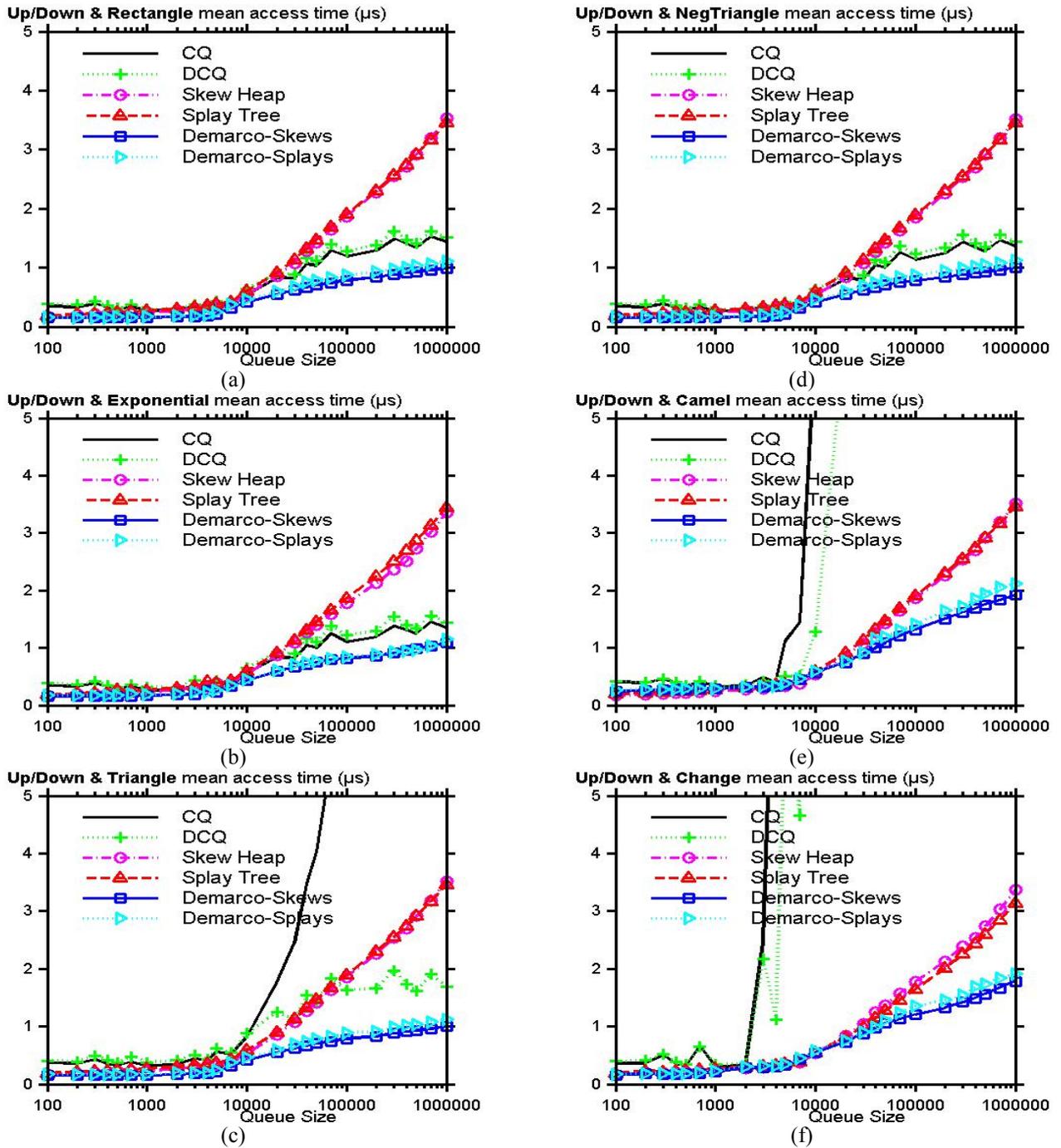


Figure 3: Performance Graphs for Up/Down Model Experiments.

Overall Performance Comparison

This section illustrates numerically the performance speedup of the *Demarco* structures over the normal single tree-based priority queues. In addition, we compare the relative performance of the *Demarco* structures and tree-based priority queues versus the multilist-based CQ and DCQ.

Table 1 shows that the speedup offered by the *Demarco* structure is more than two times, average over all queue sizes and distributions. Table 2 illustrates the relative performance of all the priority queues considered. The *Demarco-Skews* and *Demarco-Splays* outperform their

tree-based counterparts and are generally more stable than the CQs.

Table 1: Speedup Offered by the *Demarco* Structure Normalized Over a Single Tree-Based Priority Queue – Comparison by Priority Increment Distribution

Distribution	<i>Demarco-Skews</i>	<i>Demarco-Splays</i>
Rectangle	2.99	2.61
Exponential	2.62	2.66
Triangle	3.01	2.46
NegTriangle	2.99	2.66
Camel	2.05	1.31
Change	1.91	1.39
Average	2.60	2.18

Table 2: Relative Average Performance for All Distributions
(Normalized Respect to Fastest Access Time where 1.00 is the Fastest)

Model	Queue Size	Demarco-Skews	Demarco-Splays	Skew Heap	Splay Tree	CQ	DCQ
Classic Hold	100	1.07	1.14	1.31	1.61	1.59	1.00
	1000	1.15	1.26	1.76	2.15	1.00	1.41
	10000	1.00	1.10	1.55	1.70	5.32	1.23
	100000	1.00	1.12	3.27	2.59	1.20	1.81
	1000000	1.00	1.11	4.44	3.61	1.90	NA*
	Average	1.04	1.15	2.47	2.33	2.20	NA*
Up/Down	100	1.00	1.09	1.01	1.12	2.17	2.38
	1000	1.00	1.06	1.25	1.42	1.56	1.75
	10000	1.00	1.08	1.16	1.27	19.55	15.96
	100000	1.00	1.10	1.93	1.95	NA*	NA*
	1000000	1.00	1.10	2.67	2.63	NA*	NA*
	Average	1.00	1.09	1.60	1.68	NA*	NA*
Total Average		1.02	1.12	2.04	2.01	NA*	NA*

* NA is meant that some of the access times are too high in at least one or more distributions. Thus the results are not considered in this comparison.

Generality and Sensitivity of Demarco Structures

Figures 4(a) and 4(b) show the generality and insensitivity of *Demarco-Skews* under the various distributions and queue sizes (*Demarco-Splays* has similar graphs and is thus not included). Though the performance of *Demarco-Skews* may differ by as much as twice for different distributions, the complexity is still considered near $O(1)$. Furthermore, the graphs show that it is stable for all the distributions unlike the CQs which is near $O(n)$ for skewed distributions. This superior performance is made possible because of the four essential principles mentioned.

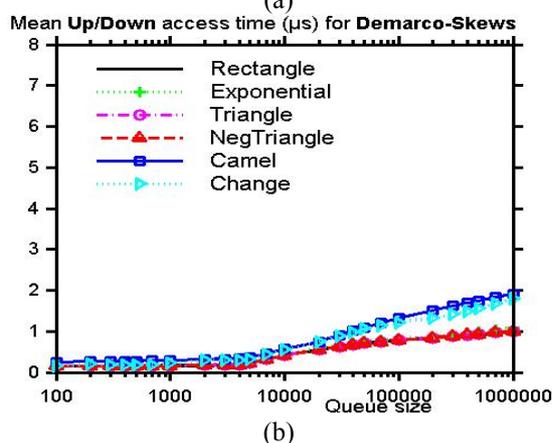
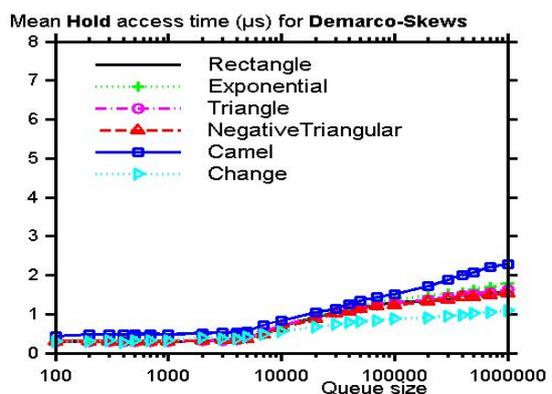


Figure 4: Performance Graphs for *Demarco-Skews*

CONCLUSION

Demarcat Construction is a new form of tree-based priority queues which employs the *demarcation* process. These new priority queues offer an average speedup of more than twice over the single tree-based counterparts and outperform the current expected $O(1)$ Calendar Queues in many scenarios. Its generality in small to large queue sizes (100 to 1 million events), insensitivity to priority increment distributions and low overhead costs, make it a superior priority queue for many applications such as the pending event set structure in discrete event simulators.

REFERENCES

- Brown, R. 1988. "Calendar Queues: A Fast $O(1)$ Priority Queue Implementation for the Simulation Event Set Problem." *Commun. ACM* 24, 12 (Dec.), 825-829.
- Comfort, J. C. 1984. "The Simulation of a Master-Slave Event Set Processor." *Simulation* 42, 3 (March), 117-124.
- Dupuy, A., Schwartz, J., Yemini, Y. and Bacon, D. 1990. "NEST: A Network Simulation and Prototyping Testbed." *Commun. ACM* 33, 10 (Oct.), 63-74.
- Fall, K. and Varadhan, K. 2002. The ns Manual. UCB/LBNL/VINT Network simulator v2. <http://www.isi.edu/nsnam/ns/>.
- Jones, D. W. 1986. "An Empirical Comparison of Priority-Queue and Event-Set Implementations." *Commun. ACM* 29, 4 (April), 300-311.
- Oh, S., and Ahn, J. 1998. "Dynamic Calendar Queue." In *Proceedings of the 32nd Annual Simulation Symposium*, 20-25.
- Rönnngren, R. and Ayani, R. 1997. "A Comparative Study of Parallel and Sequential Priority Queue Algorithms." *ACM Trans. Model. Comput. Simul.* 7, 2 (April), 157-209.
- Schwetman, H. 1996. CSIM18 User's Guide. Austin, TX: Mesquite Software, Inc.
- Rönnngren, R., Riboe, J., and Ayani, R. 1993. "Lazy Queue: New Approach to Implementing the Pending Event Set." *Int. J. Computer Simulation* 3, 303-332.
- Sleator, D. D. and Tarjan, R. E. 1985. "Self-Adjusting Binary Search Trees." *Journal of the ACM* 32, 3 (July), 652-686.
- Sleator, D. D. and Tarjan, R. E. 1986. "Self-Adjusting Heaps." *SIAM Journal of Computing* 15, 1 (Feb.), 52-69.
- Tarjan, R.E. 1985. "Amortized Computational Complexity." *SIAM Journal on Algebraic and Discrete Meth.* 6, 2 (April), 306-318.

RECURRENT NEURAL NETWORK WITH BACKPROPAGATION THROUGH TIME ALGORITHM FOR ARABIC RECOGNITION

Saliza Ismail¹ and Abdul Manan bin Ahmad²

Department of Software Engineering,
Faculty of Computer Science and Information System,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
Tel : 607-5532201² Fax : 607-5565044²
chukiwa@hotmail.com¹, manan@fksm.utm.my²

Abstract

The study on speech recognition and understanding has been done for many years. In this paper, we propose a new type of recurrent neural network architecture for speech recognition, in which each output unit is connected to itself and is also fully connected to other output units and all hidden units[1]. Besides that, we also proposed the new architecture and the learning algorithm of recurrent neural network such as Backpropagation Through Time (BPTT), which well-suited. We also re-train for the output before we analyze the result. The purpose of this training is to produce the best result. The aim of the study was to observe the difference of Arabic's alphabet like "alif" until "ya". The purpose of this research is to upgrade the people's knowledge and understanding on Arabic's alphabet or word by using Recurrent Neural Network (RNN) and Backpropagation Through Time (BPTT) learning algorithm. 6 speakers (a mixture of male and female) are trained in quiet environment.

Neural network is well-known as a technique that has the ability to classified nonlinear problem. Today, lots of researches have been done in applying Neural Network towards the solution of speech recognition[2] such as Arabic. The Arabic language offers a number of challenges for speech recognition[3]. Even through positive results have been obtained from the continuous study, research on minimizing the error rate is still gaining lots attention. This research utilizes Recurrent Neural Network, one of Neural Network technique to observe the difference of alphabet "alif" until "ya".

Keywords

Recurrent Neural Network, Backpropagation Through Time, Arabic Alphabet, Speech Recognition, Real-Time Recurrent Learning.

Introduction

Speech is human's most efficient communication modality. Beyond efficiency, human are comfort and familiar with speech. Other modalities require more concentration, restrict movement and cause body strain due to unnatural positions. Research work on Arabic speech recognition, although lagging that other language, is becoming more intensive than before and several researches have been published in the last few years [4].

The conventional neural networks of Multi-Layer Perceptron (MLP) type have been increasingly in use for speech recognition and also for other speech processing applications. Those networks work very well as an effective classifier for vowel sounds with stationary spectra, while their phoneme discriminating power deteriorates considerably for consonants which are characterized by variations of their short-term spectra. This may be attributable to a fact that feedforward multi-layer neural network are inherently unable to deal with time varying information like time-varying spectra of speech sounds. One way to cope with this problem to incorporate feedback structure in the networks to provide them with an ability to memorize incoming time-varying information. Incorporating feedback structure in feedforward networks results in so-called Recurrent Neural Networks (RNNs) which have feedback connections between units of different layers or connections of self-loop type [5].

Speech recognition is the process of converting an acoustic signal, captured by microphone or a telephone, to a set of words. The recognized words can be the final results, as for applications such as commands and control, data entry and document preparation. They can also serve as the input to further linguistic processing in order to achieve speech understanding, a subject covered in section [6]. As we know, speech recognition performs their task similar with human brain. Start from phoneme, syllable, word and then

sentence which is an input for speech recognition system [7]. Many researches that have been prove to prove to decrease the error and also any disruption while doing the recognition.

Now, students not interested in lessons regarding Arabic language such as Jawi writing even the lessons have been teaching at primary school. The purpose of the lessons is to teach the students how to pronoun and write the alphabet. Therefore, students can read Holy-Quran properly. But the students only can understand that pronoun and writing while they in Standard 6. So after that, they will forget all the lessons [8].

Architecture

RNN have feedback connections and address the temporal relationship of inputs by maintaining internal states that have memory. RNN are networks with one or more feedback connection. A feedback connection is used to pass output of a neuron in a certain layer to the previous layer(s) [9]. The different between MLP and RNN is RNN have feedforward connection for all neurons (fully connection). Therefore, the connections allow the network show the dynamic behavior. RNN seems to be more natural for speech recognition than MLP because it allows variability in input length [10].

The motivation for applying recurrent neural network nets to this domain is to take advantage of their ability to process short-term spectral features but yet respond to long-term temporal events. Previous research has confirmed that speaker recognition performance improves as the duration of utterance is increased [11]. In addition, it has been shown that in identification problems RNNs may confer a better performance and learn in a shorter time than conventional feedforward networks [12].

Recently a simple recurrent neural network, which has feedback connections of self-loop type around hidden layer units, has been proposed as an attractive tool for recognizing speech sounds including voiced plosive sounds [1]. This network has three layers such as input layer, hidden layer and output layer. Each of the output layer units has feedback connection with itself, i.e., a self-loop as shown in Fig. 1.

The output of each input layer at time $t-1$ is fed, through connections between the input and hidden layers, to all the hidden layer units at time

t and in the same manner the output of each hidden layer unit at time $t-1$ is supplied, through connections between the hidden and output layers, to all the output layer units at time t . the output at time $t-1$ of each output layer unit is feedback to itself at time t . in training the proposed recurrent neural network, weight at t of all the connections between the input and hidden layers as well as connections between the hidden and output layers are affected by all the input vectors to the input vectors to the input layer before time t .

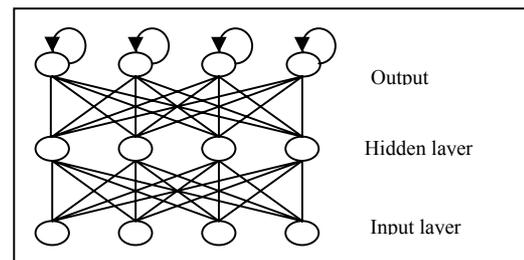


Fig. 1: RNN Architecture

Actually this architecture has been used in visual pattern recognition [13,14] but we use this architecture for speech recognition especially for Arabic speech recognition by using Backpropagation Through Time (BPTT) as learning algorithm. This architecture also have been proved that this architecture better than MLP in phoneme recognition accuracies [1] by using Backpropagation algorithm. In this paper, we want to prove that the architecture also can be used in Arabic's alphabet with Backpropagation Through Time (BPTT) learning algorithm.

The Backpropagation Through Time (BPTT) algorithm is based on converting the network from a feedback system to purely feedforward system by folding the network over time. Thus, if the network is to process a signal that is time steps long, then copies of the network are created and the feedback connections are modified so that they are feedforward connections from one network to the subsequent network. The network can then be trained if it is one large feedforward network with the modified weights being treated as shared weight [15]. Real-Time Recurrent Learning (RTRL) algorithm is based on recursively updating the derivatives of the output and error. These updates are computed using a sequence of calculations for iteration. The weights are updated either after iteration or after the final iteration of the epoch.

The major disadvantage of this algorithm is that it requires an extensive amount of computation at iteration [16]. Additionally, this algorithm is very slow because the RTRL has many weights to compute and therefore, the training process will be more slowly [9].

Speech Recognition System

Generally, speech recognition process contains three steps to process the speech which is acoustic processing, feature extraction and recognition, as shown in Figure 2. First, we digitize the speech that we want to recognize. In this paper, we digitize the Arabic's alphabet from the speakers and also digital filtering that emphasizing important frequency component in signal. Then we analyze the start-end point depends the signal of the speeches. GoldWave software is used to filter and conversion the analog to digital. From that, we can analyze the start-end point that contains the important information of speeches.

The second steps is feature extraction that digital signal in time domain will fed to LPC spectrum analysis for extract the signal or we called it as frame normalizing. Linear Predictive Coding (LPC) is used to extract the LPC coefficients from the speech tokens [17,18]. The LPC coefficients are the converted to cepstral coefficients. The cepstral coefficients are normalized in between +1 and -1. the cepstral coefficients are served as input to the neural networks.

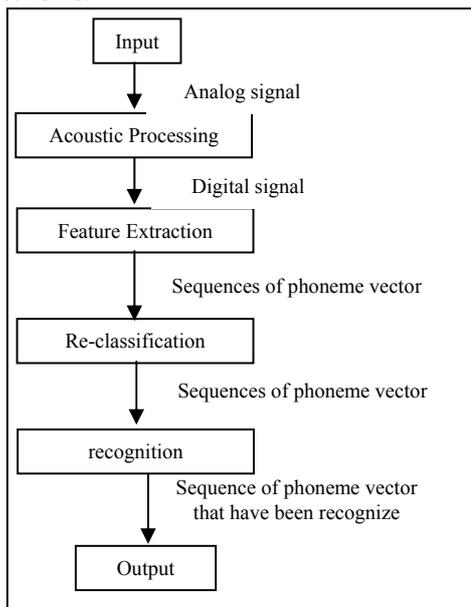


Figure 2: Process of Speech Recognition

Finally, we classify and recognize the speech with learning algorithm Backpropagation Trough Time in Recurrent Neural Network. Actually, before the recognize process, the output will be training once again for produce the best output. This process suitable for minimum input such as Arabic's alphabet that has 30 letters.

In spoken language, a phoneme is a basic, theoretical unit of sound that can change the meaning of a word. A phoneme may well represent categorically several phonetically similar or phonologically related sounds (the relationship may not be so phonetically obvious, which is one of the problems with this conceptual scheme). Depending on the language and the alphabet used, a phoneme may be written consistently with one letter; however, there are many exceptions to this rule (especially in English) [19].

The Arabic alphabet has 30 letters and it is written from right to left. Letters change shape depending on which other letters are before or after them, much like American or Continental handwriting. Phonemes are best described as linguistic units. They are the sounds that group together to form our words, although quite how a phoneme converts into sound depends on many factors including the surrounding phonemes, speaker accent and age. A phoneme is the smallest contrastive unit in the sound system of a language [20].

The Arabic's alphabet in this research contains 20 letters such as "alif", "ba", "ta", "tha", "jim", "ha", "kha", "dal", "zal", "ra", "zai", "sin", "syin", "sad", "dhad", "to", "za", "ain", "ghain" and "fa".

Experiments

The Table 1 shows the results of 20 alphabets from recognition experiments. The testing of this system has been pronounced by 6 Malay speakers (3 men and 3 women). Every speaker must repeat the Arabic's alphabet about 10 times sequentially for each alphabet. So, total of the pronoun for this experiments that includes 6 speakers x 20 alphabets x 10 times for every alphabet (6x20x10), are 400 speeches. From the table, the "ta" alphabet get 98% that higher and the lowest rate is "kha" alphabet, and also all the alphabet in the Table 1.

Table 1: Expected Result Arabic's Phoneme Recognition for Four Speakers using RNN and BPTT learning algorithm

Arabic's alphabet	Rate (%)
"alif"	85
"ba"	96
"ta"	98
"tha"	75
"jim"	87
"ha"	60
"kha"	60
"dal"	91
"zal"	84
"ra"	88
"zai"	85
"sin"	91
"syin"	90
"sad"	73
"dhad"	75
"to"	83
"za"	87
"ain"	78
"ghain"	72
"fa"	88

Conclusion

Currently development of speech recognition is widely used in industrial software market. The main contribution of proposed phoneme recognition system is encouraged to recognize the Arabic's alphabet properly. Besides, its can help the beginner to start their lessons about how to pronouns the word of Holy-Quran. Furthermore, we presented the new architecture and the learning algorithm that Backpropagation Through Time, are well-suited and better than Elman or Jordan architecture. However, the process become more effective after re-train the first output.

Findings from results of the expected experiments can be summarized as follows:

1. The low rate recognition of error to the alphabets that pronounced with bilabial (both lips) are contributed to the naturally pronunciation of those alphabet that are much clear such as "ba" and "ta" alphabets.
2. The lowest rate of recognition to the alphabets that pronounced with pharynx of those alphabets like "ha" and "kha" are contributed of pronunciation that needs 'makhrj' exactly (right pronunciation in Arabic).

Hopefully, this system will help us to recognize and differentiate the Arabic's. We also hope we'll continue the process until "ya" besides minimize the time.

References

- [1]Koizumi T., Mori M., Taniguchi S. and Maruya M., (1996). "Recurrent Neural Networks for Phoneme Recognition." Department of Information Science, Fukui University, Fukui, Japan, Spoken Language, ICSLP 96, Proceedings, Fourth International Conference, Vol. 1, 3-6 October, Page(s): 326 -329.
- [2]Turban E., (1992). "Expert Systems and Applied Artificial Intelligence." Republic of Singapore: MacMillan Publishing Company, 623-640.
- [3]Mayfield T. L., Black A. and Lenzo K., (2003). "Arabic in my Hand: Small-footprint Synthesis of Egyptian Arabic." Euro Speech 2003, Geneva, Switzerland.
- [4]Jihene El Malik, (1998). "Kohonen Clustering Networks For Use In Arabic Word Recognition System." Sciences Faculty of Monastir, Route de Kairouan, 14-16 December.
- [5]Medser L. R. and Jain L. C., (2001). "Recurrent Neural Network: Design and Applications." London, New York: CRC Press LLC.
- [6]Lippman R.P., (1989). "Review of Neural Network for Speech Recognition." Neural Computation 1.1-38.
- [7]Joe Tebelskis, (1995). "Speech Recognition using Neural Network." Carnegie Mellon University: Thesis PhD.
- [8]Siddiq Fadzil, (2001). "Martabat umat Islam Melayu menurut Hamka." Utusan Melayu, 30 April.
- [9]Ruxin Chen and Jamieson L. H., (1996). "Experiments on the Implementation of Recurrent Neural Networks for Speech Phone Recognition." Proceedings of the Thirtieth Annual Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, November, pp. 7790782.
- [10]Lee S. J., Kim K. C., Yoon H. and Cho J. W., (1991). "Application of Fully Neural Networks for Speech Recognition." Korea Advanced Institute of Science and Technology, Korea, Page(s): 77-80.
- [11]He J. and Liu L., (1999). "Speaker Verification Performance and The Length of Test Sentence." Proceedings ICASSP 1999 vol.1, pp. 305-308.
- [12]Gingras F. and Bengio Y., (1998). "Handling Asynchronous or Missing Data with

Recurrent Networks." International Journal of Computational Intelligence and Organizations, Vol. 1, no. 3, pp. 154-163.

[13]Lee S. W. and Song H. H., (1997). "A New Recurrent Neural-Network Architecture for Visual Pattern Recognition." IEEE Transactions On Neural Networks, Vol. 8, No. 2, March.

[14]Rabi G.and Lu S., (1997). "Visual Speech Recognition by Recurrent Neural Networks." Electrical and Computer Engineering, IEEE 1997 Canadian Conference on, Vol. 1, 25-28 May, Page(s): 55 -58.

[15]Werbos P., (1990). "Backpropagation Through Time: What It Does and How To Do It." Proceedings of the IEEE, 78, 1550.

[16]Sato M., (1990). "A Real Time Running Algorithm for Recurrent Neural Networks." Biological Cybernetics, 62, 237.

[17]Ting H. N., Jasmy Yunus and Sheikh Hussain Salleh, (2002). "Speaker-Independent Phonation Recognition For Malay Plosives Using Neural Networks." Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Malaysia.

[18]Rabiner L. R. and Juang B. H., (1993). "Fundamentals of Speech Recognition." 1st. Ed. New Jersey, United States of America: Prentice Hall. 58-148.

[19]El-Iman Y. A., (1989). "An Unrestricted Vocabulary Arabic Speech Synthesis System." IEEE Transactions On Acoustics, Speech and Signal Processing, 37(12):1829-1845.

[20]Greenberg S., Carvey H., Hitchcock L. and Chang S., (2001). "Beyond the Phoneme: a Juncture-Accent Model of Spoken Language." Proceedings of the Human Language Technology Conference (HLT - 2002), San Diego, California, March 24-27.

AN ARCHITECTURE FOR DISTRIBUTED SIMULATION OF WIRELESS NETWORKS

Ratan K. Guha and Oleg Kachirski

School of Computer Science

University of Central Florida

Orlando, FL 32816, USA

E-mail: {guha, oleg}@cs.ucf.edu

ABSTRACT

In this paper we describe a simulation framework designed for simulating wireless network technologies. We provide a detailed architecture of the framework and illustrate all the necessary steps to develop simulations for any particular wireless networking application. We demonstrate the use of this framework by simulating network intrusion detection system for wireless ad hoc networks. We also extend the framework architecture to multiprocessor environments and describe the optimizations to further improve the performance of the simulation system for large-scale simulations of wireless networks.

INTRODUCTION

Wireless networks today are used in almost every organization. Researchers enthusiastically work on developing new communication protocols, algorithms and methods. Many papers in the network research community propose new wireless algorithms, offer better performance, quality of service and other benefits. Technology of choice should be carefully evaluated before being adopted. Primary method of evaluating wireless networks is to create a computer simulation of a particular topology, communication mechanism or an algorithm. Our primary objective was to create a distributed simulation framework capable of running on a cluster computer, targeting wireless networks. The key requirements were flexibility, extendibility and scalability, as our final goal was to utilize a simulation system capable of being extended to very large simulations. Several well-known general purpose network simulators were considered (Heybey 1988; Grosej 1995; Mah 1998; Keshav 1997; VINT Project 1997; Chang 1999) to create our simulation model of a large-scale wireless network. After reviewing documentation and running sample simulations, we have come to the conclusion that none of the assessed simulators would meet our requirements due to several reasons, such as inflexible modeling of wireless nodes to include desired functionality (i.e., mobility models for ad hoc networks), and relying on C libraries from a particular architecture; thus making it non-portable between different computer architectures

WINDS Framework (Wireless Network Distributed Simulation Framework) was developed by our

Networks Research Group at the University of Central Florida as a research tool to provide time-saving flexible simulation test-bed targeting a wide range of wireless network architectures. The framework is cross-platform, Java-based, GUI-driven and can be used as a generic wireless network simulator for a variety of purposes, such as routing in ad hoc networks, mobility models of totally mobile wireless networks, etc.

The rest of the paper is organized as follows. First, we present the WINDS architecture design in the next section. Framework modules are described in detail in this section, as well as a generic simulation process. We also list requirements and limitations of our simulation framework. In section 3, a practical example of WINDS framework application is given as a test-bed for our Agent-based Intrusion Detection System for Ad Hoc Wireless Networks (Kachirski and Guha 2002; Kachirski and Guha 2003; Guha et al. 2002). Network clustering algorithm and agent allocation algorithm are presented, along with result discussion. Section 4 discusses on-going work on extending the framework to multiprocessor environments and distributing simulation objects efficiently to minimize inter-processor communications. Finally, a summary and conclusions are presented, along with plans for future development and parallelization of WINDS framework.

WINDS DESIGN PHILOSOPHY

There was a direct need for us to develop a network simulator that specifically targets wireless networks of diverse configurations. Our goals were:

- Clean, easy-to-understand and modify design
- Object-oriented approach, general portability
- Use of a popular programming language
- Easy-to-use, GUI driven framework
- Adequate performance and scalability

Following these guidelines, we have developed WINDS framework – a flexible, portable, wireless-network oriented framework that can be used to simulate variety of applications of wireless and hybrid networks. The objectives of this project were to reduce redundant software design efforts in the area of wireless network simulation, establish a framework general enough to be used for simulations of many wireless network-related technologies, and provide for common base for the

experimentation of various wireless infrastructures. The object-oriented nature of the software and the use of a popular programming language for implementation allow researchers to easily modify, reuse and share whole systems or system components. Graphical environment allows the system to be used for demonstrational or educational purposes.

WINDS Architecture

The architecture of WINDS (simulator and interface) is based on a building-block approach. Researcher implements an algorithm or a prototype from modules that receive inputs in the form of events, process them, and then generate outputs (events, log entries, GUI updates). The entire wireless network is built from objects – wireless nodes, event generator, and communication channels. Connections between wireless nodes are maintained by the routing object. Any type of wireless network is supported, such as ad hoc and infrastructure wireless networks. Object-oriented approach is central to the generality and flexibility of the system and allows users of our framework to reuse, share and catalog simulation components by modifying or replacing appropriate classes.

Many existing wireless network simulators (Boukerche et al. 2001; Kelly et al. 1998; Kelly et al. 2000; Liu et al. 1996) aim at supporting every aspect of wireless communications, such as, for example, every layer of many communication protocols used. This adds tremendous overhead to the simulation system, often resulting in scalability problems and slow execution times. The WINDS framework avoids these problems, by implementing only the key functionality of a wireless network – supporting simulation synchronization, wireless communications and node mobility. These

functions are implemented on an abstract level, allowing the user to include specific wireless communication protocols, wireless routing algorithms and other required simulation parameters, as necessary. Another goal of our framework development was to integrate user interaction with simulation execution as closely as possible. The intuitive graphical user interface reflects any changes in the simulation execution as they happen, in real-time, allowing the user to adjust the simulation parameters at run-time. Another advantage of our framework is the ability to use almost any data file as a source of wireless communications. Data file pre-processor converts binary packet data into the format accepted by WINDS simulation system, converting network addresses into plain addressing scheme used with our simulation framework. As an example, intrusion data from the Lincoln Laboratories IDS tests was used to test our wireless IDS system.

WINDS architecture consists of four key modules, each comprised of a number of components, as described below:

- GUI (graphical user interface)
- Simulator Core Module
 - Simulation engine, Simulation objects
- Network Traffic Module
 - Packet pre-processor, Packet generator
- Data Logging Module
 - Data parser, CSV file generator

Figure 1 shows the WINDS architecture. Some of the modules carry optional functionality and can be included into the simulation as necessary. The functionality of each module is described below.

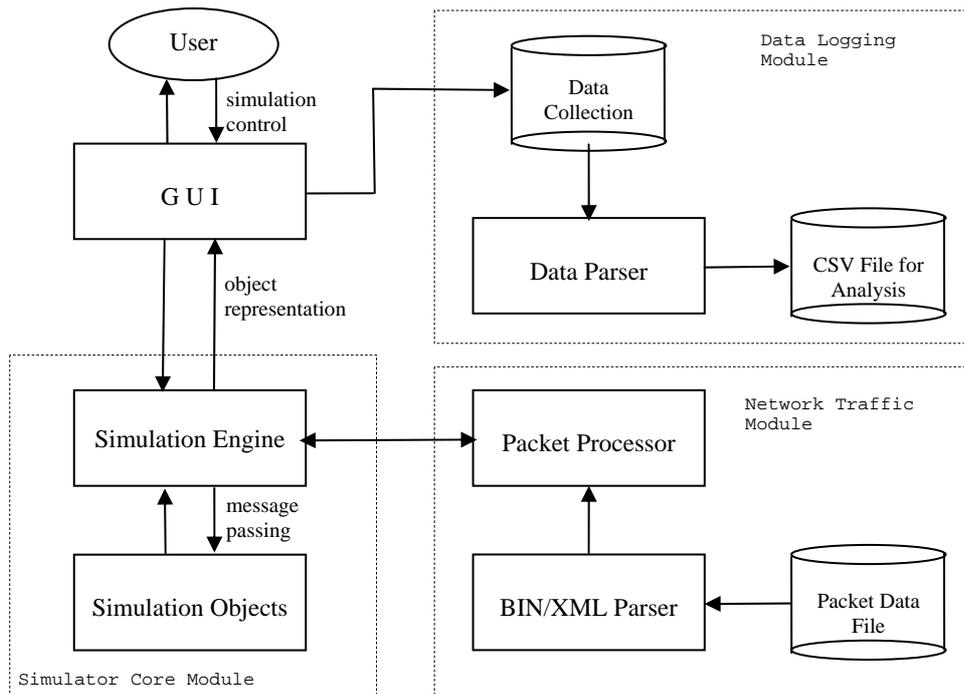


Figure 1: WINDS Architecture

Graphical User Interface Module

The graphical user interface (GUI) module shows the simulation execution in real-time. The GUI class is tied up to the simulation engine clock, and displays the required information every clock cycle. GUI class shows the simulation area with wireless node objects moving and communicating. Certain simulation parameters can be adjusted via GUI module. GUI module also provides controls for simulation execution flow. GUI module also records simulation statistics in a log window.

Simulator Core Module

Simulation Engine: the heart of the WINDS framework. One of the design requirements is that WINDS is a time-stepped simulation system. The simulation engine runs an internal clock, the speed cycle of which can be controlled at run-time. Programmatically implemented as a high-priority thread, the simulation engine runs in a loop continuously, driving execution of all other components of the simulation. Any events happening at a time must be processed by simulation objects at once, within a single simulation cycle. The simulation engine is common to all simulation models, and cannot be modified by the user. Simulation engine instantiates all simulation user objects from global definitions (such as wireless nodes, routing algorithm used, mobile base stations and stationary routers) at runtime.

Simulation Objects: execute independently and are time-synchronized via the simulation engine clock. Our framework has a few pre-defined simulation objects. One object type is a wireless node object. Wireless nodes are members of any wireless network simulation, and can be either stationary or mobile. User can modify the motion pattern by implementing a certain mobility algorithm (or even read waypoints from a data file). This allows simulations to be flexible and account for many possible node mobility patterns. Each wireless node object also includes two methods used for inter-object communication – Send and Receive methods. Send method is invoked when a wireless node is transmitting packets, and has a source, destination, protocol, port and payload as its arguments. Send method determines all the neighbors of the current wireless node, and broadcasts the packet to its neighbors by invoking Receive method on each neighbor node. Receive method first checks the packet destination, then depending on the routing algorithm used, forwards the packet to the destination or simply drops the packet. This allows the user to simulate both ad hoc and infrastructure wireless networks.

Routing protocol is another simulation object. The routing protocol included with WINDS framework is a simple table-driven protocol, which is implemented as a separate routing class. To modify the routing protocol, user must include all necessary parameters in a wireless node object, then add their own routing protocol class. Since the WINDS framework operates in exact same way as a real-life wireless network, all existing wireless

routing protocols are supported. For the simulation of totally-mobile or wireless infrastructure networks, routing protocol is modified to route all messages via base stations.

Network Traffic Module

Network traffic for WINDS framework is generated by the Network Packet Processor object. The implementation of the object is common to all the simulations and includes reading a pre-processed data file in XML format, and forwarding each packet to the appropriate wireless node (source). Pre-processing is performed on a binary data file obtained from network packet capturing software (such as TCPDUMP). WINDS uses a flat addressing scheme to reduce communication overhead, and therefore all network addresses are converted to compatible notation by the BIN/XML Parser module before running the simulation. Packet processor object generates packets at times specified by scaled timestamps of each packet processed. This speeds up simulation execution, limiting the simulation speed only by the hardware specifications and the maximum packet broadcast rate. Simulation can automatically be stopped when the end of the data file is reached.

Data Logging Module

Data logger class saves the simulation results for future analysis. During simulation execution, results are stored in memory and displayed in a human-readable form via GUI data display window for performance reasons (frequent disk I/O operations reflects negatively on simulation performance). The representation of results can be tailored to particulate simulation requirements and is defined in the GUI class.

Data parser: At the end of the simulation run, these results are first pre-processed by a data parser to format data suitable for import into the mathematical analysis software.

CSV file generator: The pre-processed output is saved as a CSV file (a widely-used comma-separated data file format) by the CSV file generator.

Simulation Process

The simulation procedure is as follows. First, a simulation diagram is devised by the user, which includes the list of all objects taking part in the simulation and interaction mechanisms between the objects. Packet and data output formats, as well as simulation stop conditions should also be specified at this point. Next, the implementation is written for all of the above objects, following the guidelines included with the simulation framework. Data and XML parsers should also be re-written to reflect the new data structures. The objects are then placed in the simulator core directory and compiled. The GUI is then started, which then instantiates all simulation objects and displays them on-screen (for those objects that have visual behavior defined). Objects then can be manipulated via graphical interface during simulation runtime. User can get a snapshot of the simulation

environment by pausing the simulator and stepping through simulation execution. If the network traffic module is selected, network packets are input from the packet data file, then converted to the appropriate format by the parser and fed into the simulation by the packet generator at specified times. Either actual time stamps may be used to pro-rate packet delay to simulation time scale, or a statistical distribution may be used, as specified in the packet generator class. Packets are then passed to appropriate objects via messages and processed. All collected statistics are displayed in a log window, which then can be saved into a simulation log file. The results of the simulation can be pre-processed for further analysis or graph plotting by the Data Parser module. The supplied data parser converts log file into a CSV file, which then can be imported into a variety of statistical programs.

WINDS FOR IDS SYSTEM DESIGN

Our first use of WINDS architecture was to develop a test-bed for a wireless intrusion detection system (IDS) (Lippmann et al. 1998; Haines et al. 2001) simulation prototype. Therefore, IDS is used as a tool to demonstrate an implementation of WINDS architecture. This section covers the architecture of our IDS system, step-by-step design process, and implementation of several IDS-related algorithms.

Wireless node object embodies mobile agent functionality for packet-monitoring and decision-making agents of an IDS system, as well as mobility pattern and clustering algorithm functionality. Send and Receive methods of a wireless node object are utilized to support voting scheme required by the cluster membership decision algorithm. These methods are also used by the network traffic generator module to propagate packets across wireless network. Packet-monitoring functionality is built-in the wireless node object for this simulation. Packet-monitoring agent processes the incoming network traffic, running it through the CASE-based intrusion detection mechanism, which bases its decisions on a library of XML rule sets, covering various communication protocols. Database lookup is performed for each network packet, classifying the packet as part of normal or intrusion traffic.

Our IDS system (Kachirski and Guha 2003; Guha et al. 2002) takes into account the specifics of wireless networks to provide a lightweight, low-overhead intrusion detection mechanism for wireless networks based on mobile security agent concept. Essentially, an agent is a small intelligent active object that travels across network to be executed on a certain host, then returns with results to the originator. All the decisions, including network traversing, are left to agents. Agents are dynamically updateable, lightweight, have task-specific functionality and can be viewed as components of a flexible and dynamically configurable IDS.

We have utilized mobile agents at several intrusion-monitoring levels and processed their response in cluster heads – special nodes that are dynamically elected within a cluster using a real-time distributed algorithm. One advantage of our approach is the efficient distribution of mobile agents with specific IDS tasks according to their functionality across a wireless ad hoc network. Another advantage is restriction of computation-intensive analysis of overall network security state to a few key nodes. These nodes are dynamically elected, and overall network security is not entirely dependent on any particular node, as in the case of a monolithic system. We have also proposed a load-balancing solution that efficiently distributes traffic monitoring and intrusion detection tasks among the wireless nodes, improving the accuracy of intrusion detection system without sacrificing the overall performance of a wireless network and functionality of each node participating in the network. At the network-monitoring level, we have developed a case-based approach to network intrusion detection, and incorporated case-based reasoning engine for detecting intrusions at the packet level in our modular IDS system. The IDS system implementation discussed here is targeted at ad hoc wireless networks.

WINDS FOR CLUSTER COMPUTER

The single-processor version of the simulation system suffices for small-scale simulations of wireless networks. However, we have had significant reduction in performance when simulations of 200 or more wireless nodes were in progress. Scalability is an important factor of every simulation system, as computer networks grow in size and become more complex in functionality. A distributed simulation is the answer to scalability problems. The idea of a distributed simulation is as follows. First, the entire scope of objects in the simulation is partitioned into several parts. For instance, we can divide 100 simulation objects equally between 5 processors. During simulation execution, each of the processors performs computations relevant to objects assigned to it. Communication between objects is handled by the communication broker – if two objects are handled by the same processor, communication happens in exact same way as in the uniprocessor system; if two (or more) objects are assigned to different processors, inter-processor communication takes place. Our distributed WINDS architecture is presented in figure 2. The distributed WINDS architecture is currently implemented on a cluster computer with 64 processors. Each processor has dedicated memory space, and can access data concurrently from a replicated disk subsystem. Inter-processor communication is achieved via high-speed Ethernet switch.

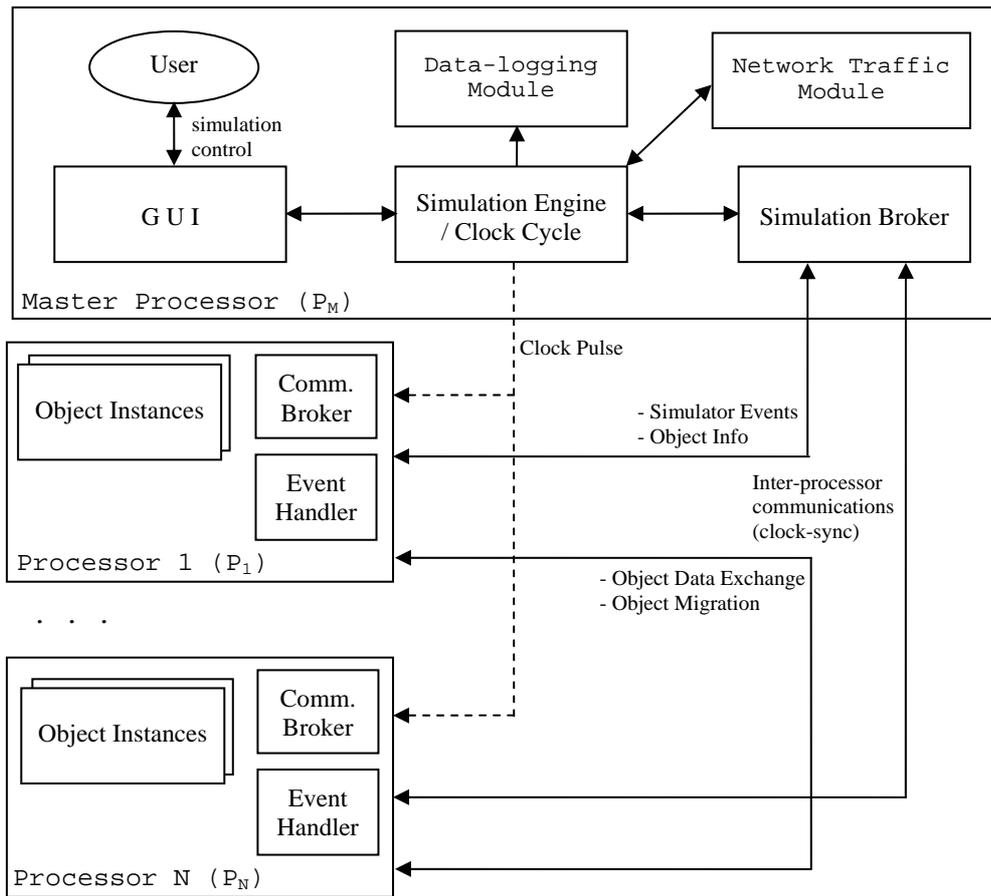


Figure 2: WINDS Architecture for Distributed Simulations in Multiprocessor Environments

Simulation Process

After object definitions have been devised and placed in a shared object directory, user starts the simulation framework on a master processor node. This in turn remotely starts simulation clients on each of the processors. Initially, user adds new objects to the simulation via graphical user interface. The objects are associated with processors in a round-robin manner. Instructions are sent to a respective processor from the master processor node to create an instance of an object and load it in memory. Object template is then read from disk by that processor, and a new object instance is created in its memory space. From now on, this object is handled by a local simulation module on that processor. Once all objects have been created, simulation run starts. Simulation engine sends a clock pulse out to every processor, and all communication between objects is clock-synchronized. Simulation broker located on the master processor keeps track of locating a specific object and serves as a routing module for the cluster communications. Other modules (like data logging module) behave in the same way as described for the single processor version of WINDS. When a packet needs to be sent from one object to another in the course of the simulation, communication broker on the node containing source object first determines if both objects reside on that node. If this is the case, then communication is performed locally by invoking the Receive method on destination object (same as for the

uniprocessor case). If the destination object resides on a different processor, first a proper destination network address of that processor is determined by consulting simulation broker on a master processor node. Then, a network communication is initiated between the local processor, and the destination processor, handled by communication brokers of both processors. When a message is received on the destination processor, it is parsed for parameters, and Receive method of the destination object is called. Apart from exchanging messages between objects, all the processors also communicate with the master processor once every few clock cycles to ensure consistent state of the simulation and to report on the state of each object taking part in a simulation (i.e., to update the GUI information for each object). Commands are also sent from the master processor to each processor to control simulation execution.

Future Work - Simulation Optimizations

One of the main reasons against distributed implementation of many network simulators is inefficient inter-processor communication. In the case of a network of computers taking part in a simulation, network delay can be a significant obstacle to the goal of improving performance and scalability through the distributed simulation. Other traffic exists on the network, affecting simulator communications. Unless network nodes are dedicated for the simulation purpose,

node stability is an issue that can disrupt simulation entirely in the event of a single node crashing during the simulation run. Only specific computation-intensive algorithms can benefit from distributing the simulation among multiple processors. We have considered these and other issues when developing distributed WINDS framework. The optimal choice of computer hardware was computing cluster, where all processors communicate via high-speed switched network connections but own independent memory space and an instance of an operating system. This allows us to simulate very large wireless networks of diverse configurations.

Still, concerns exist for certain scenarios where inter-processor communication delay is of an issue. This can happen when one object repetitively communicated with objects located on a different processor, or in the event of a lot of broadcast communications taking place. Therefore, we have considered a number of optimizations that target the problems associated with the distributed simulation system. In one such optimization, during a certain period of time, all communication patterns are recorded, and allocation of objects is then optimized. For example, if an object A communicated with object B much more frequently than other objects, and these two objects are located on different processors, then one object is serialized and migrates via the network to the processor managing another object. In many cases, especially when object functionality is sparse, the size of the object is small, justifying such a migration. In another optimization, all communications between objects are concatenated together and sent as a single network packet between a pair of processors once every few clock cycles, and then locally time-synchronized. As we haven't completed implementing the entire range of planned optimizations, little can be said here regarding the actual performance figures.

SUMMARY

In this paper we have described the WINDS architecture for wireless network simulations. It has been developed to aid our research on the agent-based ad hoc network intrusion detection system, and later used as a research tool that incorporates a flexible test-bed targeting simulations of a wide variety of wireless networks. The framework is cross-platform, easy to learn, use and modify to adjust particular requirements. WINDS is considered a generic wireless network simulator for a variety of wireless communication applications, such as wireless networks, ad hoc networks, sensor networks, etc. We have demonstrated the use of WINDS for several specific simulations, such as intrusion detection system simulation in wireless ad hoc networks. We have also extended WINDS implementation to the multiprocessor environments (such as a cluster computer), improving scalability of large wireless simulations. The WINDS project is ongoing and will be available soon in its final implementation for use by researchers worldwide.

ACKNOWLEDGEMENTS

This work was supported by the US Army Research Office, grant number DAAD19-01-1-0502. The views and conclusions herein are those of the authors and do not represent the official policies of the funding agency.

REFERENCES

- A. Boukerche, S. K. Das, A. Fabbri, "SWiMNet: a scalable parallel simulation test-bed for wireless and mobile networks", *ACM Wireless Networks*, 2001, v. 7, Issue 5, pp. 467-486.
- X. Chang, "Network simulations with OPNET", *Proceedings of Winter Simulation Conference*, 1999, pp. 307-314.
- B. Groselj, "CPSim: a tool for creating scalable discrete event simulations", *Proceedings of Winter Simulation Conference*, 1995, pp. 579-583.
- R. Guha, O. Kachirski, "Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks", *Proceedings of the IEEE Workshop on Knowledge Media Networking, KMN'02*, pp. 153-160.
- R. Guha, O. Kachirski, D. G. Schwartz, S. Stoecklin, E. Yilmaz, "Case-Based Agents for Packet-Level Intrusion Detection in Ad Hoc Networks", *Seventeenth International Symposium On Computer and Information Sciences*, Orlando, FL, October 28-30, 2002
- J. Haines, L. Rossey, R. Lippmann, R. Cunningham, "Extending the DARPA Off-Line Intrusion Detection Evaluations", *Proceedings of DARPA Information Survivability Conference & Exposition II, Volume: 1*, 2001, pp. 35-45.
- A. Heybey, "MIT Network Simulator", MIT Laboratory for Computer Science, 1988.
- O. Kachirski, R. Guha, "Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks", *Proceedings of 36th HICSS Conference*, 2003, pp. 57-64.
- O. Kelly et. al., "Scalable parallel simulations of wireless networks with WiPPET: modeling of radio propagation, mobility and protocols", *Mobile Networks and Applications*, 2000, v. 5, Issue 3, pp. 199-208.
- O. Kelly et. al., "Parallel simulations of wireless networks with TED: radio propagation, mobility and protocols", *ACM SIGMETRICS Performance Evaluation Review*, 1998, v. 25, Issue 4, pp. 30-39.
- S. Keshav, "REAL 5.0", Cornell University, 1997, <http://www.cs.cornell.edu/skeshav/real/overview.html>
- R. Lippmann et. al., "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation", *Proceedings of DARPA Information Survivability Conference & Exposition II, Volume: 2*, 1999, pp. 12-26.
- W. W. Liu et. al., "Parallel simulation environment for mobile wireless networks", *Proceedings of the 28th conference on Winter simulation*, 1996, pp. 605-612.
- B. Mah, "INSANE Users Manual", UC Berkeley, 1998, <http://www.employees.org/~bmah/Software/Insane>
- "NS-2 Simulator", VINT Project, 1997, <http://www.isi.edu/nsnam/ns/>

LOOKAHEAD ACCUMULATION IN CONSERVATIVE PARALLEL DISCRETE EVENT SIMULATION

Jan Lemeire, Erik Dirckx
Parallel Systems lab,
Vrije Universiteit Brussel (VUB)
Brussels, Belgium
{jlemeire, erik }@info.vub.ac.be

KEYWORDS

Discrete event simulation, Parallel simulation, Conservative algorithms, Lookahead.

ABSTRACT

Lookahead is a key issue in distributed discrete event simulation. It becomes explicit in conservative simulation algorithms, where the two major approaches are the asynchronous null-message (CMB) algorithms and the synchronous window algorithms (CTW). In this paper we demonstrate how a hybrid algorithm can maximize the lookahead capabilities of a model by lookahead accumulation. Furthermore, per processor aggregation of the logical processes allows for tuning of the granularity. A qualitative performance analysis shows that in case of no hop-models our algorithm outperforms the traditional conservative algorithms. This is due to reduced synchronization overhead caused by longer independent computation cycles, that are generated by the lookahead accumulation across the shortest lookahead path.

INTRODUCTION

This paper deals with parallel discrete event simulation (PDES) (Ferscha 1995, Fujimoto 1990) of logical process based models. There are 2 main approaches in conservative parallel simulation algorithms: the asynchronous approach, called CMB (after Chandy, Misra and Bryant), using null messages for synchronisation (Misra 1986, Lin 1995 & Ferscha 1995), and the synchronous window approach, CTW, (Conservative Time Windows) (Lubachevsky 1989, ayani 1992), which uses a window ceiling for synchronisation.

The algorithm that we developed is based on the deadlock avoidance CMB algorithm, and incorporates the concepts of the CTW approach. Our algorithm tries to maximize the performance by optimally tuning two attributes of the model: granularity and lookahead. Granularity or grain size is defined as 'amount of computations between communication points' (Choi 1995). Our algorithm tries to get better performance by maximizing the granularity and thus attaining less communication overhead. This is done by per processor aggregation of all its dedicated logical processes forming a multiprocess, which can be simulated

sequentially on each processor (Brissinck 1995, Praehofer 1994).

Next to granularity, our algorithm exploits maximally the performance gain coming with the lookahead capacities of the model. Better lookahead leads to less synchronisation overhead and better load (Preiss 1990, Peterson 1993, Fujimoto 1988). Our algorithm tries to accumulate lookahead while calculating the global lookahead of the multiprocess.

The next section explains the algorithm, section 3 discusses the various aspects of the algorithm and compares it with the traditional conservative algorithms. Section 4 analyses the performance on a qualitative basis and compares it with CMB & CTW performance. Section 5 finally shows the impact on 2 example models.

THE SYNCHONIZATION ALGORITHM

At first, the model is partitioned among the available processors and all logical processes on the same processor are aggregated to form a multiprocess. Parallel simulation happens in cycles of independent simulation alternated with communication of the events that travel through the channels connecting the multiprocesses. The independent simulation phase on each processor is based on the chronological processing of all events that are ordered in an *event queue*. This corresponds with 'normal' sequential simulation.

Since we use the conservative approach, simulation is only continued when all events are known until that time. The synchronization algorithm will calculate this safe time. Our algorithm therefore needs to synchronize the multiprocesses and the simulation inside each multiprocess.

Synchronization of the multiprocesses is based on the CTW approach. After a phase of independent simulation, a multiprocess will send outgoing events to the other multiprocesses. It then waits for receiving incoming events at the incoming channels from the neighbor multiprocesses. All events come together with a time window. The window assures that all events during that time period are known, so that simulation can advance. Figure 1 shows a multiprocess consisting of 5 logical processes. It has 2 input channels I_1 and I_2 at which it receives 2 windows with ceiling t_1 and t_2 .

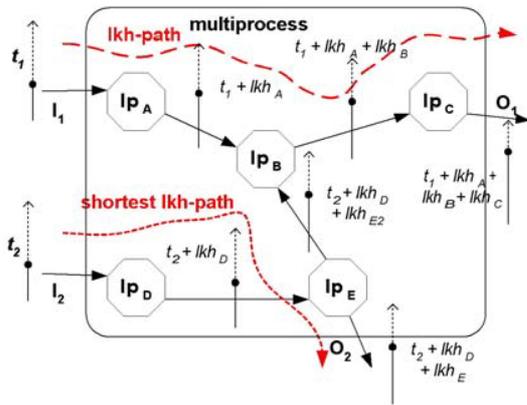


Figure 1: Synchronization Inside the Multiprocess (assume that $t_2 + lkh_D + lkh_E < t_1 + lkh_A + lkh_B + lkh_C$)

Synchronization inside each multiprocess is based on a deadlock avoidance CMB approach that uses null messages to indicate safe simulation. Null messages or null events are defined as 'a promise not to send any other message with smaller timestamp in the future' (Ferscha 1995). At start of each simulation cycle, null messages are scheduled for all global inputs of the multiprocess at the time of the incoming window ceiling. After a null message has arrived in a process, *conditional events* are possible, because it is not sure that *all* events are known in the local queue for that time. In our algorithm, a logical process will simulate *until* a first null message appears at *one* input, whereas in CMB algorithms a process has to *wait* for null messages at *all* inputs before it can simulate. This is possible because inside a multiprocess normal and null events are processed in chronological order, sharing the same event queue. When a null message enters a process, this process is *killed*, stopping the simulation of that process for that cycle. Since future events are *conditional* and may not be processed during the present cycle. They are scheduled in the conditional queue to be processed in the next cycle. Next, when the process is killed, null messages are scheduled for all output channels at the local virtual time plus the processes' lookahead. They will kill the succeeding processes (Figure 2). The first null message arriving at an outgoing channel of the multiprocess determines the window ceiling of the window that will be sent (O_2 in Figure 1). In the initialization phase of the simulation, the first windows are generated. A cycle of independent simulation is performed with empty windows at all inputs (ceiling time zero). The edge processes will be killed at time 0, generating lookahead-incremented null messages for the succeeding processes. In this way, the first global lookahead together with the first output events are generated at the processor outputs for the initial synchronisation of the multiprocesses (Fig. 2 at the right).

By construction, no event is simulated after a null message or outside a time window, hence the correctness of our synchronization algorithm is proven.

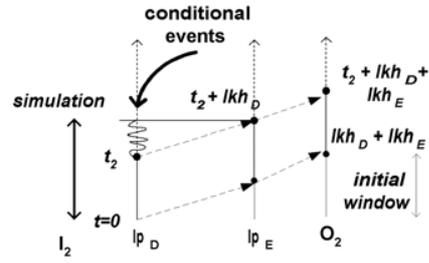


Figure 2: Null Message Propagation

Our algorithm can be seen as a window algorithm: each multiprocess receives a safe window to simulate. More precisely, each channel has a safe simulation window during each cycle. However, it is an *asynchronous* algorithm. There is no global (barrier) synchronization as with CTW algorithms. Each multiprocess decides independently when and how much it can simulate, like in CMB algorithms.

DISCUSSION OF THE CYCLE TIME

The shortest lookahead-path

Simulation takes place in cycles of communication and independent simulation. A simulation cycle on a processor lasts until the first output is reached by a null event. This null event is generated by a previously killed process, which on his turn is killed by another null event, etc. This chain of null events starts at a certain input and propagates through the model, forming what we call a *lookahead path*, and ending at an multiprocess output (Figure 1).

Each global output will be killed by a lookahead-path. The *shortest lookahead-path* kills the first output and determines thus the cycle size.

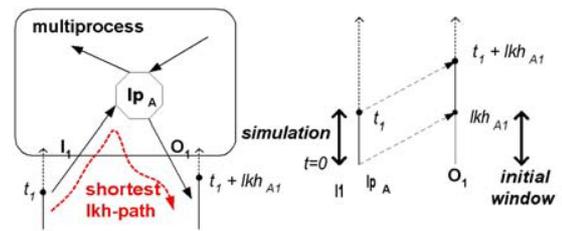


Figure 3: Models with a Hop

If the shortest lookahead-path passes through multiple processes we can speak about *lookahead accumulation*, the global lookahead of the multiprocess is formed by the sum of the lookahead of all processes in the path. If on the contrary the shortest lookahead-path comes in and leaves the multiprocess out immediately, we talk about a *hop* (Figure 3). For those models, there is no lookahead accumulation and the global lookahead simply equals the lookahead of the edge process.

The shortest lookahead-path is the largest possible safe simulation cycle. By construction, any larger cycle can cause conditional events.

The cycle in CMB algorithms

In a similar way, a cycle can also be defined for CMB algorithms. The cycle is defined as the frequency of event communication and null event generation. In the example of figure 4 (after Lin 1995), the process can simulate in cycles of the sum of the lookaheads. We see that it is also determined by the shortest lookahead path from a process output back to an input. Each process got its own shortest lookahead path, as opposed to our algorithm where it is calculated per multiprocess.

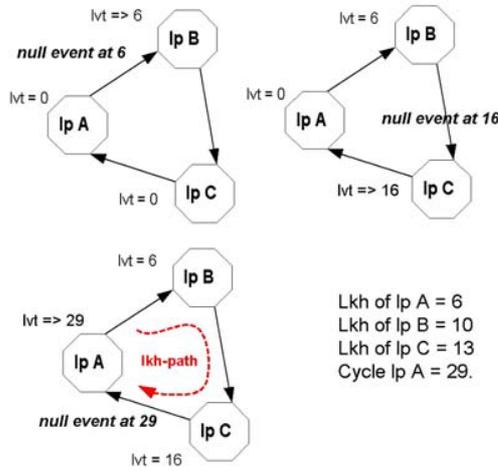


Figure 4: Cycle in CMB algorithms

The cycle in CTW algorithms

For conservative time window algorithms, the window size is calculated with the minimal lookahead of the edge processes (connected with other multiprocesses). The main CTW algorithms define a *distance* (Ayani 1992), an *event horizon* (Steinman 1994) or a *mimimum propagation delay* (static lookahead) with *opaque periods* (dynamic lookahead) (see the bounded lag algorithm, lubachevsky 1989). All these concepts reflect the lookahead of a process. For these algorithms, no lookahead accumulation takes place and thus is the cycle time the same as in our algorithm for hop-models.

QUALITATIVE PERFORMANCE ANALYSIS

This section discusses the performance of the algorithm and compares it qualitatively with the two traditional approaches. As a first order approximation, we assume the sequential simulation time $SeqSimT$ to be proportional to the number of simulated events $\#ev_{Sim}$:

$$SeqSimT = C_1 \cdot \#ev_{Sim} \quad (1)$$

Parallel simulation on p processors is then the simulation of p times less events plus the overhead induced by the parallel nature of the simulation:

$$ParSimT = C_1 \cdot \#ev_{Sim} / p + \sum_i^{#O} overheadT_i \quad (2)$$

with $ParSimT$ the parallel simulation time and $overheadT_i$ the time of overhead i , ranging from 1 to $\#O$, the number of overheads. The first term of the equation assumes equal distribution of the events among the processors, corresponding with an ideal parallel processing. The effect of unequal distributions must be seen as overhead and added to the right term. Performance is measured by the speedup, which is the ratio of sequential simulation time versus the parallel simulation time. The impact of the overhead on the speedup S is then the ratio of the overhead time with the ideal parallel simulation time:

$$S = \frac{C_1 \cdot \#ev_{Sim}}{C_1 \cdot \#ev_{Sim} / p + \sum_i^{#O} overheadT_i} \quad (3)$$

$$S = \frac{p}{1 + \sum_i^{#O} \frac{overheadT_i}{C_1 \cdot \#ev_{Sim} / p}} = \frac{p}{1 + \sum_i^{#O} Ovh_i} \quad (4)$$

The ratio $overheadT_i / ParSimT$ is defined as the *overhead ratio* Ovh_i of overhead i .

Our parallel simulation algorithm generates 3 main types of overhead: communication, synchronization and idle time. These result in 5 overhead ratios Ovh_i and 5 performance factors reflecting the impact of simulation statistics on the different overheads (Lemeire 2001), as shown in Table 1.

Table 1: Overhead classification of the conservative simulation algorithm

Overheads		Overhead Ratios	Performance factors
Communication	Ovh ₁	per event overhead	$\#ev_{Comm} / \#ev_{Sim}$
	Ovh ₂	constant overhead	$\#ev_{Sim} / \text{cycle}$
Synchronisation	Ovh ₃	synchronization	$\#ev_{Null} / \#ev_{Sim}$
	Ovh ₄	conditional queue	$\#ev_{Cond} / \#ev_{Sim}$
Idle time	Ovh ₅	load imbalance	Differences in $\#ev_{Sim}$ per processor

The communication overhead is the time not overlapping with computation for communicating the events. This can be split in the variable overhead (Ovh_1), proportional to the data size, and the constant communication overhead (Ovh_2), induced by setting up

the communication link. The communication overhead ratio Ovh_1 is proportional to the number of communicated events between the processors ($\#ev_{Comm}$) versus the number of simulated events. This results in the first performance factor, namely $\#ev_{Comm}/\#ev_{Sim}$. The constant overhead ratio Ovh_2 leads to $\#ev_{Sim}/Cycle$, the number of simulated events per cycle. This ratio is also called **granularity** or grain size (also event simultaneity in Peterson93).

The synchronization overhead is the processing in each cycle of the synchronisation information. For CMB-algorithms and our algorithm this is the null event processing, whereas for CTW-algorithms it is the window size calculation. The processing time for this depends in the first place on the number of null events $\#ev_{Null}$. This results for Ovh_3 in a performance factor $\#ev_{Null}/\#ev_{Sim}$. Our algorithm induces an extra synchronization overhead (Ovh_4) due to the conditional events $\#ev_{Cond}$ that are queued to be processed in the next cycle. This leads to a constant overhead and one proportional to $\#ev_{Cond}/\#ev_{Sim}$.

Unequal simulation phases on the different processors lead to idling, when processors have to wait for incoming events. This is mainly caused by load imbalances, here unequal number of events to be simulated. This overhead ratio (Ovh_5) is proportional to the relative deviation of the number of events simulated on each processor.

The Lookahead Accumulation Benefit

The synchronization algorithm influences all but the per event communication overhead Ovh_1 , which is only determined by the model partitioning. The other overheads depend on the cycle time (Peterson 1993, Choi 1995), which is determined by the lookahead properties of the model. In case of real lookahead accumulation (no-hop models), our algorithm gets larger cycles and will attain a better performance. There will be less constant communication overhead (Ovh_2), less synchronisation overhead (Ovh_3), discussed in the next section, and better elimination of temporal load imbalances (Ovh_5).

The Synchronization Overhead

The per cycle synchronization calculation depends strongly on the algorithm. For CMB algorithms, it is the processing of one null event per channel, whereas for CTW, it is proportional to the number of edge-processes. The synchronization information is thus the lowest for the CTW, and the highest for CMB approaches. In our algorithm it is one null event per interconnection plus the depth of the null event propagation. In case of a hop model, our algorithm loses the lookahead accumulation advantage, the synchronisation overhead will be similar as with CTW (only the lookahead of the edge-processes is taken into account) and so the performance will be equal.

The synchronization overhead is proportional to the cycle frequency. For CTW algorithms it is also proportional to the number of interconnections. Whereas for CMB, the number of null events per cycle equals the number of channels. Our algorithm performs in between both: the number of null events per cycle is proportional to the number of interconnections plus the depth of the lookahead propagation

Note that a lot of modern algorithms optimize the synchronization overhead, like diverse null event reduction techniques in CMB algorithms (Ferscha 1995, etc) and for example, the bounded lag in Lubachevsky's CTW algorithm (Lubachevsky 1989).

The overhead Ovh_4 is specific for our algorithm. The cost for the extra lookahead of our algorithm is the conditional queue. In case of a hop, simulation will stop by the first killed process, no other processes were killed so far and thus, there are no conditional events and no conditional queue overhead. But in case of lookahead accumulation, conditional events of the killed processes must be stored in the conditional queue to be simulated in the next cycle. These extra operations cause the extra overhead: the check whether the process is killed and the queuing. These events come in chronological order out of the event queue and therefore sorting of the conditional queue is not necessary. This results in one extra operation for each event and one for each conditional event. In Figure 2 it can be seen that the number of conditional events could reach half of the number of processed events, as for lp D. But in most cases, it will be much less, because the last lookahead of the lookahead-path causes no conditional events. Moreover, deep processes (far from the edge) will not be killed soon. In total, the extra overhead is thus between 1 and maximally 1.5 extra operations (check and append) per simulated event, which will be much smaller compared to the time to simulate one event C_1 . We can conclude that the extra overhead induced by our algorithm is small, as is confirmed by the experimental results.

EXPERIMENTS

Two models will demonstrate our claims. One gives good results by exploiting the lookahead accumulation, while the other fails due to low lookahead. Both are simulated on a cluster of 4 Pentium II processors of 333MHz connected by a 100Mb/s non-blocking switch.

Fpga

Field Programmable Gate Arrays (FPGAs) are prefabricated devices used to implement digital logic. They feature a matrix structure of logic cells interconnected by routing channels, and a periphery of I/O cells. FPGAs can be programmed by a stream of configuration bits to form a logic circuit. The simulation model consists of 2387 processes and 10978 channels (Bousis 2000). Geometrical partitioning (the dashed

lines in Figure 5) gives best load balancing and least communication. However, the model is heavily interconnected and contains many hops (namely 453). The shortest lookahead path is only 8 ns, resulting in only 70 events simulated per cycle of 8ns. The performance results are shown in Table 2.

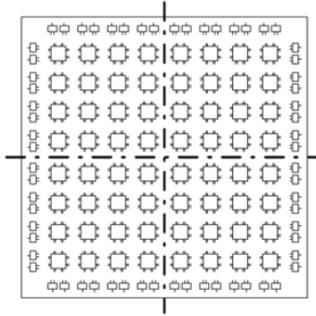


Figure 5: FPGA Model with Partitioning

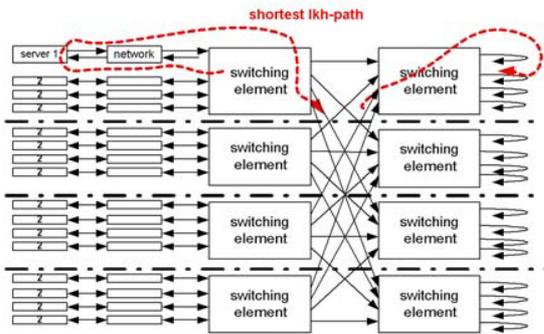


Figure 6: ATM Switch Model with Partitioning and Shortest Lookahead Path

ATM switch

The high capacity ATM switch model (Geudens 2000) demonstrates the benefits of our algorithm (Figure 6). The model consists of a detailed 4 by 4 switch with 16 entries. Each input receives IP-traffic by a simulated network.

Table 2: Performance results for parallel simulation with 4 processors

	FPGA	ATM switch
Global Performance		
Speedup	0.74	3.5
#ev _{Sim} per realtime second	6592 events/s	44000 events/s
Cycle time	8ns	50000ns
Communication overhead		
Ovh ₁ #ev _{Com} / #ev _{Sim}	18%	5.7%
Ovh ₂ #ev _{Sim} / Cycle	70	10100
Synchronisation overhead		
Ovh ₃ #ev _{Null} / #ev _{Sim}	470%	0.45%
Ovh ₄ #ev _{Cond} / #ev _{Sim}	0	1.6%
Ovh ₅ Idle time	9.4%	11%

Here again, a geometrical partitioning (horizontal) is the only plausible one (dashed lines in Figure 6). The model can accumulate the lookahead along a path that leaves the switch, passes the network, enters the server and returns back to the switch. This results in long cycles giving an quasi ideal speedup of 3.5 as shown in Table 2.

CONCLUSION

In this paper, we demonstrated the benefit of accumulating lookahead with a hybrid conservative parallel simulation algorithm, based on per processor aggregation of its processes. The processors' global lookahead is determined by lookahead accumulation across the shortest lookahead path, which results in longer simulation cycles.

A qualitative performance analysis showed that our algorithm gets a performance benefit over the traditional (non-optimized) conservative algorithms CMB (asynchronous null-message algorithms) and CTW (synchronous window algorithms) in case of partitioned models without 'hops'.

REFERENCES

- Ayani R., Rajaei H. "Parallel simulation using conservative time windows". In 1992 *Winter Simulation Conferences Proceedings*, pp 709-717, 1992.
- Bousis L. "Study and Implementation of a Scalable Simulator for Complex Digital Systems". Master Thesis, Free University of Brussels, 2000.
- Brissinck W., Steenhaut K., Dirx E. "A Combined Sequential/Distributed Algorithm for Discrete Simulation". *Proceedings of IASTED, Modelling and Simulation*, Pennsylvania, 1995.
- Choi E., Chung M. J. "An important factor for optimistic protocol on distributed systems: granularity". In 1995 *Winter Simulation Conferences Proceedings*, pp 642-649, 1995.
- Ferscha A. "Parallel and Distributed Simulation of Discrete Event Systems". *Handbook of Parallel and Distributed Computing*, McGraw-Hill, 1995.
- Fujimoto R.M. "Parallel Discrete Event Simulation". *Communications of the ACM*, 33, pp 29-53, October 1990.
- Fujimoto R.M. Performance "Measurements of Distributed Simulation Strategies". *Proc. 1988 SCS Multiconference on Distributed Simulation Strategies*, pp 14-20, February 1988.
- Geudens S. "Quantitative Study of a Highly Formant Network Switch with Distributed Simulation". Master Thesis, Free University of Brussels, 2000.
- Lemeire, J. and Dirx, E.: "Performance Factors in Parallel Discrete Event Simulation". In: *Proc. of the 15th European Simulation Multiconference (ESM)*, Prague, 2001.
- Lin Y., Fishwick P.A. "Asynchronous Parallel Discrete Event Simulation". 1995.
- Lubachevsky B.D. "Efficient distributed event-driven simulations of multiple-loop networks". *Communications of the ACM*, 32, 111-123. 1989.
- Misra J. "Distributed Discrete-Event Simulation". *ACM Computing Surveys*, Vol. 18, No. 1, March 1986.

- Peterson G.D., Chamberlain R.D. "Exploiting lookahead in synchronous parallel simulation". In *1993 Winter Simulation Conferences Proceedings*, pp 706-712, 1993.
- Praehofer H. and Resinger G. "Distributed Simulation of DEVS-Based Multiformalism Models". IEEE, 1994.
- Preiss B.R., Loucks W.M. "The impact of Lookahead on the Performance of Conservative Distributed Simulation". 1990.
- Steinman J.S. "Discrete-event simulation and the event horizon". *Proceedings of the 8th Workshop on Parallel and Distributed Simulation(PADS)*, 1994.

AUTHOR BIOGRAPHIES

JAN LEMEIRE was born in Malmédy, Belgium. He obtained his masters degree in electrotechnics engineering in 1994 at the VUB. After an additional masters in Computer Science, he started working for 3.5 years in the private sector. First for Cap Gemini, an IT consulting firm, then for Warmoes & Van Damme, a company specialised in knowledge systems. There, he developed his professional skills, but in 1999 he returned to the VUB to prove himself in a scientific

career. He was first allocated on a project on parallel simulation in cooperation with Alcatel Bell . However, since march 2001 he is employed as an assistant, teaching with a lot of enthusiasm and pursuing a PhD about parallel performance and inference. His e-mail address is Jan.Lemeire@vub.ac.be and his webpage is <http://parallel.vub.ac.be/~jan>.

ERIK DIRKX was born in Brussels, Belgium. He obtained a MSc. in Electrotechnics Engineering, in Computer Science, an MBA and a PhD in Computer Science at the Vrije Universiteit Brussel. He was a visiting scientist at IBM T.J. Watson Research lab, ETL-Tsukuba (Japan) and Xilinx Research lab (San Jose CA). He is currently an associate professor at the VUB, teaching and coordinating research in the field of parallel and distributed computing. His e-mail address is : Erik.Dirks@vub.ac.be and his webpage can be found at <http://parallel.vub.ac.be/~efdirkx>.

RATE ADJUSTMENT MODES FOR RESILIENT PACKET RING NETWORKS

Ahmad M. Al-Banna and Saleh R. Al-Araji
College of Engineering and Information Sciences
Etisalat University
Sharjah, UAE
Email: alarajis@ece.ac.ae

KEYWORDS

High Performance Area Networks, Communication Networks and Protocols, Resilient Packet Ring (RPR), RPR Fairness algorithm

ABSTRACT

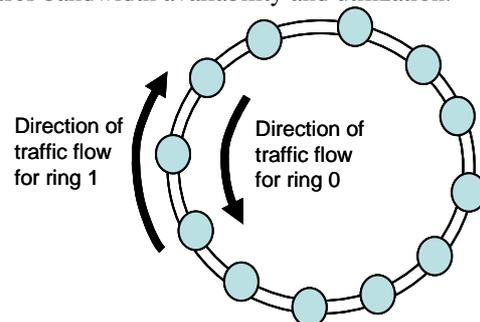
The RPR rate adjustment modes are analyzed and simulated with short and long distances between stations using Java based model. The conservative and aggressive modes were investigated. The former mode in most cases is more stable than the aggressive mode. The instability of the aggressive mode, which will lead to rate oscillation, is clearly shown when the separation between stations is large.

1.0 INTRODUCTION

Today's networks require a migration of packet-based technologies from Local Area Networks to Metropolitan Area Networks (MANs). The fast increase of data traffic in MAN networks is challenging the capacity limits of existing circuit-oriented technologies like SONET and ATM. Carrying an increased data traffic over voice-optimized circuit-switched makes the capacity inefficiently managed and difficult to provision new services. Packet-based transport technology is considered to be the only alternative for scaling metropolitan area networks to meet the demand [1,2]. An IEEE 802 Resilient Packet Ring (RPR) Standard Committee is currently working to set the standards for this new packet based technology that will solve the problems associated with the circuit switched technology.

Resilient Packet Ring (RPR) Figures 1, for MAN, and WAN regional networks, is a new media access control (MAC) protocol closely related to Ethernet but designed to optimize bandwidth utilization and facilitate services over a ring network. It is designed to provide the carrier-class attributes normally associated with SONET and SDH. RPR is a set of switching nodes interconnected along a bi-directional, double fiber ring. Data is transmitted and load balanced on both rings. Unlike ring-based LAN technologies, RPR packets do

not have to circulate the full ring, which effectively increases bandwidth availability and utilization.



Figures 1: Typical 12 Node RPR Ring Topology.

The RPR MAC will offer the following features:

- New data link layer technology (new MAC layer)
- Ring protection and fast restoration
- Support of multiple classes of service
- Controlled dynamic bandwidth on the ring
- Controlled latency and jitter
- Controlled traffic congestion

RPR implements a three level classes based on traffic priority scheme. The aim of the class based scheme is to let class A be a low latency and low jitter class, class B be a class with predictable latency and jitter, and finally class C be a best effort transport class. On the other hand RPR ring does not discard frames to resolve congestion. Hence when a frame has been added on to the ring, even if it is a class C frame, it will eventually arrive at its destination.

Class A traffic is divided into subclasses A0 and A1, and class B traffic is divided into class B-CIR (Committed Information Rate) and B-EIR (Excess Information Rate). The two traffic classes C and B-EIR are called Fairness Eligible (FE).

The bandwidth around the ring is pre-allocated in two ways. The first is called "reserved", which can only be used by class A0 traffic. If stations are not using their pre-allocated A0 bandwidth, this bandwidth is wasted. In this way TDM-like traffic can be sent by RPR stations as A0 frames. The other pre-allocated bandwidth is called "reclaimable". A station that has class A1 or B-CIR traffic to send, preallocates "reclaimable" bandwidth for these types of traffic. If not in use, such bandwidth can be used by FE traffic. In

addition, any bandwidth not pre-allocated is also used to send FE traffic. The distribution and use of unallocated and unused reclaimable bandwidth is dynamically controlled by the fairness algorithm.

The objective of the fairness algorithm is to distribute unallocated and unused reclaimable bandwidth fairly among the contending stations and use this bandwidth to send class B-EIR and class C traffic, i.e. the fairness eligible (FE) traffic. Class A0 traffic is obviously not affected, since bandwidth is reserved for this class exclusively. Classes A1 and B-CIR are indirectly affected, as will be explained below.

The fairness algorithm starts working to distribute bandwidth fairly when the bandwidth on the output link of a station is exhausted (the link is congested). The most probable cause of congestion is the station itself and its immediate upstream neighbors. Hence by sending a so called fairness message upstream (on the opposite ring) the probable cause of the congestion is reached faster than by sending the fairness message downstream over the congested link [5]. In the proposed work, the impact of RPR fairness algorithm on traffic congestion will be analyzed, and simulated.

2.0 FAIRNESS ALGORITHM

The fairness algorithm is within the MAC control sub-layer. There are two fairness instances in each station, one for each ringlet that will support independent fairness operation in each ringlet. The use of fair rates prevents one station from occupying the shared bandwidth with respect to other stations on the ringlet. The fairness algorithm controls the access of fairness eligible traffic (class C and class B Excess Information Rate (EIR)) to a ringlet is as follows:

- The traffic congestion is controlled.
- The activities of the controlled congestion will have a minimum effect on the throughput.
- A fair rate limit would be applied across stations contributing to congestion.

2.1 IDENTIFYING CONGESTION

A station is congested when one or more of the following conditions are identified:

- i) If the secondary transit queue (STQ) is exceeding certain limit.
- ii) If the transmission rate exceeds the link bandwidth.
- iii) When the traffic is delayed excessively while awaiting transmission.

Congestion is undesirable scenario as it can result in a failure to meet the end-to-end commitments relative to the service classes.

Condition (i) is applicable only in the case of a dual-queue MAC. Conditions (ii) & (iii) are applicable only in the case of a single-queue MAC.

2.2 RATE ADJUSTMENT MODES

There are two modes of rate adjustments in RPR namely “ Aggressive” and “Conservative”. The former provides responsive adjustments that favor utilization of capacity over rate stability. Where as the conservative mode provides highly damped adjustments that favor rate stability over utilization of capacity.

The main difference between conservative and aggressive fairness is the way the fair rate is initially estimated, and how it is adjusted towards the real fair rate.

In the conservative mode, the congested station calculates the initial fair rate either by 1) dividing the available bandwidth between all upstream stations that are currently sending frames through this station or by 2) use its own current add rate. A timer is used to ensure that additional rate changes is made only when the congested station have had time to see how this new fair rate affects the congestion (i.e., gets better or worse). The period of this timer is referred to as the Fairness Round Trip Time (FRTT). FRTT is an estimate of the time it takes for a congested station to see the full effect of the fairness message it sent to upstream stations. FRTT consists of two parts: 1) the propagation delay for a class A frame when transmitted from the congestion domain head (i.e. the congested station) to the congestion domain tail (the station at the other end of the congestion domain) and back (LRTT – Loop Round Trip Time). 2) The difference between the propagation delay for a class C and a class A frame is sent from the tail to the head (FDD – Fairness Differential Delay). LRTT needs to be computed on initialization of the ring and when the topology changes, while FDD is computed when a station becomes tail of a congestion domain and thereafter at configurable intervals. FDD reflects the congestion situation, i.e. the STQ fill levels on the transit path from head to tail. As the congestion domain changes, so does the FRTT. LRTT and FDD frames are special types of control frames.

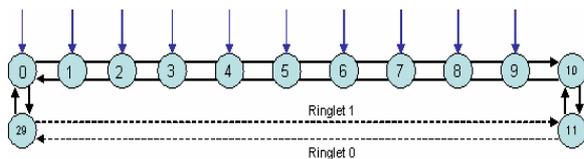
In the Aggressive mode, the congested station makes a first estimate of the fair rate equal to the rate the station itself lately have been able to add to the ring. Since the station is congested, this means that it has been able to send very little traffic onto the ring recently. Hence this estimate is probably too low, but it is used as a starting point and a way to alleviate congestion. When congestion is indeed removed, the (previously congested) station will not send any more fairness messages upstream, or more correctly it will send fairness messages with a default fair value representing the full link rate (such frames are sent all the time with preset intervals as heart beats.) A station receiving a fairness message indicating no congestion (i.e., full link rate) will increase its add traffic (assuming the station’s demand is greater than what it is currently adding). In this way (if the traffic pattern is stable) the same station will become congested again after a while, but this time the estimated fair rate will be closer to the real fair rate,

and hence the upstream stations do not have to decrease their traffic rate as much as previously [5].

3.0 SIMULATION RESULTS

An RPR ring was simulated to generate the results for both aggressive and conservative rate adjustment modes [3]. Instead of using the usual OPNET simulation tool for RPR, the authors simulated these conditions with Java based program that was recently developed by Simula Research Laboratory, Norway [4].

The number of stations used in this simulation is 30 as shown in Figures 2. Station 0 to 9 send class C greedy user datagram protocol (UDP) traffic on ringlet (0) to station 10, with aging and advertising intervals of 100 microseconds each. The distance between each RPR station on this ring is 50 Km resulting a delay of 0.25msec [6]. The link speed is 622 Mbps and the Packet size is 500 Byte. Table 1 shows the sending start and stop time of the UDP traffic from the different stations 0-9 to station number 10.

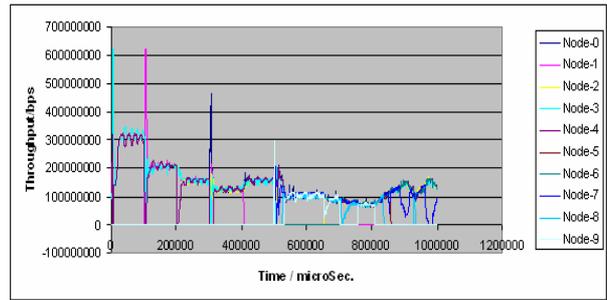


Figures 2 RPR Ring with 30 Stations

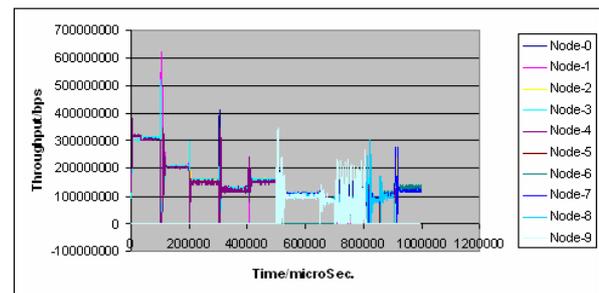
Table 1: Start-Stop Stations Timing

Station Number	Start Time (second)	Stop Time (second)
0	0.3	0.8
1	0.1	0.4
2	0.2	0.5 and Start Again at 0.65 and stop at the 1 second
3	0	1
4	0	1
5	0.7	0.85
6	0.7	1
7	0.5	1
8	0.5	0.9
9	0.5	0.8

The simulation plots for conservative and aggressive modes are shown in Figures 3 and 4. It can be seen from the plots that the aggressive mode shows more oscillation in the region between 0.7-0.9 sec when compared with the conservative mode. This is due to the high number of starting and stopping flows in that time frame.

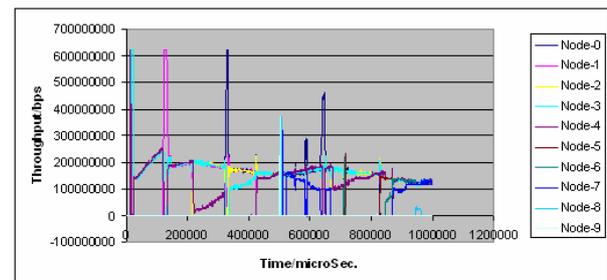


Figures 3: Conservative Mode for all Stations 0-9 Sending UDP Traffic to Station 10

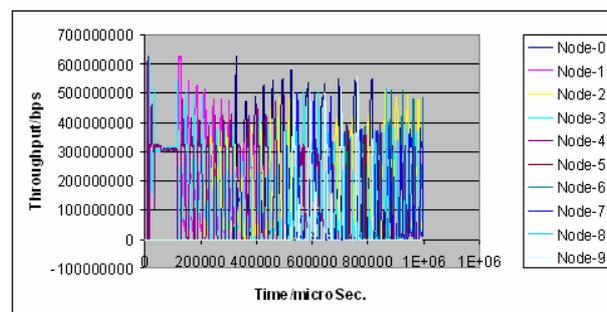


Figures 4: Aggressive Mode for all Stations 0-9 Sending UDP Traffic to Station 10

The network is now simulated with a new distance between each node of 400Km while keeping the same specification as above. The resulting propagation delay of from this change of distance is 2msec. It can be seen from the plots of Figures 5 and 6 that the conservative mode has minimal oscillation when compared with aggressive mode.



Figures 5: Conservative Mode for all Stations 0-9 Sending UDP Traffic to Station 10



Figures 6: Aggressive Mode for all Stations 0-9 Sending UDP Traffic to Station 10

From the figures of throughput versus time shown below, it can be seen that the conservative mode in general is more stable than the aggressive mode particularly at large distances between stations. The figures shown are only samples of the total simulation results.

4.0 CONCLUSIONS

The simulation results have clearly shown that the conservative mode of UDP traffic in an RPR mode is more stable than the aggressive mode and not normally subjected to oscillation. In the aggressive mode, increasing the distance between RPR stations will dramatically affect the traffic pattern and it will lead to oscillation. The Aggressive rate adjustment is therefore not recommended for large distance separation between stations.

5.0 REFERENCES

1. Minoli, D., Johnson, P., and Minoli, E., "Ethernet-Based Metro Area Networks", McGraw Hill, 2002.
2. IEEE 802.17 RPR Draft 2.6, September 2003, Las Vegas, NV, USA.
3. Ramakrishnan, K. K., Interactions between Class A1, Class B and Class C Traffic with Conservative Mode, IEEE802.17RPR Interim Meeting, Montreal, PQ, Canada, May 2003.
4. Gjessing, S., and Maus, A., A Fairness Algorithm for High-speed Networks based on a Resilient Packet Ring Architecture, IEEE Proceedings of the International Conference on Systems, Man and Cybernetics, 2002.
5. Davik, F., Gjessing, S., Uzun, N., Yilmaz, M., IEEE 802.17 Resilient Packet Ring Tutorial, March 2004.
6. Stein Gjessing, The Simula RPR Simulator implemented in Java, Simula Research Laboratory Technical Report 2003-12, December 2003.

AUTHOR BIOGRAPHIES

AHMAD M ALBANNA was born in Dubai, UAE and completed his B.Eng degree, with first class honors, in Computer Engineering at Etisalat College of Engineering, Sharjah, UAE in 2001. After graduation, Mr. Al Banna joined Etisalat (the Telecom Company in UAE) working as an Internet Backbone engineer and later was promoted to his current position as a project manager in the IP Development Department and at the same time working on his MSc. Thesis in the field of Resilient Packet Ring (RPR). His e-mail address is : albanna@emix.net.ae

Professor SALEH AL-ARAJI received the B.Sc., M.Sc., and Ph.D. degrees from the university of Wales Swansea (UK) in 1968, 1969, and 1972 respectively. Since September 2002, Professor Al-Araji was appointed Professor and Head of Communications Engineering Department at the College of Engineering and Information Sciences, Etisalat University, Sharjah, UAE. Prior to that and for six years he was working at the Transmission Network Systems, Scientific-Atlanta, Atlanta, Georgia, USA as Senior Staff Electrical Engineer. During the academic year 1995/1996, Prof. Al-Araji was a visiting professor at the Ohio State University, Columbus, Ohio. He was also visiting professor at King's College, University of London, England, during the summers of 1988 and 1989. Prof. Al-Araji was professor and Department Head at the University of Baghdad, Iraq, and the University of Yarmouk, Jordan.

Prof. Al-Araji was awarded the British IERE Clerk Maxwell Premium for a paper published in 1976 and the Scientific-Atlanta award for outstanding achievement in 2000. He was an Iraqi National member of URSI commissions C and D, and the ITU (CCIR Group 8). He has published over 40 papers in international Journals and Conferences and holds 5 US Patents and one International Patent. Prof. Al-Araji is a senior member of the IEEE. His e-mail address is : alarajis@ece.ac.ae

HADES – A Highly Available Distributed Main-Memory Reliable Storage

Matthias Meixner, Alejandro Buchmann
Datenbanken und Verteilte Systeme
Department of Computer Science, Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany
E-mail: {meixner,buchmann}@informatik.tu-darmstadt.de

Abstract

Fast persistent storage is a requirement in many applications in which slow disk access times become the bottleneck. This paper describes HADES, a main memory storage system, that uses a fault tolerant partitioning scheme to reliably store data in main memory of a distributed computer system and is, therefore, able to improve performance. It presents the principles involved, discusses performance evaluation and compares HADES to other systems.

1 Introduction

We are starting to see new applications that make high demands on data storage systems regarding both availability and access time. Examples are highly available systems using IP take-over and publish/subscribe systems.

1.1 IP take-over

One practical way of achieving high availability in distributed systems is IP take-over (Fetzer et al., 2003). A service is bound to an IP address and port number. If the service fails, the client tries to reconnect, which is detected and another server takes over the IP address and continues to provide the service.

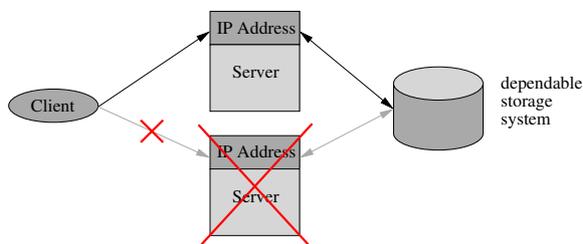


Figure 1: IP take-over

To process client requests correctly, most services need to keep some state. When a server crashes, state stored in memory is lost. One way of dealing with this problem is to source out state information to stable storage and treat the service as if it was stateless (figure 1).

The response time of this system heavily depends on the performance of the storage system. Each operation that changes the state that is associated with a client must go to the storage system before replying to the client so that another server could resume the operation in case of a failure. In this application it is of capital importance that the storage system is highly available, since the availability of the whole system depends on it.

1.2 Publish/Subscribe Systems

Publish/subscribe systems are used to build large distributed systems in which the components are loosely coupled (Eugster et al., 2003; Carzaniga et al., 2000):

- A component does not need to know which other components exist. It just needs to know about the format and structure of events that it is interested in.
- Components can be added and removed without directly affecting other components.

Components communicate via *notifications* that signal the occurrence of an event. Clients subscribe to notifications. Within the *event notification system event brokers* use this information to forward notifications from producers to consumers. Filters are used to avoid flooding and limit the forwarding to those consumers interested in certain notifications. Furthermore, filters may aggregate information (e.g. calculate min/max values over a period of time) or transform information (e.g. convert degrees Fahrenheit into degrees Celsius). Figure 2 gives an ex-



Figure 2: Flow of data in an event based system

ample of the flow of data from a producer to a consumer showing only the active components that are relevant for this flow of information. In this system there cannot be an end to end error correction, since a connection between the producer and the consumer does not exist and the producer does not know the identity or even the number of consumers. Therefore, within this system notifications may be lost or duplicated.

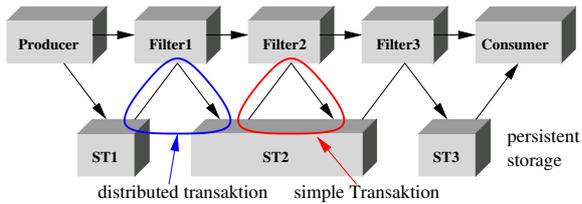


Figure 3: notifications and transactions

This property is not suitable for applications that depend on an “exactly once” delivery of notifications, e.g. applications that count the occurrence of certain events. The same applies to event aggregation that needs to store intermediate results without losing them. This problem can be solved by using transactions and persistent intermediate storage (figure 3). Notifications are passed on by transferring them from one persistent storage to another within a transaction. The atomicity of the transaction guarantees an exactly once delivery and the persistent storage guarantees that no notification is lost in case of a failure.

Using this approach poses some challenging requirements on the storage system: the end to end delay depends on the time needed for the data access required by the transaction, since the next filter in the path cannot start processing the data until the preceding operation has been committed. Therefore, the data access time is critical to the performance of the whole system. If the storage system goes down, the flow of notifications is interrupted until the storage system is restarted. To minimize this effect a storage system is needed that additionally provides high availability and reliability.

1.3 Common requirements

Both applications described above impose similar requirements on the storage system: the performance heavily depends on the access time which is dominated by latency. The amount of data to be stored is relatively small but data is modified frequently. For performance reasons we can assume that all data should be stored within the same LAN. Both applications require high availability of the storage system.

Conventional storage systems are not suitable to be used within these applications: the use of hard disks would be a bottleneck since even the fastest available disks have an access time of more than $5ms$. Flash ROM storage systems cannot be used due to the frequent modification of data: flash memory has a limited number of write cycles, e.g. 100000 for *Intel StrataFlash Memory (J3)*. DRAM based solid state disks do not have this limitation, but their cost is 2 orders of magnitude higher than that of conventional disks. For example for the price of an Athena-2 plus disk of 800MB one would easily get 8 complete computers with 8 GB of main memory in total. Furthermore, a single disk cannot provide high availability, therefore at least two of them are needed plus some RAID system that manages redundancy.

1.4 Aim of this paper

The aim of this paper is to describe how fault tolerant partitioning schemes can be used to provide fast reliable storage and improve the performance of the applications presented above. In section 2 we describe the implementation of HADES, a data management system, that offers fast access time and high availability through the use of fault tolerance. In section 3 we give a performance evaluation and in section 4 we discuss the features that distinguish HADES from existing systems. Finally we give a short summary and outlook in section 5.

2 HADES architecture

HADES follows the principle of achieving persistence by redundantly storing data in main memory of at least two nodes of a cluster. If a node fails, there is still a copy and data can be accessed without downtime. After a failure has been detected, redundancy is restored so that another failure can be masked.

2.1 Fault model

The fault model in HADES is based on is *Crash* (Gärtner, 1999), i.e. a computer operates flawlessly or not at all. While redundancy is reestablished after a failure, no other node in the cluster may fail. Node failures are detected using a distributed heartbeat based fault detector.

It is assumed that cluster nodes fail independently of each other. Therefore, all nodes should be guarded against power failure using an uninterruptable power source. A redundant network interconnection prevents that the connection to more than one computer is lost in case of network problems. Since all nodes are within the same LAN this does not pose a big problem. One simple solution is to use one switch per node and redundantly connect these switches. The spanning tree protocol used in the switches deactivates redundant links and reactivates them in case of a failure. A failure of a switch is equivalent to the failure of one single node, a network split does not occur.

2.2 Addressing

One of the main problems is to efficiently address data in the cluster. Network addresses cannot be part of an addressing scheme, since the location where data is stored may change in case of a node failure. The addressing scheme needs to support redundancy in a way that makes it possible to access the data even after a crash of one node. It must support the reestablishment of redundancy after a crash, i.e. it must be able to change the node where it expects to find data. The addressing scheme should provide an efficient data access, i.e. in normal operation no more than one request should be required to read or write data.

slice	primary server	secondary server		slice	primary server	secondary server		slice	primary server	secondary server
0	A	B	Reestablishing of redundancy →	0	A	B	→	0	B	A
1	A	B		1	A	B		1	A	B
2	B	C		2	B	A		2	B	A
3	B	C		3	B	A		3	B	A
4	C	A		4	A	B		4	A	B
5	C	A	5	A	B	5	A	B		

Figure 4: Data distribution: second stage

HADES solves this problem using the following addressing scheme: data is stored in pages that are numbered in ascending order (0, 1, 2, . . .). Unlike disk blocks or MMU-pages, pages in HADES are not of fixed length and may have an arbitrary size. Therefore, we can assume that one page holds one record without degrading the resource utilization.

HADES uses a two stage mapping from pages to nodes. The first stage maps page addresses to *slice addresses*, which is equivalent to a horizontal fragmentation of data:

$$slice_number = page_number \bmod slice_count$$

The slice count is constant and does not change even when nodes join or leave (fail) the cluster. It is specified at start time and it should be at least as high as the number of servers in the cluster considering possible future extensions. A higher number of slices results in a more uniform load distribution and offers better extensibility but at the same time generates a higher communication overhead.

The second stage uses a mapping table to map slice numbers to cluster nodes. It assigns a primary and secondary server to each slice. The secondary server acts as a backup server that can seamlessly take over the tasks of the primary server in case of a failure. The mapping adapts to achieve a uniform load distribution: each node is assigned not more than $2 \times \lceil \frac{slice_count}{number_of_nodes} \rceil$ slices (The factor of 2 is due to assigning a slice to two servers: primary and secondary server). When recalculating this distribution, after a node has failed or a new node has been added, HADES regards former data distributions and minimizes changes, thereby reducing the amount of data that needs to be copied to other nodes. Figure 4 gives an example of this: after a failure of node C data stored in slices 0 and 1 does not need to be copied, only data in slices 2-5 is copied to restore redundancy (highlighted in grey). A final swap of primary and secondary servers is used to achieve further load balancing between their roles as primary and secondary server. This mapping table is calculated and distributed by an elected coordinator, therefore problems with consistency cannot arise.

This addressing scheme offers several advantages: the first mapping stage is fixed and, therefore, it does not need to be transmitted to other nodes. The *mapping table* of the second stage needs to be updated only in case of a change of nodes due to node failure or insertion, in

particular it does not require an update when inserting or deleting data. The mapping table is small (it has only *slice_count* entries) and, therefore, it can be easily distributed to all other servers and clients. Using this mapping table clients are able to determine the storage location of data. Therefore, in normal operation only one request is required to access data.

2.3 Data consistency

As soon as several copies of data exist, there is always the problem of consistency: which copy of data is valid and may be accessed? To be used as a replacement for other storage systems, HADES provides similar guarantees regarding consistency: only write accesses may change data. While this sounds simple, it is not as the following example illustrates:

Client *A* writes page 1 which resides on server *X*. This page is read by client *B*. If server *X* fails before synchronizing page 1 with the backup server *Y* a subsequent read request of client *B* will read the previous content of page 1. In this case client *B* sees a change of data back to the old content although there was no write access. This must not happen.

To avoid these types of problems HADES uses one primary server for each slice (which may be different for different slices) that is responsible for maintaining data consistency. Clients communicate only with the primary server. This server coordinates read and write accesses and synchronization of data with the secondary server: write accesses are acknowledged only after the data has been successfully copied to the secondary server, read requests only read already synchronized data and, if necessary, they are delayed until the synchronization of the accessed page has been completed. The delay time is up to no more than half the time required for a write access (about 0.3ms), i.e. delaying the request does not lead to performance problems.

If the primary server fails, the secondary server takes over the role of the primary server and the vacant role of the secondary server is assigned to one of the remaining nodes of the cluster. Note that nodes may play several different roles at the same time, e.g. primary server for one set of slices and secondary server for another set.

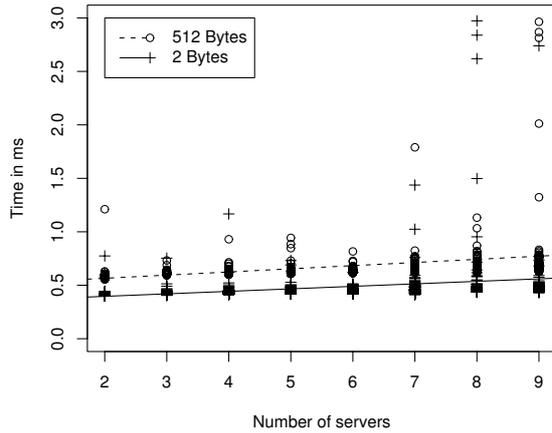


Figure 5: Writing data

2.4 Transactions

HADES has two modes of operation: with and without support for transactions. Using the mode without transactions, HADES can be used as a fast highly available replacement for hard disks. Using the mode with support for transactions HADES provides classical *ACID properties* (atomicity, consistency, isolation and durability (Gray, 1979)) to coordinate the simultaneous access of several clients or the support of reliable publish/subscribe. Transactions are controlled by an elected coordinator and backup coordinator. If a client crashes, transactions are automatically aborted.

HADES uses shadow paging (Lorie, 1977): changes are stored within separate memory regions and in case of a commit only a pointer needs to be switched. Since one page only contains one record, concurrency problems that result from storing several records in one block do not exist.

HADES supports distributed transactions spanning several clusters. This is realized by attaching *external* transactions to a normal transaction running on a different cluster thus forming one distributed transaction spanning several clusters. Control passes to the transaction to which an external transaction was attached. External transactions differ from normal transactions in that they cannot abort on their own in case of a failure of a client but only as part of the distributed transaction. This is required to prevent race conditions that only abort parts of the distributed transaction while others are committed if a client fails shortly after sending commit in the presumed commit two phase commit protocol used.

When an external transaction is attached to another transaction, the cluster containing the latter transaction becomes transaction coordinator for this distributed transaction and is responsible for either aborting or committing all parts of the transaction. The transaction coordinator is also responsible for initiating the abort of the distributed transaction in case of a failure of the client. Since HADES uses a complete cluster as transaction coordinator and the failure of one node does not affect the availability of the cluster as a whole, the use of two phase

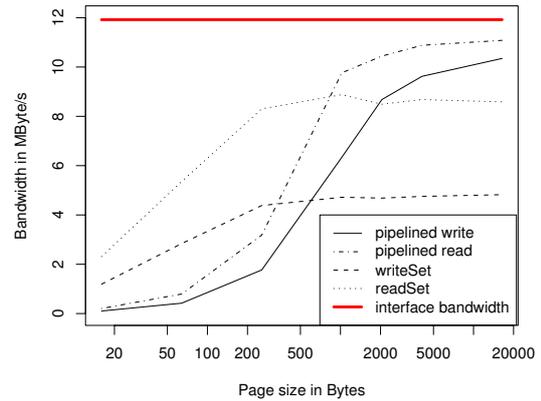


Figure 6: Bandwidth

commit (Gray, 1979) is sufficient and three phase commit (Skeen, 1981) is not needed since no blocking can occur.

External transactions are created and attached to another transaction using only one atomic step. Therefore, there is no risk of client failure between the creation and attaching of the external transaction.

2.5 Optimizations

Operations in HADES are not limited to read/write. More complex operations and operations that operate in parallel on whole slices can be added to reduce network traffic and improve speed. One good example is searching data: instead of reading all the data and searching on client side, the search request is transmitted to the cluster and performed locally.

3 Performance results

A cluster of 10 identical computers with AMD Athlon XP2000+ and 1 GBytes of main memory running HADES on top of Linux at user level was used as a test environment. The computers were connected using fast Ethernet.

Several performance measurements of different operations and different configurations were done. Due to space limitations we selected some representative results.

Write operation

This operation was performed using page sizes of 2 and 512 bytes and using a different number of servers. The result using 512 bytes can be directly compared to hard disk accesses. For each number of servers 100 write accesses were performed and the time of each single access was measured. HADES reaches access times of about $0.6ms$ (figure 5) and is therefore, about one order of magnitude faster than the access time of the fastest hard disks that currently reach about $5.4ms$ (IBM/Hitachi Ultrastar: $3.4ms$ seek time, $2ms$ latency).

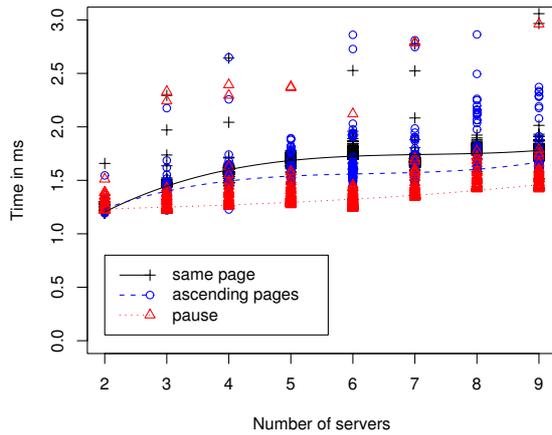


Figure 7: Transaction: read - modify - write

Bandwidth

Although not as important as the other operations for our target applications HADES supports optimizations of bandwidth: pages may be read and written in parallel (pipelined) or in sets of several pages. Figure 6 gives the results obtained using a single client using no transactions. Even for moderate page sizes of about 1024-2048 bytes the transfer speed is quite close to the maximum speed offered by the network interface.

Transactions

The scenario given in the introduction in which information is disseminated in reliable mode was evaluated. Within a transaction data is read, modified and written back. The transaction terminates with commit. Three cases were considered:

1. Always the same page is read and written. The next transaction, therefore, possibly needs to wait until all locks of the previous transaction are cleaned up and released.
2. A different page is read and written each time. Therefore, locks do not have any influence, but completing the cleanup of the preceding transaction may still consume some computing power.
3. A small pause is inserted between transactions, therefore, each transaction may use the full computing power.

The first two cases correspond to a system operating near full load, whereas the last one corresponds to a system that has some reserves left. HADES reached times around 1.5ms for the whole transaction (figure 7), i.e. the whole transaction can be completed faster than one third of the time required by one single access to hard disk.

Tablescan

Searching is a task that can be easily parallelized and therefore can utilize the parallel computing power of a cluster. This performance test used a table with 2.000.000 random entries (each of 48 bytes). A selection was performed that is equivalent to the following SQL-expression:

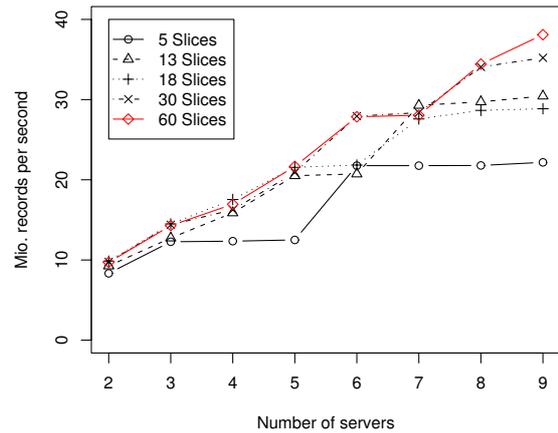


Figure 8: Tablescan performance

```
SELECT COUNT(*) FROM benchmark WHERE
```

```
Category='a';
```

The measurement was performed using several slice counts to see the effect of the number of slices. Figure 8 gives the obtained results. Overall HADES scales quite well as long as there are enough slices to evenly distribute the load. Uneven load distribution can be seen as levels in figure 8. These levels result from the server with the highest load dominating the required time. The following example will illustrate this effect: when distributing 6 slices to 4 or 5 nodes, in both cases at least one node will be assigned 2 slices whereas the others will be only assigned one slice. The node assigned 2 slices will dominate the performance and therefore there will be no performance gain when going from 4 to 5 nodes. Altogether, the measured performance numbers indicate that the optimal number of slices should be around 3-4 times the expected number of servers.

4 Comparison with other systems

There are several systems that *seem* to be suitable as a storage architecture for the applications presented or look similar to HADES, but they all lack at least one important property.

4.1 Conventional Database Management Systems

Conventional Database Management Systems (DBMSs) store data on hard disks. Therefore, these systems are limited by the disk access. Although these systems try to reduce the effect of disk accesses by caching, they cannot totally suppress slow disk access times. Write accesses cannot be safely cached, data is stored safely only after it has actually been written to disk or the log has been flushed to disk. Notably short transactions consisting of few writes suffer most from disk accesses, since they cannot take advantage of the high bandwidth of modern disks but mainly depend on the access time, which is only very slowly improving compared to bandwidth and density.

To demonstrate this effect we did some performance evaluations using PostgreSQL and BerkeleyDB. PostgreSQL needed about $100ms$ to insert a single data item and BerkeleyDB about $51ms$ (using a standard IDE hard disk). So both were about two orders of magnitude slower than HADES. We also measured the search performance. PostgreSQL reached about 0.9 Mio. scanned records per second and BerkeleyDB – since BerkeleyDB only supports searching for keys but not for data we had to implement the searching within the application – reached about 0.4 Mio. scanned records per second compared to at least 8 Mio. records per second that were achieved by a HADES cluster of two nodes. This is due to the fact that PostgreSQL and BerkeleyDB are designed to take into account that data is stored on disk whereas HADES can directly access data in main memory.

Even main memory DBMSs are affected by hard disk access times: although main memory databases like e.g. PRISMA/DB (Apers et al., 1992) store data in main memory, they use hard disks for logging just like other databases do. Pre-committing and group commit (DeWitt et al., 1984) allow to improve throughput, but they cannot improve commit latency. Therefore, also in this case hard disk access limits the response time of transactions.

4.2 RAID

Using RAID systems is the standard way of improving disk bandwidth (Patterson et al., 1988). However RAID systems cannot improve latency. Since small reads and writes are dominated by latency RAID systems cannot improve performance in our case.

4.3 Distributed checkpointing

If a node fails in a distributed system all intermediate results are lost and the (probably expensive) computation needs to be restarted. To avoid a restart from the very beginning some systems regularly take checkpoints (snapshots) of the distributed system state and are therefore able to continue the computation starting from the last complete checkpoint. Similar to HADES, RDSM (Ker-marrec, 1997) and (Plank and Li, 1994) keep all information in main memory. Different from HADES these systems can only guarantee persistence as soon as a new checkpoint has been taken and all modifications since then are lost. To reach the same degree of persistence of HADES using these systems a checkpoint would have to be taken as part of each write access, which would be prohibitively expensive.

4.4 Distributed hash table

Distributed hash tables (DHTs) try to redundantly store and locate information in a WAN using peer to peer techniques. Examples are Chord (Stoica et al., 2001), Pastry (Rowstron and Druschel, 2001), Tapestry (Zhao et al., 2001) and CAN (Ratnasamy et al., 2001). While both

HADES and DHTs store data in a network of computers this is already where the similarities end, since the design goals were totally different. Chord, for example, was designed with scalability in a WAN in mind whereas HADES was tuned for access time in a LAN. HADES requires $O(1)$ messages to access data compared to $O(\log N)$ messages required by Chord. Chord uses a probabilistic approach for load balancing. While this works on a large scale it may lead to a very uneven distribution on a smaller scale. Therefore, HADES uses explicit load balancing, which can guarantee an even distribution of data. HADES supports both atomic insert/update and transactions none of which is supported by Chord. So although HADES and Chord might look similar at first sight, they have very different properties and fields of application. The same applies for other DHT systems.

Another example is the LH*m scheme (Litwin and Neimat, 1996). It extends linear hashing to a scalable distributed data structure: each bucket is stored on one computer per site. Two sites are used, one being the backup of the other. For load balancing also the clients are divided into two groups having computers from one site as primary servers and computers from the other site as secondary servers. Since LH*m does not support atomic insert/update and transactions it has problems regarding consistency: if the same data record gets written at the same time by clients having the primary server in different sites, the messages for copying the data record from one site to the other may cross each other and after this different versions of the same object exist in the different sites. As a result, reading the same record results in different data, depending on which site the client belongs to. This is not acceptable for applications we are considering.

4.5 Distributed write cache

LND (Mao et al., 2002) implements a distributed fault tolerant write cache to reliably cache data to be able to speed up operation by asynchronously writing data to disk. This approach does not support transactions and is limited to one client which represents a single point of failure. If the client fails a recovery phase is needed, therefore, unlike HADES, which can serve several clients in parallel, LND is not able to provide uninterrupted service.

5 Summary and outlook

HADES offers significant improvements regarding access time and is, therefore, in particular useful for reliable publish/subscribe systems, publish/subscribe systems that aggregate events and therefore must store state, and systems using IP take-over.

Further speed improvements can be expected by moving from Ethernet and TCP/IP to network technologies, that were designed to reach low latencies like SCI,

Myrinet or Infiniband.

Although no real-time operating system and network was used during the performance evaluation, the performance measures show an astonishingly low number of outliers (figure 5). Therefore, a tuning of HADES towards real-time databases seems promising.

But the mechanisms used by HADES are also very interesting for a very different field of applications: sensor networks (Akyildiz et al., 2002). Sensor networks follow the idea of having very small, battery powered sensor nodes, that are so cheap, that they even might be deployed by throwing them from an aircraft in large quantities. Since these devices are battery powered, energy consumption is critical and determines the lifetime of the nodes in the network. Typically most energy is used by routing information through the network, therefore, data aggregation and data fusion are among the most important methods of saving energy. To do this, data needs to be stored until aggregated within the network, but the longer data is stored in the network the higher is the risk of data loss. Furthermore, aggregated data is 'more valuable' than raw data and should be protected according to its value against loss. Therefore, a reliable local storage is needed. Persistent storage like e.g. flash memory is not useful, since a node failure is equivalent to a node loss in sensor networks. Therefore, redundancy is the only way of protecting data against loss.

Of course there needs to be some tuning: HADES was optimized to achieve high speed, whereas in sensor networks the optimization must be done with regard to minimal power and hardware requirements, e.g. by adjusting HADES to use simpler network protocols than TCP and/or by replacing parallel, interleaved operations by serial operations that require less memory.

Another potential application of HADES is for support of mobility in publish/subscribe systems, where events and event histories must be made persistent (Cilia et al., 2003).

References

- Akyildiz, I. F., Su, W., Sankarasubramanian, Y., and Cayirci, E. (2002). A Survey on Sensor Networks. *IEEE Comm. Magazine*, pages 102–114.
- Apers, P. M. G., van den Berg, C. A., Flokstra, J., Grefen, P. W. P. J., Kersten, M. L., and Wilschut, A. N. (1992). PRISMA/DB: A parallel main memory relational DBMS. *Knowledge and Data Engineering*, 4(6):541–554.
- Carzaniga, A., Rosenblum, D. S., and Wolf, A. L. (2000). Achieving scalability and expressiveness in an Internet-scale event notification service. In *Symposium on Principles of Distributed Computing*, pages 219–227.
- Cilia, M., Fiege, L., Haul, C., Zeidler, A., and Buchmann, A. P. (2003). Looking into the past: enhancing mobile publish/subscribe middleware. In *Proceedings of the 2nd international workshop on Distributed event-based systems*, pages 1–8. ACM Press.
- DeWitt, D. J., Katz, R. H., Olken, F., Shapiro, L. D., Stoberaker, M. R., and Wood, D. (1984). Implementation Techniques for Main Memory Database Systems. In *Proc. of the ACM SIGMOD Conference*, pages 1–8.
- Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131.
- Fetzer, C., Högstedt, K., and Suri, N. (2003). Practical Aspects of Designing an IP Take Over Mechanism. In *Proc. of WORDS*.
- Gray, J. (1979). Notes on Data Base Operating Systems. In *Operating Systems: An Advanced Course*, pages 393–481. Springer-Verlag.
- Gärtner, F. C. (1999). Fundamentals of fault-tolerant distributed computing in asynchronous environments. *ACM Comput. Surv.*, 31(1):1–26.
- Kermarrec, A.-M. (1997). Replication For Efficiency And Fault Tolerance In A DSM System. In *Proc. of PDCS'97 Ninth International Conference on Parallel and Distributed Computing and Systems*.
- Litwin, W. and Neimat, M.-A. (1996). High-Availability LH* Schemes with Mirroring. In *Conf. on Cooperative Information Systems*, pages 196–205.
- Lorie, R. A. (1977). Physical integrity in a large segmented database. *ACM Trans. Database Syst.*, 2(1):91–104.
- Mao, Y., Zhang, Y., Wang, D., and Zheng, W. (2002). LND: a reliable multi-tier storage device in NOW. *SIGOPS Oper. Syst. Rev.*, 36(1):70–80.
- Patterson, D. A., Gibson, G., and Katz, R. H. (1988). A case for redundant arrays of inexpensive disks (RAID). In *Proc. of the 1988 ACM SIGMOD international conf. on Management of data*, pages 109–116. ACM Press.
- Plank, J. S. and Li, K. (1994). Faster Checkpointing with N+1 Parity. In *24th Annual int. symposium on Fault-Tolerant Computing*, pages 288–297.
- Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Schenker, S. (2001). A scalable content-addressable network. In *Proc. of the 2001 conf. on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM Press.
- Rowstron, A. I. T. and Druschel, P. (2001). Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Proc. of the IFIP/ACM Intl. Conference on Distributed Systems Platforms Heidelberg*, pages 329–350. Springer-Verlag.
- Skeen, D. (1981). Nonblocking commit protocols. In *Proc. of the 1981 ACM SIGMOD international conf. on Management of data*, pages 133–142. ACM Press.
- Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM*, pages 149–160. ACM Press.
- Zhao, B. Y., Kubiawicz, J. D., and Joseph, A. D. (2001). Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Technical Report UCB/CSD-01-1141, UC Berkeley.

ALGORITHMIC DIFFERENTIATION OF DIFFERENT ALGORITHMS FOR THE SAME PROBLEM: A CASE STUDY*

Christian H. Bischof
H. Martin Bucker
Arno Rasch
Emil Slusanschi
Institute for Scientific Computing
RWTH Aachen University
D-52056 Aachen, Germany

KEYWORDS

Algorithmic/Automatic Differentiation, Sensitivity Analysis, Numerical Simulation, FLUENT, SEPRAN

ABSTRACT

Large-scale numerical simulation is one of the basic blocks in computational science and engineering. Researchers in academia and industry rely on highly complex simulation codes that are able to simulate even the most complicated physical phenomena. Several approaches are possible in order to validate the results of a numerical simulation, such as examining the results obtained from different simulation packages. In addition one could analyze the impact of certain input parameters on the solution. Such sensitivities of the computed solution can be obtained by automatic differentiation, a technique for computing truncation error-free derivatives of functions given in the form of a computer program. For a standard flow problem, we examine the results obtained from the two simulation packages FLUENT and SEPRAN, and we compare the derivatives which are computed by automatic differentiation. We show that, although the two packages employ fundamentally different algorithms, the results obtained, i.e., the velocity fields and their respective sensitivities, are comparable.

INTRODUCTION

How can you be sure that your computer simulation is correctly modeling your scientific or engineering problem? To what extent does the output of your computer simulation depend on the actual algorithm used to solve your problem? It is often difficult to obtain a consensus about the answers to such questions. A comparison of the same computational problem by different numerical simulations, probably implemented by different computer codes developed by different authors, is one way

to give more insight into these issues. Such benchmarking of different codes, however, should be approached with care (Roache 1998). Another option to assess the behavior of a computer code is to consider the sensitivities of the output of a computer simulation with respect to its inputs. These sensitivities can be evaluated without truncation error by automatic differentiation. The idea behind this technique is to transform a given computer code into another code capable of evaluating the derivatives of selected output variables with respect to selected input variables. So, the algorithm used to implement a function, rather than the function itself, is differentiated. This is the reason why automatic differentiation (AD) is also called algorithmic differentiation. In the context of high-performance computing, where the underlying function is given by a large-scale simulation code, AD is often the only way to reliably evaluate derivatives.

Now, consider the scenario where AD is applied to different algorithms to solve the same computational problem. Here, it is not only interesting to study the differences in the original simulations but also to investigate how the AD-generated derivatives differ because they depend on the different algorithms used in the original simulations. Under certain circumstances, applying AD in a black box fashion can lead to surprising results because the automatic process is not only applied to the solution of the algorithm but also to the solution procedure itself (Eberhard and Bischof 1999; Bucker 2002).

To study the application of AD to different algorithms for the same problem, we consider a standard test problem in computational fluid dynamics (CFD), the flow over a backward-facing step, using the two multi-purpose simulation packages FLUENT (Fluent Inc. 1997) and SEPRAN (Segal 1993b). Both software packages are successfully used in academia and industry to simulate large-scale complicated scientific and engineering systems in various application areas. FLUENT and SEPRAN are first used to compute the flow field in this test problem. Then we apply the AD system ADIFOR (Bischof et al. 1996) to FLUENT and

*This research is partially supported by the Deutsche Forschungsgemeinschaft (DFG) within SFB 540 "Model-based experimental analysis of kinetic phenomena in fluid multi-phase reactive systems," RWTH Aachen University, Germany.

SEPRAN to automatically generate two AD codes computing the derivatives of the velocity field with respect to the maximum inflow velocity, one of the input variables of the test problem.

This note is structured as follows. In the following section, some details on applying AD to FLUENT and SEPRAN are given. Thereafter, the specification of the standard test problem is described. Finally, the derivatives computed by the two AD codes are compared.

AUTOMATIC DIFFERENTIATION OF FLUENT AND SEPRAN

In the present study we use the simulation packages SEPRAN and FLUENT. SEPRAN is a general-purpose finite element package developed at “Ingenieursbureau SEPRAN” and Delft University of Technology. The package is employed in various scientific areas ranging from fluid dynamics to structural mechanics to electromagnetism (Bosch and Lasance 2000; Segal et al. 1998; van Keken et al. 1995) and consists of approximately 800,000 lines of mostly Fortran 77, including comments. In this project we use version 12/2002 of the SEPRAN simulation package.

FLUENT, developed by Fluent Inc., is one of the leading simulation packages for computational fluid dynamics (CFD). It uses a pressure-based finite volume method for incompressible and mildly compressible flows. In the present case study we use version 4.5.2 of the FLUENT code which is basically written in Fortran 77 with some additional Fortran 90 language elements that are mainly used for the dynamic memory management. The whole source code consists of more than 2 million lines Fortran, including comments. Furthermore there are approximately 50,000 lines of C code (including comments) primarily concerned with the graphical user interface and system calls. However, most of the C code is not mandatory for the computational part of FLUENT.

Automatic differentiation (AD) is a set of techniques for transforming a given computer program implementing some mathematical function into another program evaluating the original function and its derivatives for a given input. The AD technology is applicable whenever derivatives of functions given in the form of a high-level programming language, such as Fortran, C, C++, or MATLAB, are required. The key idea of AD is that every computer program, no matter how complicated, is a—potentially very long—sequence of elementary operations such as addition or multiplication, for which the derivatives are known. Then, the chain rule of differential calculus is applied repeatedly, combining these step-wise derivatives to yield the derivatives of the whole program. This mechanical process can be automated, and several AD tools are available for transforming a given code to the new *differentiated* code. Note, that in contrast to numerical differentiation, derivatives obtained by AD are free from trunca-

tion error. The reader is referred to (Griewank 2000; Rall 1981) for a detailed introduction to this field.

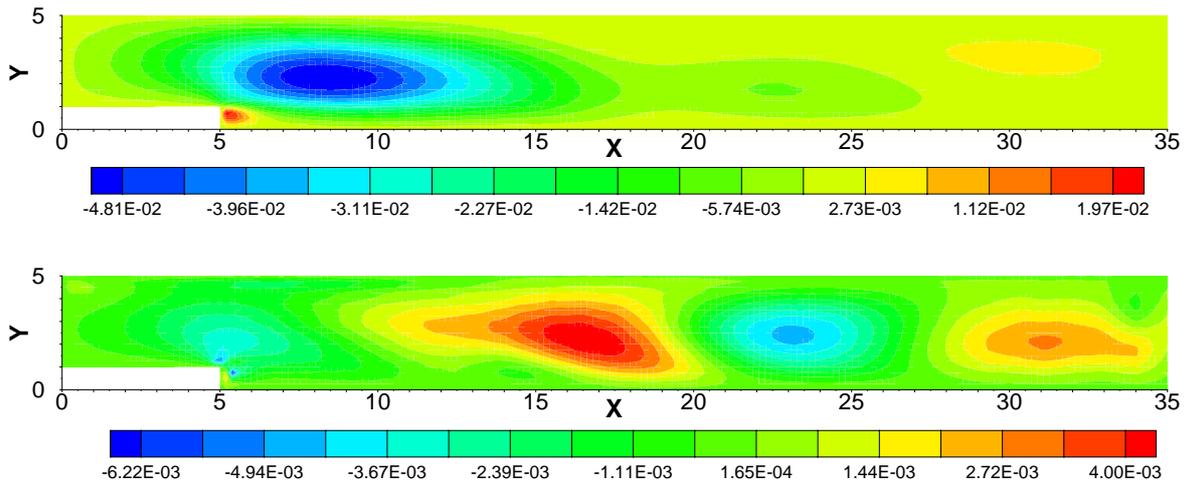
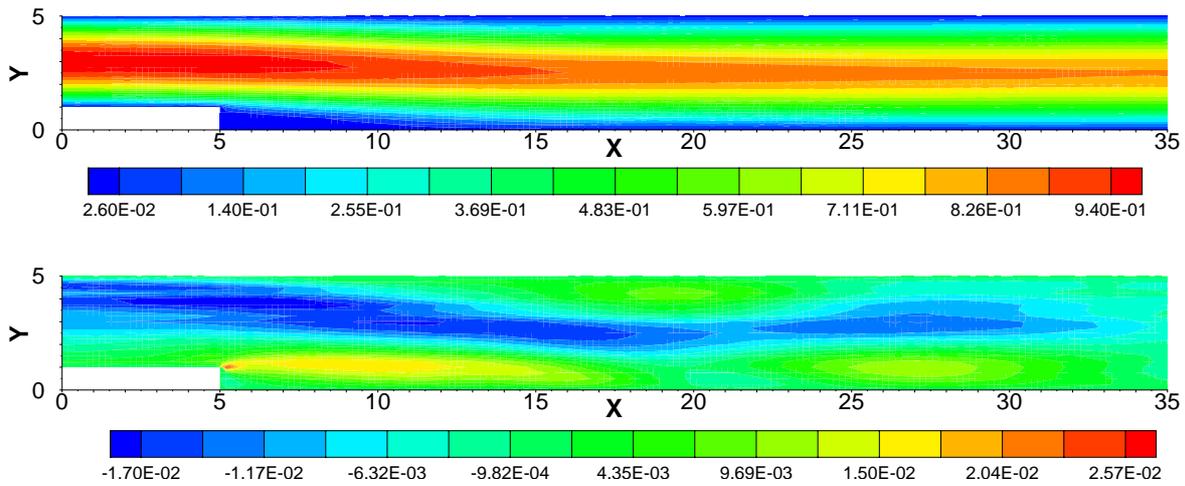
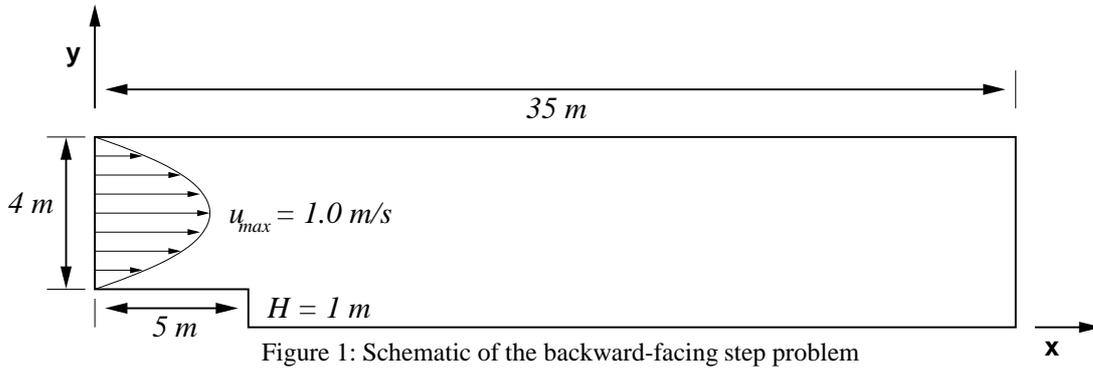
Although automatic differentiation is in principle applicable to computer codes of arbitrary size, practical experiences have shown that automatically differentiating large simulation packages consisting of several hundreds of thousands of lines of code is still a challenging task. Non-standard programming techniques used in legacy codes often require manual modifications before a software tool implementing AD could perform the actual transformation. A detailed description of the manual “code massaging” process is beyond the scope of this paper. The reader is referred to (Bischof et al. 2003; Bischof et al. 2001) for a detailed description of applying AD to the SEPRAN package. We used the AD tool ADIFOR (Bischof et al. 1996), developed at Argonne National Laboratory and Rice University, for generating differentiated versions of FLUENT and SEPRAN.

SIMULATING THE FLOW OVER A BACKWARD-FACING STEP

For comparing the differentiated versions of SEPRAN and FLUENT, we consider the simulation of an incompressible flow over a two-dimensional backward-facing step which is a common benchmark problem for CFD codes. A sample implementation of this problem is part of the SEPRAN distribution (Segal 1993a). The FLUENT tutorial guide (Fluent Inc. 1995) also contains a section on the backward-facing step problem.

In the following, we consider a two-dimensional rectangular domain of $35\text{ m} \times 5\text{ m}$ and a step height of $H = 1\text{ m}$ as depicted in Fig. 1. At the inlet, $x = 0$, a parabolic velocity profile with a maximum velocity $u_{\max} = 1.0\text{ m/s}$ is defined. The scalar input parameter u_{\max} specifies the complete parabolic velocity profile in horizontal direction. The vertical velocity component at the inlet is zero. Furthermore, we assume a fluid density of $\rho = 1.0\text{ kg/m}^3$ and a dynamic viscosity of $\eta = 1.0 \times 10^{-2}\text{ kg/(m s)}$. Based on the step height, the Reynolds number is 100. In addition to the flow field (u, v) , where u and v denote the horizontal and vertical velocity components respectively, we are interested in the derivatives of u and v w.r.t. the maximum inflow velocity, u_{\max} , i.e., $\partial u / \partial u_{\max}$ and $\partial v / \partial u_{\max}$.

Before comparing these derivatives which are to be computed by automatically differentiated versions of SEPRAN and FLUENT, we need to ensure that the values of (u, v) resulting from FLUENT and SEPRAN simulations are comparable. The structured computational grid used for the FLUENT simulation consists of approximately 3,000 rectangular cells whereas an unstructured mesh consisting of 5,000 nodes and 10,000 triangle elements is used for the SEPRAN simulation. In order to compare the results obtained from SEPRAN and FLUENT, we extract linearly interpolated (u, v) -values at 4,000 sample points arranged in



form of a regular grid covering the whole domain. All the plots presented in this note are based on this grid which enables the pointwise comparison of the results of the two simulations. Let u_F and v_F denote the interpolated values for u and v respectively, at the 4,000 sample points, based on the FLUENT solution. Similarly, u_S and v_S represent the corresponding values at the same points, where the SEPRAN solution is considered. In the upper plot of Fig. 2, the horizontal component of the velocity field, computed with FLUENT, u_F , is depicted. Since the corresponding results computed by SEPRAN look almost identical to the FLUENT results, the lower plot of Fig. 2 shows the difference of u computed by FLUENT, and u computed by SEPRAN. More precisely, the values of $u_F - u_S$ are depicted in the lower plot of Fig. 2. In Fig. 3 the vertical velocity component, v_F , and the difference of the vertical velocity components, $v_F - v_S$ are displayed. Note that for both components of the velocity vector, the differences of the two solutions found by SEPRAN and FLUENT are rather small, typically up to one order of magnitude, compared to the components of the velocity field. In comparison to the horizontal components, there are three separate clusters where the larger differences in the vertical components occur, located in the far flow field. However, from an engineering point of view, the recirculation area in the vicinity of the backward facing step is more interesting than the far flow field.

DERIVATIVES OF THE FLOW FIELD

Now we compute the derivative of the velocity field w.r.t. the maximum inflow velocity, i.e., $\partial u / \partial u_{\max}$ and $\partial v / \partial u_{\max}$ using the differentiated versions of FLUENT and SEPRAN. In order to compare the results of these two differentiated simulation codes we interpolate values at certain points within the computational domain, as described in the previous section. Let du_F and dv_F denote the derivative values $\partial u / \partial u_{\max}$ and $\partial v / \partial u_{\max}$, respectively, computed by the differentiated version of FLUENT. In an analogous manner du_S and dv_S denote the corresponding derivative values obtained by the differentiated version of SEPRAN. In Fig. 4 the derivative values du_F are depicted in the upper plot. With the naked eye, the derivative values du_S are almost indistinguishable from du_F . Similarly to the previous figures, the lower plot of Fig. 4 shows the difference, $du_F - du_S$ of the derivative values. Note that this difference is generally one order of magnitude below the derivative values. The derivatives of the vertical velocity component, computed by FLUENT and SEPRAN are compared in Fig. 5, where the upper and middle plot show the values for dv_F and dv_S , respectively, and the lower plot shows the difference, $dv_F - dv_S$. It turns out that dv_F and dv_S differ in three areas around $x \approx 18$, $x \approx 25$, and $x \approx 33$, which loosely correspond to the three clusters given in Fig. 3. In these clusters, the

differences, $dv_F - dv_S$, are in the same order of magnitude as dv_F and dv_S . In the remaining parts of the domain, dv_F and dv_S agree quite well. In particular, the extreme positive derivative values occurring behind the backward-facing step and the extreme negative values occurring in the area between $x \approx 10$ and $x \approx 15$ are computed similarly by the differentiated versions of both FLUENT and SEPRAN, so that the difference of dv_F and dv_S in those “interesting” parts is almost zero.

SUMMARY

A commonly used technique for “validating” a given computer simulation is to compare various simulation codes, implementing different methods for solving the underlying numerical problems. Another option is to observe the sensitivities of the solution with respect to certain input parameters of the simulation. Such derivatives can be obtained by automatic differentiation, a technique for reliable and accurate computation of derivatives of functions given by a computer program.

In this study, the comparison of the numerical computations using two different simulation packages, FLUENT and SEPRAN, is described by considering the standard benchmark problem of an incompressible flow over a backward-facing step. For the specified flow problem, both simulation codes produce similar solutions, i.e., the resulting velocity fields are almost identical. Furthermore we investigate the derivatives of the flow field with respect to the maximum inflow velocity, which is a free input parameter of the simulation. These derivatives, computed by automatically differentiated versions of FLUENT and SEPRAN, show quite a good agreement with each other, although the two simulation packages employ fundamentally different algorithms for solving the flow problem. This demonstrates that automatic differentiation is a valuable tool for sensitivity analysis which can be successfully applied on large-scale simulation codes. This work suggests the assessment of the quality and robustness of numerical simulation codes not only by comparison of the results, but also by comparison of the corresponding sensitivities. The methods presented in this work provide a way towards “validating” numerical simulation codes, thus paving the way to reliable modeling of engineering problems.

ACKNOWLEDGMENTS

This research is partially supported by the Deutsche Forschungsgemeinschaft (DFG) within SFB 540 “Model-based experimental analysis of kinetic phenomena in fluid multi-phase reactive systems,” RWTH Aachen University, Germany.

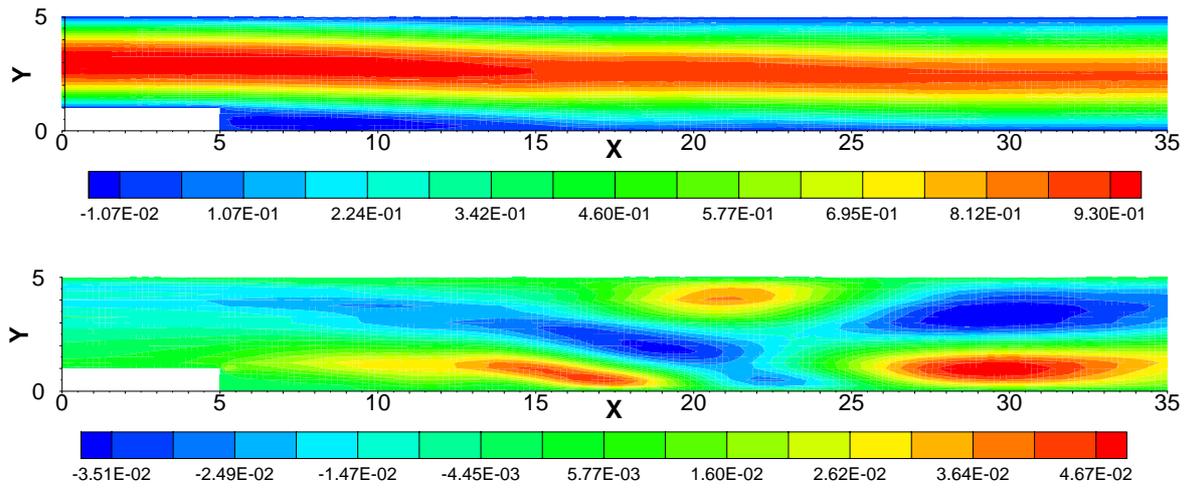


Figure 4: Derivative of u w.r.t. u_{\max} , computed with FLUENT, i.e., du_F (top), and difference between the derivatives of the horizontal velocity components, computed by FLUENT and SEPRAN, i.e., $du_F - du_S$ (bottom).

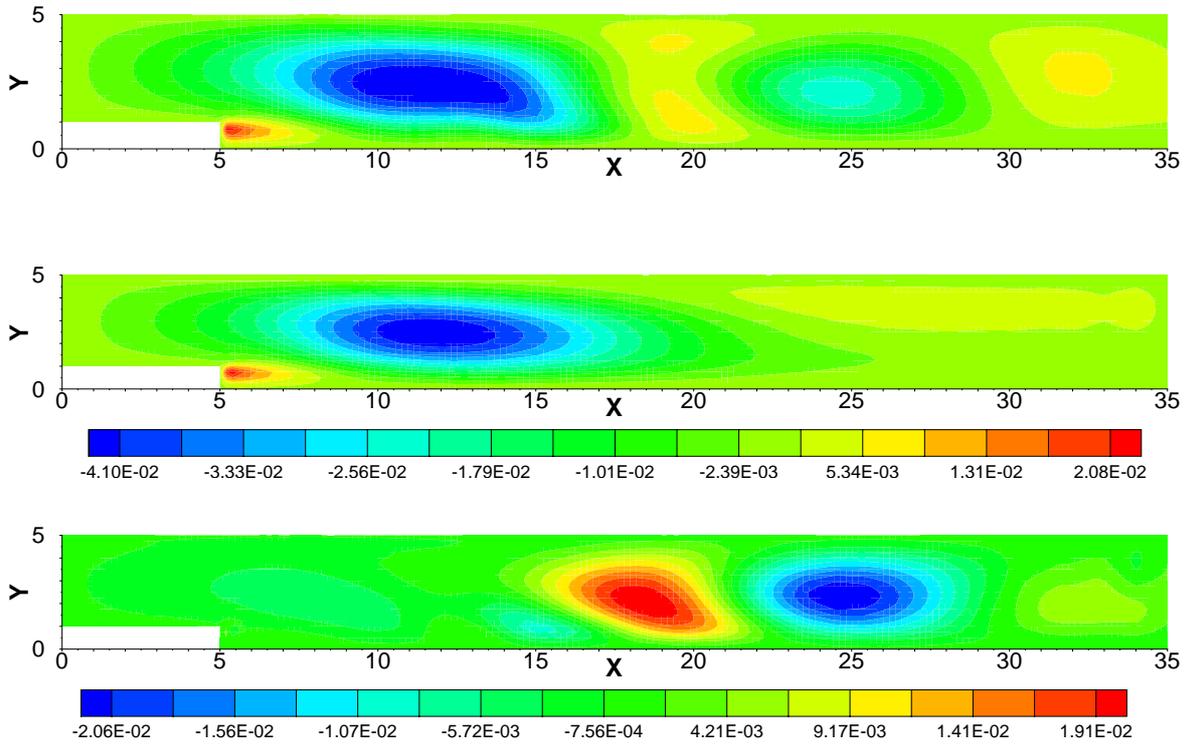


Figure 5: Comparison of the derivative of v w.r.t. u_{\max} computed by differentiated versions of FLUENT and SEPRAN: dv_F (top), dv_S (middle), $dv_F - dv_S$ (bottom).

REFERENCES

- Bischof C.; Carle A.; Khademi P.; and Mauer A. 1996. "ADIFOR 2.0: Automatic Differentiation of Fortran 77 Programs". *IEEE Computational Science & Engineering* 3, no. 3, 18–32.
- Bischof C.H.; Bücker H.M.; Lang B.; Rasch A.; and Risch J.W. 2001. "On the Use of a Differentiated Finite Element Package for Sensitivity Analysis". In V.N. Alexandrov; J.J. Dongarra; B.A. Juliano; R.S. Renner; and C.J.K. Tan (eds.), "Computational Science – ICCS 2001, Proceedings of the International Conference on Computational Science, San Francisco, USA, May 28–30, 2001. Part I", Springer, Berlin, vol. 2073 of *Lecture Notes in Computer Science*, 795–801.
- Bischof C.H.; Bücker H.M.; Lang B.; Rasch A.; and Risch J.W. 2003. "Extending the functionality of the general-purpose finite element package SEPRAN by automatic differentiation". *International Journal for Numerical Methods in Engineering* 58, no. 14, 2225–2238.
- Bosch E.G.T. and Lasance C.J.M. 2000. "High accuracy thermal interface resistance measurement using a transient method". *Electronics Cooling Magazine* 6, no. 3, 26–32.
- Bücker H.M. 2002. "Hierarchical Algorithms for Automatic Differentiation". Habilitationsschrift, Faculty of Mathematics, Computer Science, and Natural Sciences, Aachen University, Aachen.
- Eberhard P. and Bischof C.H. 1999. "Automatic Differentiation of Numerical Integration Algorithms". *Mathematics of Computation* 68, 717–731.
- FLUENT Tutorial Guide*. Fluent Inc., Lebanon, NH, USA.
- FLUENT 4.4 User's Guide*. Fluent Inc., Lebanon, NH, USA.
- Griewank A. 2000. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia.
- Rall L.B. 1981. *Automatic Differentiation: Techniques and Applications*, vol. 120 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin.
- Roache P.J. 1998. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, Albuquerque, New Mexico.
- Segal G. 1993a. *SEPRAN Standard Problems*. Ingenieursbureau Sepra, Leidschendam, NL.
- Segal G. 1993b. *SEPRAN Users Manual*. Ingenieursbureau Sepra, Leidschendam, NL.
- Segal G.; Vuik C.; and Vermolen F. 1998. "A Conserving Discretization for the Free Boundary in a Two-Dimensional Stefan Problem". *Journal of Computational Physics* 141, no. 1, 1–21.
- van Keken P.E.; Yuen D.A.; and Petzold L.R. 1995. "DASPK: a new high order and adaptive time-integration technique with applications to mantle convection with strongly temperature- and pressure-dependent rheology". *Geophysical & Astrophysical Fluid Dynamics* 80, 57–74.

AUTHOR BIOGRAPHIES

CHRISTIAN H. BISCHOF is head of the Institute for Scientific Computing and of the Center for Computing and Communication of Aachen University. He received a Ph.D. in computer science from Cornell University in 1988. His interests lie in semantic transformation, especially automatic differentiation, high-performance computing, virtual reality, and problem solving environments.

H. MARTIN BÜCKER received the Dipl.-Ing. degree in electrical engineering, the Dipl.-Inform. degree in computer science, and the Ph.D. degree in electrical engineering from RWTH Aachen University in 1993, 1994, and 1997, respectively. He worked for the Central Institute for Applied Mathematics, Research Center Jülich, from 1993 to 1998, participating in the design of parallel algorithms for sparse matrix problems. Since 1998 he has been with the Institute for Scientific Computing, RWTH Aachen University, where he received his Habilitation in 2003. His current areas of research include high-performance computing, numerical linear algebra, and automatic differentiation.

ARNO RASCH received his Dipl.-Inform. in computer science from RWTH Aachen University in 2000. He is currently working as research assistant and PhD student at the Institute for Scientific Computing, RWTH Aachen University. His research interests include parallel computing, automatic differentiation, and engineering applications.

EMIL-IOAN SLUSANSCHI received his Dipl.-Ing. in computer science from University Politehnica of Bucharest at the Faculty of Automatic Control and Computers in 2000. He also received his Advanced Studies Diploma at the same University in 2001. He is currently working as research assistant and PhD student at the Institute for Scientific Computing, RWTH Aachen University. His research interests include parallel computing and parallel architectures, automatic differentiation, and engineering applications.

THE CONNECT FRAMEWORK: A SIMULATION TOOL FOR NETWORKS OF COMMUNICATING OBJECTS

Gerd Kock

Fraunhofer Institute for Computer Architecture and Software Technology
Kekuléstr. 7, D 12489 Berlin, Germany
E-mail: gerd.kock@first.fraunhofer.de

KEYWORDS

Object Orientation, Modelling, Simulation, .NET

ABSTRACT

The CONNECT Framework presented in this paper is a simulation tool supporting the modelling of networks of communicating objects. It is implemented on the base of the .NET Technology. From the programmers point of view, it extends the C# language by a few attributes and offers a few classes. The paper introduces the new constructs and demonstrates their use by giving two examples. The first example is the Game of Life, and the second one, with special emphasis on signal based communication, is the Backpropagation Network.

INTRODUCTION

A modelling pattern found in areas like Artificial Neural Networks (Rojas 1996), Scientific Computing (Heath 2002), Swarm Intelligence (Kennedy and Eberhart 2001) and others requires working with arrays of communicating objects. Modelling large arrays of objects in a homogeneous manner is a non trivial problem. In some cases it is the index of an object that determines the processing (e.g. state-based manipulation or communication), in other cases all objects of an array are processed in a uniform way. A provision of special linguistic constructs for handling such arrays would allow for more elegant and error-proof modelling and programming of a whole range of different applications.

Modern software development usually takes place in some object oriented environment like the .NET Framework (Beer et al. 2003) or the Java 2 Enterprise Edition (Perrone et al. 2003). Considering the modelling pattern mentioned above in the context of such environments, we have to look at the concepts of classes and arrays, as any object is an instance of a class type, and as arrays are important means for establishing networks. The CONNECT Framework combines features from both of these concepts, and is an implementation for the .NET platform. The im-

plementation uses the fact, that C# supports reflection, and that it is one of the first languages, which can be extended with the help of so called attributes (Drayton et al. 2003).

For any given type T, a corresponding *layer type* Layer_T can be generated, and these layer types reflect features of class and array types. To give an example consider class Unit:

```
[unit]
public class Unit {
    [item]
    public double myVar;
    [item]
    public void MyFun(double x) {...}
}
```

Attribute [unit] turns the class into a *unit type*, where attribute [item] turns the field myVar into an *item field* and the method MyFun() into an *item method*. The interface of the generated *layer type* Layer_Unit looks as follows:

```
public class Layer_Unit : Layer {
    // constructor
    public Layer_Unit(params int[] dim);
    // indexer
    public Unit this [params int[] ix]
    { get; set; }
    // lifted field
    public Layer_Double myVar { get; }
    // lifted method
    public void MyFun(System.Double x);
}
```

With the aid of the constructor and the indexer layer objects can be instantiated and used similar to array objects. For example, let us create a two dimensional *base layer*:

```
Layer_Unit ulayer;
ulayer = new Layer_Unit(m,n);
```

The item field myVar (or item method myFun()) with index [i,j] can be accessed as follows:

```
ulayer[i,j].myVar = 3.14;
ulayer[i,j].MyFun(3.14);
```

```

[unit] public class Life {
    [item] public int state;
    [item] public Life[] vector;

    [item] public void InitState(Random r) { state = r.Next(2); }

    private int next;
    [item] public void NextState()
    {
        int sum = 0;
        for (int i = 0; i < vector.Length; i++) { sum += vector[i].state; }
        if (sum < 2 || sum > 3) { next = 0; }
        else if (sum == 3) { next = 1; }
        else { next = state; }
    }

    [item] public void UpdateState() { state = next; }
}

```

Figure 1: The unit type Life

But layer types are not identical to array types. If they are generated from unit types, the public fields and methods being marked by attribute `[item]` are “lifted” to the interface of the layer type. If, for a given layer instance, the lifted identifiers are used to access the layer, actually all corresponding fields or methods of the layer elements are addressed. In the case of methods, this is a kind of shorthand for a loop. Invoking a lifted method results in invoking the underlying unit method for all elements of the given layer:

```
uLayer.MyFun(3.14);
```

A lifted field is an *item layer* (e.g. `uLayer.myVar` has type `Layer_Double`), which opens up the possibility to invoke methods defined for layer types. For example, the overloaded method `Set()`, which can be used to initialize all layer elements with the same constant:

```
uLayer.myVar.Set(3.14);
```

Accessing lifted items can be combined neatly with the fact, that the underlying unit items can also be accessed by index. This is used by the CONNECT Framework to offer powerful means for connecting (or setting) the item fields of layers. The parameters of the corresponding methods are, on the one hand, lifted item fields, and on the other hand, relations or functions over indices.

The next section presents the CONNECT Framework components. After that, it is explained in detail how layer items can be connected, and the Game of Life is used to demonstrate this process. Then the focus is switched to signal based communication. Here, as an example, the Backpropagation Network is given. The concluding section

summarizes the lessons learned from the two examples and relates the CONNECT Framework to previous work of the author. At the end, a description of further research can be found.

CONNECT FRAMEWORK COMPONENTS

The CONNECT Framework consists of a library and a command line tool. Using the framework for the modelling of an application can be sketched as follows.

At first, the unit types of the application have to be identified. Let file `app.units` contain the source code of these types (class `Unit`, ...). Except for the attributes `[unit]` and `[item]` this is usual C# code.

Then, the command line tool can be used to generate all implied layer types (`Layer_Unit`, `Layer_Double`, ...), and to put them into file `app.layers`. The generation process relies on the types provided by the CONNECT library.

The rest of the application’s source code can use the unit types as well as the generated layer types.

The CONNECT library includes class `Layer` (the base class of all layer types), signal classes for simple types like `int`, `double` etc., and class `Generator`. Signal classes are explained below. Class `Generator` contains methods for generating layer or signal types. These methods are provided for the rare cases, where the command line tool or the predefined classes do not suffice.

CONNECTING LAYER ITEMS

For discussing the way, in which objects can be connected, let us start with unit type `Elem`:

```

[unit]
public class Elem {

```

```

public class LifeNetwork {
    private Layer_Life layer;

    public LifeNetwork (int n, int m) {
        layer = new Layer_Life(n, m);
        layer.vector.Connect(layer, new Relation(IsNeighbour));
    }

    public void InitPopulation() {
        Random r = new Random();
        layer.InitState(r);
    }

    public void NextGeneration() {
        layer.NextState();
        layer.UpdateState();
    }

    private bool IsNeighbour(int[] x, int[] y) {
        if ( (x[0]!=y[0] || x[1]!=y[1]) && ( Math.Abs(x[0]-y[0])<=1 ) && ( Math.Abs(x[1]-y[1])<=1 ) )
            return true;
        else
            return false;
    }
}

```

Figure 2: Using the layer type `Layer_Life`

```

[item] public Elem transpose;
[item] public Elem[] line;
}

```

Our goal is to organize `Elem` objects into a square matrix,

```
Layer_Elem matrix = new Layer_Elem(n,n);
```

such that the item layers `matrix.transpose` and `matrix.line` correspond to the transpose or lines, respectively, of that matrix. For a given line `i` and for all `j` and `k`, the one dimensional arrays `matrix[i,j].line` and `matrix[i,k].line` would contain the same objects.

To establish the requested connections, a function and a relation over indices are needed:

```

int[] transpose( int[] ix ) {
    int[] val = { ix[1], ix[0] };
    return val;
}

bool line( int[] ix, int[] iy ) {
    if (ix[0]==iy[0]) return true;
    return false;
}

```

The `CONNECT` Framework includes special types for such functions and relations:

```

Function fun = new Function(transpose);
Relation rel = new Relation(line);

```

Now, the layer methods `Set()` and `Connect()` can be applied:

```

matrix.transpose.Set(matrix, fun);
matrix.line.Connect(matrix, rel);

```

To explain the meaning of `Set()`, let `ix` be an index, for which `transpose(ix)` is defined. Then `matrix[ix].transpose` is set to `matrix[transpose(ix)]`.

To explain the meaning of `Connect()`, let `iy` be another index, and let the result of `line(ix,iy)` be `true`. Then `matrix[ix].line` is extended by one component, and this component is set to `matrix[iy]`.

THE GAME OF LIFE

The first example is the famous Game of Life (Gardner 1970). The complete modelling can be found in Figures 1 and 2. Figure 1 shows the definition of unit type `Life`, and Figure 2 shows, how the generated type `Layer_Life` can be used to define a class `LifeNetwork`.

A unit of type `Life` has two public item fields and three public item methods.

Item `state` is used to signal, whether a unit is alive (`state==1`) or dead (`state==0`), and the one dimensional array `vector` is used to get the corresponding states of the immediate neighbours — where item method `NextState()` assumes, that the `vector` has been set accordingly.

Method `InitState()` is used to randomly set the state, method `NextState()` checks, how many of the neighbours are alive, and uses the result to set the private variable `next`, and method `UpdateState()` updates the state.

In class `LifeNetwork`, a private layer of type `Layer_Life` can be found. The constructor creates a two dimensional layer, and establishes the necessary connections. In this, the relation `IsNeighbour()` is used. For indices `ix` and `iy` the result `IsNeighbour(ix, iy)` is `true` exactly in the case, that `ix` and `iy` are the indices of immediate neighbours.

The meaning of the methods of type `LifeNetwork` is straightforward: `InitPopulation()` initializes the network, and `NextGeneration()` computes the next generation.

SIGNAL BASED COMMUNICATION

In the example given above, a `Life` unit communicates by reading the `state` of neighbouring units with the help of item field `vector` (see Figure 1). In a sense, the field `vector` allows a `Life` unit to access the “outside world”.

However, it might be advantageous to communicate just by sending and receiving signals, without accessing the state of other units. To support this procedure, the `CONNECT` Framework offers the possibility to generate signal classes. More precisely, for any given type `T`, classes `Fanout_T`, `IVector_T`, and `OVector_T` can be generated. (For simple types like `double`, the signal classes are part of the `CONNECT` library.)

The signal classes represent *fanout signals*, *input* and *output vectors*. To give an example, let us start with three unit types based on the signal classes generated from type `double`:

```
[unit] public class FUnit {
    [item] public Fanout_Double y;
}

[unit] public class IUnit {
    [item] public IVector_Double ivec;
}

[unit] public class OUnit {
    [item] public OVector_Double ovec;
}
```

We consider the layers `flayer` (of type `Layer_FUnit`), `ilayer` (type `Layer_IUnit`) and `olayer` (type `Layer_OUnit`). At first, `n` fanout signals are created and initialized:

```
flayer = new Layer_FUnit(n);
for (int i = 0; i < n; i++)
    flayer[i].y.Value = i;
```

Then, `n` input and output vectors are created:

```
ilayer = new Layer_IUnit(n);
olayer = new Layer_OUnit(n);
```

Initially, these vectors are empty. However, input vectors can be connected with fanout signals and output vectors analogous to the way described above. Let `Full` be the relation, which includes all index pairs, i.e. for all pairs `(ix, iy)` we have:

```
Full(ix, iy) == true
```

This relation is used for establishing connections:

```
Relation full = new Relation(Full);
ilayer.ivec.Connect(flayer.y, full);
```

As all index pairs belong to the relation `Full`, the result of this connect statement is, that for all $0 \leq i, j < n$ the input vector `ilayer[i].ivec` is connected with fanout signal `flayer[j].y`. This means, the input vector is extended by one component, and this component is set to the fanout signal. After that, all input vectors have `n` components and, as the connections are established in ascending order, we have:

```
ilayer[i].ivec[j] == j
```

Modifying the fanout signals

```
for (int i = 0; i < n; i++)
    flayer[i].y.Value = -i;
```

has an immediate consequence:

```
ilayer[i].ivec[j] == -j
```

Now, the input vectors are connected to the output vectors:

```
ilayer.ivec.Connect(olayer.ovec, full);
```

Again, the connections are established in ascending order. For all $0 \leq i, j < n$ the input vector `ilayer[i].ivec` is connected with output vector `olayer[j].ovec`. Here, both vectors are extended by one component, and the new component of input vector `ilayer[i].ivec` refers to the new component of output vector `olayer[j].ovec`. This results in $n \cdot n$ components

for each input, and n components for each output vector.

By default, the new components of the output vectors are set to zero. So we have:

```
ilayer[i].ivec[n+j] == 0
```

If we assign other values to the output vectors,

```
olayer[i].ovec[j] = i*i;
```

we have:

```
ilayer[i].ivec[n+j] == j*j
```

THE BACKPROPAGATION NETWORK

The Backpropagation (BP) Network is capable of learning a functional mapping $x \mapsto y$. In this, network input $x = (x_1, \dots, x_i)$ and network output $y = (y_1, \dots, y_o)$ are vectors of numbers. The BP Network can be applied in areas such as sensor processing, pattern recognition, data analysis, and control (Rojas 1996).

Such a network consists of one input, one or more hidden, and one output layer. In the forward pass, the input layer is used to present the network input x to hidden layer 1. For $n \geq 1$, hidden layer n does some processing and presents its results to hidden layer $n+1$ or, finally, to the output layer. The output layer does an analogous processing to compute the network output y .

The output of a network depends on the weights, where each hidden and output unit is associated with one bias weight and one weight for each incoming signal.

For a given set of training data $\{(x, t)\}$ such a network is able to learn the mapping $x \mapsto t$ where $t = (t_1, \dots, t_o)$ is a target vector. Learning means to adapt the weights correspondingly. In general, learning a good mapping of the given targets is not the only goal of training. Another goal is that the network is able to “generalize”, i.e. after training it should be able to map an input x , which has not yet been seen so far, in a “sensible” way to an output y .

During training, the result $y = (y_1, \dots, y_o)$ of mapping a sample $x = (x_1, \dots, x_i)$ is compared with the associated target $t = (t_1, \dots, t_o)$. The backpropagation algorithm essentially is a gradient descent method minimizing the quadratic error measure $\sum_{i=1}^o (t_i - y_i)^2$ (seen as a function of the weights). Learning can be done either in online or in batch mode. Within online learning,

the weights are adjusted each time, a training example has been presented; within batch learning, weight adjustment takes place only after all training examples have been seen. Once a network is trained, a recall is done in one forward pass.

Now we consider, how the CONNECT Framework can be used to program a Backpropagation Network (implementing online learning). In Figure 3 the unit types `Input`, `Hidden`, and `Output` can be found, and Figure 4 contains the class `BPNetwork`, which is based on these types.

The only item of type `Input` is a fanout signal y , which is used to present the network input to hidden layer 1. The types `Hidden` and `Output` both are derived from class `BPUnit`, which contains all common elements:

- a fanout signal y used to present the result to the next layer or as network output, respectively;
- an input vector x for collecting the fanout signals of the preceding layer, and an associated bias b and weight vector w ;
- an output vector `outErr` used to send back error signals.

All layers (except for the input layer and hidden layer 1) send back error signals to the preceding layer. It is important, that error signal `outErr[i]` is send to the unit, where input `x[i]` stems from. Specifically, the size of error vector `outErr` either is 0 (for hidden layer 1) or has to coincide with the sizes of the vectors x and w .

The item methods `AdjustVector()` and `InitWeights()` are used during network initialization. For each incoming signal, a weight component is established, and all weights are initialized with random numbers in between 0 and 1.

Item method `Forward()` performs the computation, which is done in the forward pass. Actually, the so called net input $\text{net} = w * x + b$ is mapped by the sigmoid function. In this, $w * x$ is the vector dot product and the result of `Sigmoid(net)` is $1 / (1 + \exp(-\text{net}))$.

Item method `Backward()` is used for learning. The parameters are the so called learning rate `eta` and a value `delta`, which depends on the given target and is computed differently for hidden and output layers. Based on the given parameters, at first the error signals for the previous layer are set, and then the weights are modified.

```

[unit] public class Input {
    [item] public Fanout_Double y = new Fanout_Double();
}

public class BPUnit {
    [item] public Fanout_Double y = new Fanout_Double();

    [item] public IVector_Double x = new IVector_Double();
    [item] public double b;
    [item] public double[] w;
    [item] public OVector_Double outErr = new OVector_Double();

    [item] public void AdjustWeightVector() { w = new double[x.Length]; }
    [item] public void InitWeights( Random r ) {
        b = r.NextDouble();
        for (int i = 0; i < x.Length; i++ ) w[i] = r.NextDouble();
    }

    [item] public void Forward() { y.Value = NN.Sigmoid( w * x + b ); }

    public void Backward(double eta, double delta) {
        for (int i = 0; i < outErr.Length; i++ ) outErr[i] = delta * w[i];
        b = b + eta * delta;
        for (int i = 0; i < x.Length; i++ ) w[i] = w[i] + eta * delta * x[i];
    }
}

[unit] public class Hidden : BPUnit {
    [item] public IVector_Double inErr = new IVector_Double();
    [item] public void Backward(double eta) {
        double delta = y * (1-y) * inErr.Sum;
        base.Backward(eta, delta);
    }
}

[unit] public class Output : BPUnit {
    [item] public double t;
    [item] public void Backward(double eta) {
        double delta = y * (1-y) * (t-y);
        base.Backward(eta, delta);
    }
}

```

Figure 3: The unit types for the Backpropagation Network

In the derived types `Hidden` and `Output` the value `delta` is computed and passed to the `Backward()` method of type `BPUnit`. In one case (type `Output`) the value depends on the difference between target and network output ($t-y$), and in the other case (type `Hidden`) it depends on the sum of incoming error signals (`inErr.Sum`).

In Figure 4, a Backpropagation Network with two hidden layer is presented. At first, we find the private layer fields `input`, `hidden1`, `hidden2`, and `output`. The constructor creates a network with `i` input and `o` output units, establishes the necessary connections between the layers, and performs the corresponding adaptations of the weight vectors.

Method `InitWeights()` initializes the weights randomly, method `Forward()` implements the forward and method `Backward()` implements the

backward pass. Note the details of the `Forward()` and `Backward()` methods.

The parameter `x` of network method `Forward()` is the network input, which is assigned to the fanout layer `input.y`. Then, method `Forward()` is called for hidden layers 1 and 2, and finally for the output layer. At the end, the fanout layer `output.y` is returned as network result.

The parameters `eta` and `t` of network method `Backward()` are the learning rate or target vector, respectively. The target vector is assigned to the item layer `output.t`. Then, method `Backward()` is called for the output layer, and for hidden layers 2 and 1. At the end, function `sqdist()` computes the quadratic error, which is returned as the result.

```

public class BPNetwork {
    private Layer_Input input;
    private Layer_Hidden hidden1, hidden2;
    private Layer_Output output;

    public BPNetwork( int i, int h1, int h2, int o ) {
        input = new Layer_Input(i);
        hidden1 = new Layer_Hidden(h1);
        hidden2 = new Layer_Hidden(h2);
        output = new Layer_Output(o);

        hidden1.x.Connect( input.y, new Relation(Full) );
        hidden1.inErr.Connect( hidden2.outErr, new Relation(Full) );
        hidden2.x.Connect( hidden1.y, new Relation(Full) );
        hidden2.inErr.Connect( output.outErr, new Relation(Full) );
        output.x.Connect( hidden2.y, new Relation(Full) );

        hidden1.AdjustWeightVector();
        hidden2.AdjustWeightVector();
        output.AdjustWeightVector();
    }
    public void InitWeights() {
        Random r = new Random(1);
        hidden1.InitWeights(r); hidden2.InitWeights(r); output.InitWeights(r);
    }
    public double[] Forward( double[] x ) {
        double[] y = new double[output.Length];
        for (int i = 0; i < input.Length; i++ ) input.y[i].Value = x[i];
        hidden1.Forward(); hidden2.Forward(); output.Forward();
        for (int i = 0; i < output.Length; i++ ) y[i] = output.y[i];
        return y;
    }
    public double Backward( double eta, double[] t ) {
        for (int i = 0; i < output.Length; i++ ) output.t[i] = t[i];
        output.Backward(eta); hidden2.Backward(eta); hidden1.Backward(eta);
        return sqdist( output.t, output.y );
    }
}

```

Figure 4: The Backpropagation Network

CONCLUSIONS

The Game of Life (Gardner 1970) and the Backpropagation Network (Rojas 1996) have been used to demonstrate how the CONNECT Framework can be employed for modelling applications. The examples show that this can be done in a compact and elegant way, such that the resulting C# programs almost have the character of specifications.

Modelling a network of communicating objects can be done by considering a given application from two points of view. On the one hand, one can consider the functionality of the objects of that application, and in this can take the necessary communication structure as given. And on the other hand, one can take the functionality of the objects as given, and can concentrate on the global aspects of connecting objects and invoking complete layers of them.

The predecessor of the CONNECT Framework is the Neural Network (NN) description language

CONNECT, which had been developed in the nineties. A compiler translating CONNECT specifications into C++ classes is the software kernel of the NeuroLution system, which integrates hardware and software components (Kock et al. 1999).

The NN language CONNECT allows for flexible definitions of networks of simple processing units, each of them communicating with the others by sending simple signals, and can be applied to design neural networks, cellular automata as well as other simple distributed systems (Fabiunke and Kock 1999). But communication is restricted to simple signal types, and it is not possible to connect (or set) other than signal items.

The NN language CONNECT is a specific domain language, where the CONNECT Framework actually is a simple extension of the C# language by the two attributes [unit] and [item]. Using these attributes allows for solutions, the abstraction level of which is high and makes it simple to

reuse them. As the modelling language is C#, there is no restriction on the domain of possible applications.

FURTHER RESEARCH

The two examples given in this paper stem from the area of swarm intelligence (Kennedy and Eberhart 2001) or Artificial Neural Networks (Rojas 1996), respectively. For the latter domain it already has been proven, that many network models can be programmed easily by using the CONNECT Framework. The corresponding specifications written in the Neural Network description language CONNECT (Kock et al. 1999) can be translated easily into corresponding C# programs — which use the CONNECT Framework. Other possible application domains are Scientific Computing (Heath 2002) and Web Services (Alonso et al. 2004).

All domains mentioned above will be studied, and key applications will be developed to trigger further implementation and theoretical issues.

There are several implementation issues. One goal is to bring the current implementation into a product version state. Also, the use of threads and a distributed implementation will be considered. With respect to the generics of C# 2.0, a future version of the CONNECT Framework will include parameterized types `Layer<Type>`, `Fanout<Type>`, `IVector<Type>` and `OVector<Type>` — as an alternative for the `Generator` class mentioned above. Finally, an implementation for the Java platform will be considered.

Two kinds of “theoretical” issues occur. At first, the “final” form of the interface has to be fixed; this refers to the work out of layer methods for accessing and connecting layers, or to the question, whether beside `Connect()` methods there also should be `Disconnect()` methods, etc. Secondly, the meaning of networks consisting of interconnected layers mainly depends on the meaning of unit and layer types and on the meaning of layer methods, which globally access and (dis)connect layer items; it might be helpful to develop some mathematical means for treating the formal semantics of such networks.

AUTHOR BIOGRAPHY

GERD KOCK studied Mathematics and Economy (University of Münster). For a few years, he worked for Siemens in Munich. His doctoral thesis was in Computer Science (Technical University of Karlsruhe). Since 1992, he acts as se-

nior researcher at the Fraunhofer Institute for Computer Architecture and Software Technology (Fraunhofer FIRST) in Berlin. For many years, he worked in the areas of Programming Languages, Compilers and Software Technology. In cooperation with industry, he developed software concepts for a Neural Network simulation tool. He gave lectures and seminars at many places, e.g. at the Humboldt University of Berlin and at the Technical University of Berlin.

REFERENCES

- Alonso, G.; F. Casati; H. Kuno; and V. Machiraju. 2004. *Web Services. Concepts, Architectures and Applications*. Springer.
- Beer, W.; D. Birngruber; H. Mössenböck; and A. Wöß. 2003. *Die .NET-Technologie*. dpunkt.verlag.
- Bonabeau, E. and C. Meyer. 2001. “Swarm Intelligence: A Whole New Way to Think About Business”. *Harvard Business Review (May)*, 107–114.
- Drayton, P.; B. Albahari; and T. Neward. 2003. *C# in a Nutshell*. O’Reilly.
- Fabiunke, M. and G. Kock. 1999. “A Connectionist Method to Solve Job Shop Problems”. *Second International Conference on Intelligent Processing and Manufacturing of Materials (IPMM’99)*, Big Island, Hawaii, July 1999.
- Gardner, M. 1970. “Mathematical games: The fantastic combinations of John Conway’s new solitaire game “life””. *Scientific American*, 120–123.
- Heath, M.T. 2002. *Scientific Computing: An Introductory Survey*. McGraw-Hill.
- Kennedy, J. and R.C. Eberhart. 2001. *Swarm Intelligence*. Morgan Kaufmann Publishers.
- Kock, G.; T. Fischer; W. Eppler; H. Gemmeke; and T. Becher. 1999. “NeuroLution: Integrated Hardware and Software for the Development of Neural Network Applications”. *Systems Analysis – Modelling – Simulation* 35, No. 4, 447–481. Gordon And Breach Science Publishers.
- Perrone, P.; S.R. Venkata; and T. Schwenk. 2003. *J2EE Developers’s Handbook*. SAMS.
- Rojas, R. 1996, *Neural Networks - A Systematic Introduction*. Springer.
- Winder, R. and G. Roberts. 2000. *Developing Java Software, 2nd Edition*. Wiley.

SWARM Simulation of Multi-Agent Fault Mitigation in Large-Scale, Real-Time Embedded Systems

Derek Messie

Jae C. Oh

Department of Electrical Engineering and Computer Science

Syracuse University

Syracuse, NY 13244

Email: dsmessie@syr.edu, jcoh@ecs.syr.edu

KEYWORDS

SWARM, BTeV, Emergent Behavior, Subsumption Architecture, Fault Mitigation

ABSTRACT

This paper presents a SWARM multi-agent simulation of fault mitigation within BTeV, a large-scale, real-time embedded system. The Real-Time Embedded Systems (RTES) group collaborates on designing real-time embedded intelligent software to ensure data integrity and fault tolerance within BTeV, a triggering and data acquisition system for particle-accelerator-based High Energy Physics experiments at Fermi National Laboratory. The hardware layout spans 2,500 digital signal processors and approximately 2,500 Linux computers. Adaptive and small in footprint, very lightweight agents were designed to apply both reactive and proactive rules to accomplish fault tolerance within the system. The scale and real-time requirements of this system make it ineffective to design a traditional expert system that relies on centralized processing to determine appropriate rule actions for every possible system state. Instead, a decentralized approach based on Rodney Brooks' *subsumption architecture* is presented. A SWARM simulation is used to investigate emergent behavior of the multi-layered, distributed approach to fault mitigation within the RTES/BTeV environment.

INTRODUCTION

This paper describes the design and implementation of a SWARM simulation of multi-agent fault mitigation within BTeV, a large-scale, real-time embedded system. BTeV is a particle accelerator-based High Energy Physics (HEP) experiment studying matter-antimatter asymmetries in the decays of particles containing the bottom quark. The Real-Time Embedded Systems Collaboration (RTES) was formed with the purpose of designing real-time embedded intelligent software to ensure data integrity and fault-tolerance within the BTeV data acquisition system. Multiple

levels of adaptive Very Lightweight Agents (VLAs) are one of the primary components responsible for fault mitigation within this environment. The VLA design was implemented and presented in a prototype of the RTES/BTeV system at the SuperComputing 2003 (SC2003) conference.

Given the number of components and countless fault scenarios involved, it would be impossible to organize various levels of VLAs using an 'expert system' that applies mitigative actions triggered from a central processing unit acting on rules capturing every possible system state. Rather, this project uses Brooks' (Brooks 1986) multi-layer, decentralized *subsumption architecture* from mobile robot design, and adapts it to achieve specific global behavior within the BTeV fault mitigation system. The SWARM simulation is used for studying the scaling qualities of emergent behavior resulting from the subsumption approach.

This paper is divided into six sections. Section 2 provides some background about the BTeV experiment and the RTES collaboration. Section 3 presents a brief overview of Brooks' *subsumption architecture*, and describes the details of the subsumption model for multi-agent fault mitigation within RTES/BTeV.

Section 4 describes the SWARM multi-agent simulation of the RTES/BTeV environment, including details on the system components and fault scenarios modeled, VLA subsumption fault mitigation rule firing, and simulation time steps.

Next steps for the project are provided in section 5, followed by a conclusion in section 6.

RTES/BTeV

Overview

BTeV is a particle accelerator-based High Energy Physics (HEP) experiment currently under development at Fermi National Accelerator Laboratory. The goal is to study charge-particle violation, mixing, and rare decays of particles known as beauty and charm hadrons, in order to learn more about matter-antimatter asymmetries that exist in the universe to-

The BTeV Spectrometer

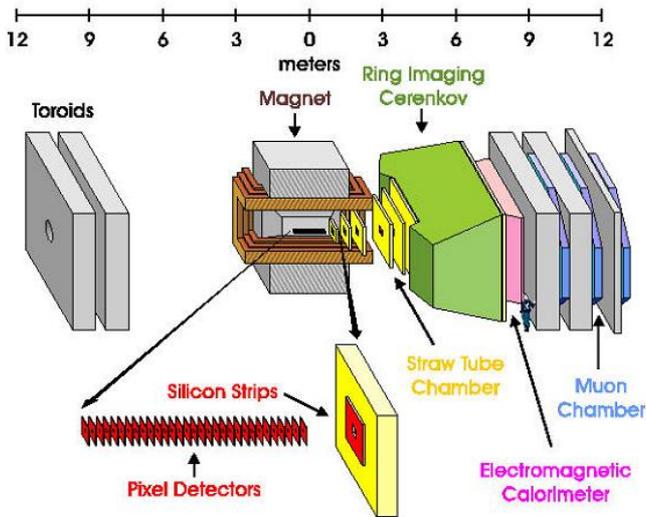


Figure 1: BTeV pixel detector layout.

day (Kwan 2002). The project is sponsored by the National Science Foundation.

The BTeV experiment will exist inside a particle accelerator where the collision of protons with antiprotons can be recorded and examined for detached secondary vertices from charm and beauty hadron decays. The layout for the BTeV detector is shown in figure 1.

The experiment uses 30 planar silicon pixel detectors to record interactions between colliding protons and antiprotons in the presence of a large magnetic field. A schematic view of the pixel detector is shown in figure 2. These detectors, along with readout sensors are embedded in the accelerator, which are connected to specialized field-programmable gate arrays (FPGAs). The FPGAs are connected to approximately 2,500 digital signal processors (DSPs).

The interactions resulting from the collision of protons and antiprotons are carried via custom circuitry hardware to localized processors that reconstruct the 3-dimensional crossing data from the 30 silicon pixel detectors in order to examine the trajectories for detached secondary vertices (Nordstrom 2003). These detached vertices are indicators of the likely presence of beauty or charm decays.

BTeV will operate at a luminosity of $2 \times 10^{32} \text{cm}^{-2} \text{s}^{-1}$ corresponding to about 2 interactions per 7.6 MHz beam crossing rate (Kwan 2002). Average event sizes will be around 200 Kilobytes after zero-suppression of data is performed on-the-fly by front-end detector electronics. Every beam crossing will be processed, which translates into the extremely high data rate of approximately 1.5 Terabytes of data every second, from a total of 20×10^6 data channels.

A three tier hierarchical trigger architecture will be used to handle this high rate. Data from the pixel detector and muon detector will be sent to the level 1

trigger processor, where an accept or reject decision will be made. The level 1 vertex trigger processor will perform pattern recognition, track, and vertex reconstruction on the pixel data for every interaction (Kwan 2002). It has been estimated that 99% of all minimum-bias events will be rejected by the level 1 vertex trigger, while 60-70% of the events containing beauty or charm decay will still be accepted for further evaluation.

Level 2 and 3 will be implemented on a cluster of CPU nodes, and data that makes it past the level 1 filter will be assigned to one of these level 2/3 processors for further analysis. Data that survives level 2 will be passed to level 3 algorithms to determine whether or not it should be recorded on archival media (Butler 2002). It is estimated that level 2 will decrease the data rate by a factor of 10, and level 3 will further reduce the incoming rate by a factor of 2. Once data is filtered through all three levels, and additional data compression is performed, it is expected that the resulting data rate will be approximately 200 Megabytes per second.

The events that are actually accepted within this system occur very infrequently, and the cost of operating this environment is high. The extremely large streams of data resulting from the BTeV environment must be processed real-time with highly resilient adaptive fault tolerant systems (Butler 2002). For these reasons, a Real-Time Embedded Systems Collaboration (RTES) was formed with the purpose of designing real-time embedded intelligent software to ensure data integrity and fault-tolerance within this data acquisition system.

Each of the 2500 DSPs are assigned a unique Very Lightweight Agent (VLA) responsible for a specific set of proactive and reactive fault mitigation rules. Figure 3 shows one such VLA, along with the two other components (Local Manager, Physics Application(PA)) found at every DSP.

The group of components found at each DSP are referred to as individual *Workers*. Multiple Workers are grouped into a single *Farmlet* of nodes. Similarly, proceeding up the chain, regional managers

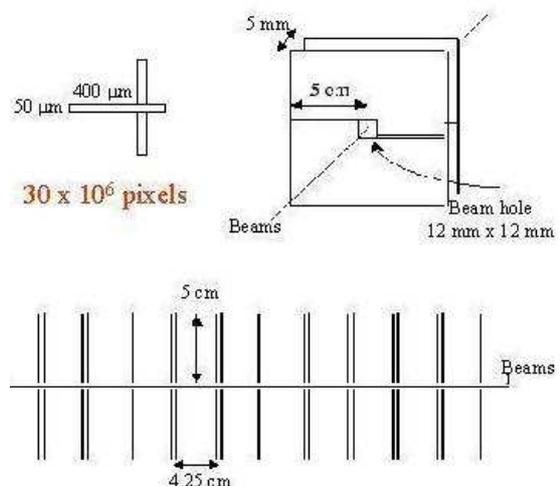


Figure 2: Schematic drawing of the pixel detector.

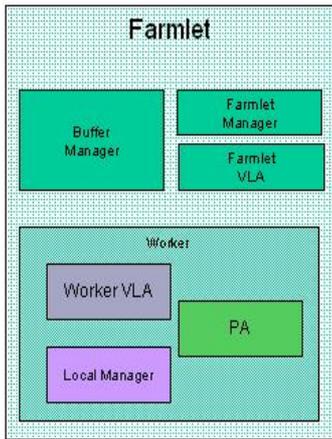


Figure 3: BTeV Level 1 Worker

are assigned groups of Farmlets. Individual Farmlet VLAs are capable of communicating with each of the associated Worker VLAs under them. Likewise, regional VLAs communicate directly with all Farmlet VLAs within their region.

Very Lightweight Agents (VLAs)

Multiple levels of very lightweight agents (VLAs) as described in (Tamhankar et al. 2003) are one of the primary components responsible for fault mitigation across the BTeV data acquisition system.

The primary objective of the VLA is to provide the BTeV environment with a lightweight, adaptive layer of fault mitigation. One of the latest phases of work at Syracuse University has involved implementing individual proactive and reactive rules for specific system failure scenarios.

A scaled prototype of the level 1 RTES/BTeV environment was presented at the SuperComputing 2003 (SC2003) conference. The prototype hardware consisted of a 7-slot VME crate with 4 fully populated motherboards and 16 DSPs. The DSPs were Texas Instruments C6711 with 64MB of RAM each, running at 166 MHz. Graphical Modeling Environment (GME) software was used to model the RTES/BTeV data acquisition system (Bapty et al. 2003). GME was developed by the Institute for Software Integrated Systems (ISIS) at Vanderbilt University as part of research investigating core technology for model-integrated computing, and applications for software integrated systems.

Reactive and proactive VLA rules were integrated within this level 1 prototype and served a primary role in demonstrating the embedded fault tolerant capabilities of the system.

Challenges

While the SC2003 prototype was effective for demonstrating the real-time fault mitigation capabilities of VLAs on limited hardware utilizing 16 DSPs, one of the major challenges is to find out how the behavior of the various levels of VLAs will scale when implemented

across the 2500 DSPs projected for BTeV. In particular, how will agent rules within each VLA interact as they are activated in parallel at multiple layers of the system, and how will this affect other components and the overall behavior of a large-scale, real-time, embedded system such as BTeV.

Given the number of components and countless fault scenarios involved, it would be impossible to design an ‘expert system’ that applies mitigative actions triggered from a central processing unit acting on rules capturing every possible system state. Rather, one alternative is to use Rodney Brooks’ multi-layer, decentralized subsumption approach for mobile robot design, and adapt it to achieve specific layers of global behavior within large-scale fault mitigation systems.

SUBSUMPTION MODEL FOR MULTI-AGENT FAULT MITIGATION

Layers of Subsumption

The phrase *subsumption architecture* was first used by Brooks (Brooks 1986) to describe a bottom-up approach for mobile robot design that relies on multiple layers of distributed sensors for determining actions. Until that time, designs relied heavily on a centralized location where most, if not all, of the decision making process took place. In fact, only initial sensor perception and motor control were left to distributed components. As a result, the success and adaptability of these systems was almost entirely dependent on the accuracy of the model and actions represented within the central processing unit (Brooks 1991).

In contrast, Brooks proposed that there should be essentially no central control. Rather, there should be independent layers each made up of a large number of sensors, with each layer responsible for distinct behavior. Communication and representation is developed in the form of action and inaction at each of the individual layers, with certain layers subsuming other layers when necessary. In this way, layer after layer is added to achieve what Brooks refers to as *increasing levels of competence*. This in turn breaks the problem down into what Brooks describes as ‘desired external manifestations’, as opposed to slicing the problem on the basis of ‘internal workings’ of the solution as was typically done in the past.

Figure 4 shows the *levels of competence* that Brooks defined for his mobile robots. In order to implement the subsumption design across a multi-agent fault mitigation system, the first step is to clearly define the distinct layers of subsumption that may be required to achieve the effective fault mitigation behavior desired. Just as Brooks’ provided, a description that distinguishes each layer needs to be outlined, along with detail on the specific roles and responsibilities of each.

Figure 5 defines these layers for the system fault mitigation implementation presented in this paper.

0. *Avoid contact with objects (moving or stationary).*
1. *Wander aimlessly around without hitting things.*
2. *"Explore" the world by seeing places in the distance which look reachable and heading for them.*
3. *Build a map of the environment and plan routes from one place to another.*
4. *Notice changes in the 'static' environment.*
5. *Reason about the world in terms of identifiable objects and perform tasks related to certain objects.*
6. *Formulate and execute plans which involve changing the state of the world in some desirable way.*
7. *Reason about the behavior of objects in the world and modify plans accordingly.*

Figure 4: Layers of competence for mobile robots as defined by Brooks.

The first layer (layer 0) is responsible for simple reactive rule activation within each component of the system. Just as layer 0 for mobile robots is responsible for the rudimentary task of avoiding objects, layer 0 here takes the most basic actions as defined in the local reactive rule base of each corresponding component. At this layer, incoming messages from connected components are typically reacted to by either sending simple remedial actions to problem components, or by forwarding error messages to higher level components within the system. The roles and responsibilities for each of the higher layers is also provided.

Implementation

In (Brooks 1986), layers 0 and 1 were implemented, and work had just begun on designing rules that would begin to address attributes of the levels of competence associated with layer 2. Similarly, the fault mitigation simulation described in this paper has implemented layers 0 and 1, along with a limited number of rules addressing layer 2.

The first step in this approach was to define individual VLA rules within the system that will contribute to meeting the responsibilities assigned to the lower layers of the subsumption architecture. Basic fault mitigation rules that address levels 0 and 1 define both reactive (level 0) and proactive (level 1) rules for the system. Ten of the error scenarios that are simulated in this implementation are listed in figure 6, and figure 7 lists the specific fault mitigation rules activated in each case.

Actions triggered from rules deemed to contribute to higher levels of competence are given priority at each individual component, and in this way subsume lower level rules when conflicts arise. At each layer, rules are experimented with and adjusted to compliment emerging behavior that contributes to meeting the fault mitigative responsibilities of each layer.

The general procedure for adding higher levels of competence is described by Brooks (Brooks 1986).

0. *Act on local reactive rules.*
1. *Monitor local components and activate local proactive rules.*
2. *Analyze and act on statistics gathered from rule firing.*
3. *Map the local state of the environment.*
4. *Reason about the system in terms of accumulated statistics and the state of connected components.*
5. *Formulate and execute actions which involve changing the state of the system in some desirable way.*
6. *Reason about the behavior of components in the system and modify plans accordingly.*

Figure 5: Layers of competence for fault mitigation within BTeV.

Layers of the control system that correspond to specific levels of competence are built, and new layers are added to the existing set in order to move to the next higher level of overall competence. A complete control system that accomplishes level 0 is first implemented. It is debugged thoroughly, and is not altered from that point on. Another separate layer of control is then built on top of the zeroth level, and is called control level 1. It is capable of accessing and examining data from the level 0 interface, essentially suppressing the normal data flow at level 0. This layer, along with the behavior of level 0, achieves level 1 competence. However, level 0 continues to run unaware of the layer above that may or may not have interfered with its data paths. This process is repeated to achieve higher and higher levels of competence.

SWARM

Overview

The primary focus of this simulation is on effective ways to implement the multi-agent subsumption approach within a complex system such as BTeV so that qualities of emergent fault mitigation behavior can be evaluated. A high volume of rules at various levels will be used in order to evaluate the interaction and behavior of the VLAs and other components across 2500 DSPs. This requires a simulation environment that will allow abstract representation of some of the complex integration within BTeV. However, for the results to be of any use at all, this abstraction must be done in a way that still accurately models the actual behavior of the components involved. The level of success of the results are directly dependent on this accuracy.

SWARM (<http://www.swarm.org>), distributed under the GNU General Public License, is software available as a Java or Objective-C development kit that allows for the multi-agent simulation of complex systems. It consists of a set of libraries that facilitate implementation of agent-based models.

The basic architecture of SWARM provides for the

ID	Description	Possible Causes
e1	DSP over time budget on crossing processing.	Crossing was too complex to complete and developer was not careful to give up in time.
e2	PA is stuck in a loop (within software timer control)	Improper error handling caused the program to get stuck in an infinite loop.
e3	DSP application framework is stuck in a loop (outside of software timer control)	Logic error in code that manipulates the board's communications facilities.
e4	DSP application branches to an illegal instruction.	Logic error any place in the code that causes corruption of memory.
e5	Processing times per crossing are too long.	SAF reported crossing processing times are consistently falling out of range.
e6	Too many track segments. Not necessarily a fault at the source.	The front-end hardware is malfunctioning, more particles collided than can be managed, bug in the upstream algorithms.
e7	Corrupt data in a crossing (truncated, misaligned, or bad header)	Bad checksum or incorrect header data in a crossing due to transmission failure or upstream logic error.
e8	Corrupt data - no such channels in the detector.	Logic error in the front-end electronics or firmware. (byte swapping)
e9	Crossing data lost.	DSP was reset or reboot while an event was being processed, FPGA input queue overflow, FPGA output queue overflow.
e10	Failed to transfer results down the DSP L1 buffer link. (buffer ready flag not set in time)	The level-1 buffers were not ready to receive data, the farmlet output queues overflowed.

Figure 6: 10 BTeV Error Scenarios

simulation of collections of concurrently interacting agents (Daniels 2000). It provides an environment that can model various components of the BTeV system, assigning dynamic states to each agent, which can then be altered in time steps following various user-specified rules (Burkhart 1997). Both proactive and reactive rules are triggered after the current state of a given agent(or component) is evaluated against the state of other connected agents(or components).

SWARM Simulation of RTES/BTeV

Since the VLA is the primary focus for evaluating and acting on a number of sample fault mitigation scenarios, VLAs across the system are simulated. Another main component of the simulation is the Physics Application (PA). Since it is central to many of the defined fault scenarios, it is vital that the state and behavior of the PA is simulated accurately. The PA is located at the Worker level, where the local manager and Worker VLA (WVLA) also reside. All three of these components are modeled. In addition, Farmlet managers and Farmlet VLAs (FVLAs), as well as corresponding regional managers and regional VLAs (RVLAs) are also simulated.

A sample screen shot of the RTES/BTeV SWARM Simulation is shown in Figure 8. This particular configuration has defined 6 Workers per Farmlet, 10 Farmlets per region, and groups consisting of 7 regions. This provides a simulation of 2,520 DSPs, just over the ac-

ID	How and where detected	Action
e1	Timer interrupt on DSP caught by VLA.	VLA resets PA. VLA sends m08 to FM VLA. VLA sends m29 if rate of these is too high.
e2	Many m08 messages over a time period containing PC within a region of code. Detected by Farmlet Manager VLA.	Send m26 to RM VLA. Notify OP.
e3	Many m06, m07 messages over a time period containing PC within a region of code. (FM, RM)	FM sends m26, RM notifies OP.
e4	m06, m07 message contains PC that is not in program space. (FM)	FM sends m26, RM notifies OP.
e5	Ensemble of m15 messages does not match known distribution.	DSP or FM send m14, RM notify OP.
e6	PA discovers this. (DSP)	PA sends m18, DSP/FM records and sends m18 if rate too high. RM notifies OP.
e7	FPGA cannot verify checksums or lengths, SAF cannot unpack header or sees bad status from FPGA (DSP)	DSP sends m16, FM records and sends m16 if rate too high, RM notifies OP.
e8	PA discovers this while unpacking the crossing data. (DSP)	DSP sends m17, FM records and sends m17 if rate too high, RM notifies OP.
e9	Discovered during the reset/reboot sequence using NVRAM history, crossing with only a header appears. (DSP)	DSP sends m11, FM records and sends m11 if rate too high, RM notifies OP.
e10	Timeout during transfer of algorithm results in SAF. (DSP)	DSP sends m10 and m11, FM records and sends m10 if rate too high, RM notifies OP.

Figure 7: Fault Mitigation Rules Activated for 10 Error Scenarios

tual target number projected for RTES/BTeV. Each Farmlet column is made of 6 boxes, with each box representing 1 Worker within the Farmlet. The Farmlet is represented by the yellow box located slightly above the column of 6 Workers, and simulates the actions taken by the Farmlet manager and Farmlet VLA. The centered green box at the top of the region simulates the regional manager and regional VLA.

Random fault scenarios activate specific system error messages at individual Worker, Farmlet, and regional nodes. The frequency with which each particular fault occurs is user-defined. This enables the system to simulate multiple fault scenarios in parallel, allowing for the demonstration of how simultaneous errors occurring at various levels within the system are processed. Errors introduced across the system are detected both proactively and reactively by individual components within the model, and are indicated within the simulation when the color of a given box turns red. A log of the sequence of errors occurring at each component is tracked, and can be viewed by clicking on the box representing the particular node of interest. In addition to a real-time status screen available for each Worker, Farmlet, region, and group node, a full error log tracks all error messages occurring at each time step across the system.

Individual proactive and reactive VLA rules are responsible for fault mitigation within the simulation at

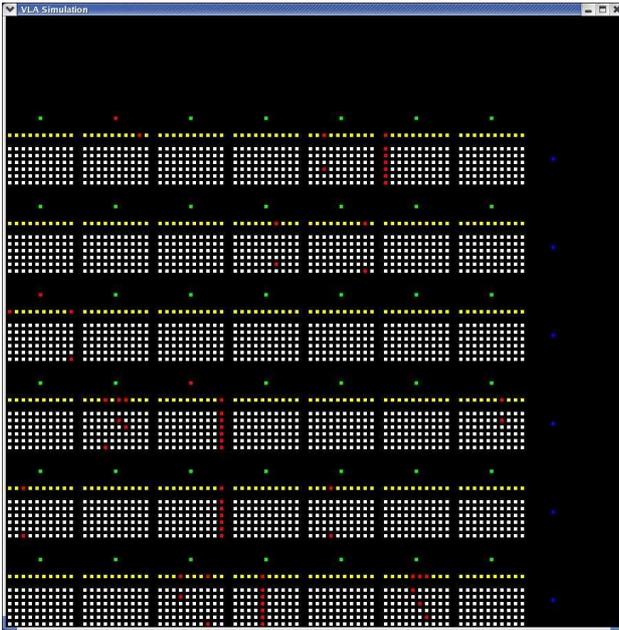


Figure 8: RTES/BTeV SWARM Simulation

each node. Every Worker VLA has a unique set of rules that it follows at each time step. Each rule is assigned a layer within the subsumption model with which it is associated. Rules assigned to higher layers that are responsible for higher *levels of competence* as defined earlier, subsume rules associated with lower layers. Likewise, every Farmlet, region, and group VLA has a unique set of rules that it follows, with higher layer rules subsuming lower ones.

As mentioned earlier, since scalability issues are a primary concern, the results must detect the degree to which any exhibited behavior is tied to specific system configurations. The SWARM model includes dynamic variables that can be modified to reflect various hardware layout configurations, such as the number of Workers per Farmlet, the number of Farmlets per region, and the number of regions per group.

Determining how to appropriately simulate time steps within any multi-agent simulation typically proves challenging. It is a particularly difficult issue in this case since the behavior of a large-scale, real-time, embedded system must be modeled. Since the primary goal is to simulate the interaction of component rules at various levels within the system, it suffices to demonstrate this interaction without explicitly capturing many of the exact processing rates and timing issues involved. Again, it is the general interaction of the rules at the various layers of subsumption that the implementation is primarily interested in simulating.

NEXT STEPS

After implementing all of the basic reactive and proactive rules of layers 0 and 1, the next step is to continue to evaluate and act on the rule firing statistics gathered at various levels of the system as defined for layer 2. From there, higher layers of the subsumption

model will be implemented in order to continue to classify characteristics of observed emergent behavior. It is expected that each layer that is implemented will provide further insight into effective strategies and techniques that can be used for adjusting individual agents towards desired behavior.

CONCLUSION

A SWARM multi-agent simulation is used to model BTeV, a large-scale, real-time, embedded system. The subsumption model detailed provides a framework for implementing specific levels of competence for multi-agent systems in incremental steps. The SWARM simulation of the RTES/BTeV environment provides an effective way for experimenting with local agent rules throughout the system to evaluate the impact each have on these individual layers. In this way, more may be understood about the specific qualities and structure of local rules and actions at these various layers that lead to global fault mitigative emergent behavior.

REFERENCES

- Brooks, R.A., 'Intelligence Without Representation', *Artificial Intelligence*, 47, 1991, 139-160.
- Brooks, R.A., 'A Robust Layered Control System for a Mobile Robot', *IEEE Journal of Robotics and Automation*, RA-2, (April 1986).
- Bapty, T. et. al, 'Modeling and Generation Tools for Large-Scale, Real-Time Embedded Systems'. IEEE Conference on the Engineering of Computer Based Systems (ECBS), Huntsville, Alabama, (April, 2003).
- Burkhart, R., 'Schedules of Activity in the SWARM Simulation System'. Position Paper for OOPSLA Workshop on OO Behavioral Semantics. (1997).
- Butler, J.N., et. al, 'Fault Tolerant Issues in the BTeV Trigger'. FERMILAB-Conf-01/427, (December 2002).
- Daniels, M., 'An Open Framework for Agent-based Modeling'. Applications of Multi-Agent Systems in Defense Analysis, a workshop held at Los Alamos Labs. (April 2000).
- Kwan, S., 'The BTeV Pixel Detector and Trigger System'. FERMILAB-Conf-02/313, (December 2002).
- Nordstrom, S., *A Runtime Environment to Support Fault Mitigative Large-Scale Real-Time Embedded Systems Research*, Master's thesis, Graduate School of Vanderbilt University, May 2003.
- Tamhankar, S., Oh, J., Mosse, D., 'Design of Very Lightweight Agents for Reactive Embedded Systems'. IEEE Conference on the Engineering of Computer Based Systems (ECBS), Huntsville, Alabama, (April 2003).

FROM π -CALCULUS SPECIFICATION TO SIMULATION OF A MOBILE AGENT USING JINI

Andreea Barbu and Fabrice Mourlin

University of Paris 12, LACL,
61. avenue du Général de Gaulle
94010 Créteil, France

barbu@univ-paris12.fr, fabrice.mourlin@wanadoo.fr

KEYWORDS: π -Calculus, process algebra, mobile agents, Jini

Abstract

This paper presents a formalism related to mobile agents. It describes the higher order π -calculus, an extension of the π -calculus who was introduced by R. Milner and later developed by Sangiorgi. This formalism defines a mathematical framework that can be used to reason about mobile code; it varies greatly in its expressiveness, in the mechanism it provides to specify mobile code based applications and in its practical usefulness for the validation and the verification of such applications.

In this paper we show how this formalism can be used to represent the mobility and communication aspects of a mobile code environment called HoPiTool. We developed the HoPiTool, which is a Jini-based tool to implement mobile agents who respect an initial higher order π -calculus specification. We will also introduce the structure of the HoPiTool with its most important elements.

1. INTRODUCTION - WHAT IS AN AGENT IN OUR FRAMEWORK?

Mobile agents are omnipresent in today's software applications and Java is a significant language to develop these agents. For a few years, Java Intelligent Network Interface (Jini) [1] has been more and more essential on the framework market, allowing the development in distributed networks. Jini is a tool based on Java and it enables us to realize, in a rather simple way, shared applications. Just as robots automate many aspects of manufacturing a computer, Jini automates and abstracts distributed applications' underlying details. These details include the

low-level functionality (socket communication, synchronisation) necessary to implement high-level abstractions (such as service registration, discovery, and use) that Jini provides. Jini was designed assuming that the network is not reliable. Things join the network and leave the network. There is no central control. Also, Jini blurs the distinction between hardware and software, dealing only with services. The objective of this work is first to describe the modelling of mobile agents and second to implement a Java Tool, which is able to generate Jini code from our π -calculus specification. Mobile agents or transportable agents are codes, which move on the network to fulfil a mission. For a better understanding of their behavior and to validate properties of them, such as for example, their return on their starting machine, it is necessary to formalize these aspects. A first realized study [2] leads to a higher order π -calculus formalisation. We use the formal language higher order π -calculus for the specification of the mobile agent and we chose the Java/Jini framework for its development. It is very significant to accentuate the higher order aspect because, to be able to develop a mobile agent, we need a language, which is able to express it like an essential characteristic. A mobile agent is a piece of code, which achieves a task required by a user. It must have a certain mobility to be able to move between various computers. After a correct specification of our mobile system, we can generate Jini code with the help of our HoPiTool and the generated files can communicate with each other in a network.

In a distributed environment, we can have one-to-many agent-hosts as well as one-to-many agents. To be an active agent platform, a given node in the system must have at least one active agent-host. Figure 1. describes the framework components. This component scheme can be mapped easily to the the Jini model. Jini, at the highest level, provides the infrastructure that enables clients to discover and use various services. Jini also provides a programming model for developers of Jini clients and services.

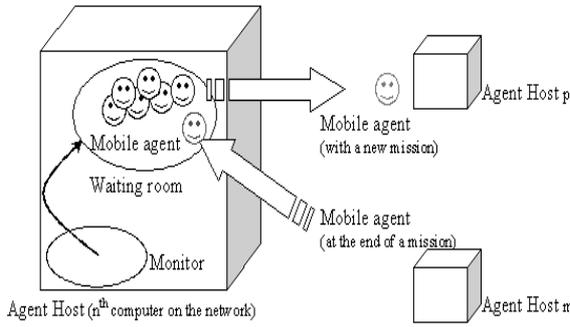


Figure 1. The software architecture of our framework

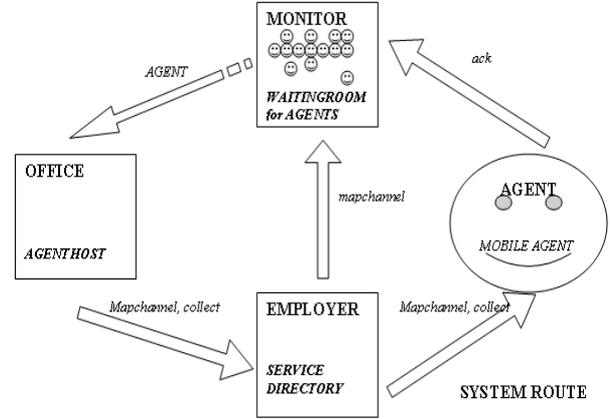


Figure 2. Our Route System

1.1. Case study

Our objective is to simulate a *MOBILEAGENT* that wants to get a job from a service collector called *EMPLOYER*. The *MOBILEAGENT* operates in a *ROUTE* system composed by a finite number of agent hosts, which we call *OFFICES*. Each *OFFICE* is able to make jobs and routes available by registering these to the *EMPLOYER* system. The *EMPLOYER* contacts the *MOBILEAGENT* to give him a certain job to do (in our case it will be *collect*) and a route map called *Mapchannel*. With this information, the *MOBILEAGENT* contacts directly the *OFFICE*, moves and gets the desired information from him (import of SQL orders into its bag). A *MOBILEAGENT* can ask also for services and contact many *EMPLOYERS*. It is possible to have more than one *EMPLOYER* service for a big *ROUTE* system. We can specialize many *EMPLOYERS* in order to make the offers of services manageable. To simplify matters we will not discuss this option in our paper. The *MONITOR* is a waiting room, where all the *MOBILEAGENTS* wait for some jobs. The *MONITOR* is also the starting and the ending machine for the *MOBILEAGENTS*. After carrying out the job, an acknowledgement *ack* will be sent back to the monitor. The higher order aspect in our case study is represented through the *MOBILEAGENT* that contains the job to do (*collect*), the route map (*Mapchannel*) or even another agents. Figure 2. depicts this simulation. In the context of our mobile agent framework, the agent host(s) provides Jini "collect" services. The mobile agent(s) is the Jini client. It can have a lease for on particular node and this one can be modify during the action of the agent. Our "collect" service provides a set of SQL statements which have to be executed later on a database. Jini services register with one or more Jini lookup-services by providing a service proxy for perspective clients. In turn, clients query

the lookup service(s) for particular services. For detail information about Jini structure look at [1, 3].

2. SPECIFICATION PART

The recent developments of data processing in a network confront the field of analysis and checking of parallel systems with new and complex questions. One concept which is the core of this evolution is that of mobility (of code, and more general of calculus over a network whose communication topology is dynamically changing).

A specification language must be powerful enough to express the mapping of a simulation model to any protocol on any target mobile architecture. The Pi-calculus expresses the move of data but also of code. This offers a solid base to make use of the Jini [3] generation. The Higher Order paradigm is a construct where mobility is achieved by allowing agents to be passed as values in a communication. The prototypical calculus in the first-order paradigm is the π -Calculus that was introduced by Milner, Parrow and Walker in [6] and later refined by Milner [5] with the addition of sorts and of communication of tuples (Polyadic π -Calculus). The π -Calculus is a way of describing and analyzing systems consisting of agents which interact among each other, and whose configuration or neighborhood is continually changing. The Higher Order π -Calculus ($HO\pi$) is an extension of the first order π -Calculus introduced by D.Sangiorgi [7]. This calculus enriches the π -Calculus with explicit higher order communications. In the $HO\pi$ -Calculus not only names, but also agents of arbitrarily high order, can be transmitted. A higher-order π -Calculus process is given by the following syntax:

$$P ::= \sum \alpha_i.P_i \mid P_1|P_2 \mid P_1 + P_2 \mid \nu x.P \mid [x = y]P \mid$$

$D\langle\tilde{K}\rangle \mid X\langle\tilde{K}\rangle$
 $\alpha ::= x\langle\tilde{U}\rangle \mid \bar{x}\langle\tilde{K}\rangle$

Where X is an agent (process) variable, $\langle\tilde{K}\rangle$ stands for any tuple of agent or (channel) name, and $\langle\tilde{U}\rangle$ stands for any tuple of variable or (channel) name. The constants D are defined as $D \stackrel{def}{=} (\tilde{U})P$. Constants are to be seen as functions whose parameters can be processes or other functions and:

- $\bar{x}(K).P$ can send the name or process K via the name x and continue as P .
- $x(U).P$ can receive any name or variable U and continue as P with the received name substituted for U .
- in the composition $P_1|P_2$, the two components can proceed independently and interact via shared names or processes.
- $\nu x.P$ is called the restriction and means that the scope of name x is restricted to P .
- in the sum $P_1 + P_2$ either P_1 or P_2 can interact with other processes.
- The matching $[x = y]P$ denotes the activation of a process which is selected by other processes on depend of a condition $([x = y])$.

The difference between first-order and higher-order π -Calculus resides in the fact that parameters can be channels and/or processes in the higher-order π -Calculus, while in first-order π -Calculus only channels can be passed as parameters.

Example: in $\bar{x}\langle P \rangle.Q \mid x\langle X \rangle.X$, once the interaction between the two processes has taken place, the resulting process is $Q|P$. Indeed, process $x\langle X \rangle.X$ was waiting for X to be sent along channel x , i.e., it was waiting for a process X defining its subsequent behavior.

The operational semantics is given in terms of a labeled transition system. There are three labels for the transitions: the silent step τ , the input action $x\langle\tilde{K}\rangle$ and the output action $\bar{x}\langle\tilde{K}\rangle$.

Output action: $\bar{x}\langle\tilde{K}\rangle.P \xrightarrow{\bar{x}\langle\tilde{K}\rangle} P$ means that after having sent message \tilde{K} (tuples of channels or processes) over channel x , process $\bar{x}\langle\tilde{K}\rangle.P$ behaves like P .

Input Action: $x\langle\tilde{K}\rangle.P \xrightarrow{x\langle\tilde{U}\rangle} P\{\tilde{U}/\tilde{K}\}$ means that if message \tilde{U} (tuples of channels or processes) is sent over channel x , then the process $x\langle\tilde{K}\rangle.P$, waiting for a process or channel name on x , receives it and instantiates the process or channel name to \tilde{U} , it then behaves like P , where all occurrences of \tilde{K} are replaced by \tilde{U} . $\langle.\rangle$ stands for real parameters, while $(.)$ stands for formal parameters.

Interaction between two processes:

$$\frac{P \xrightarrow{(\nu\tilde{K})\bar{x}\langle\tilde{K}\rangle} P', Q \xrightarrow{x\langle\tilde{K}\rangle} Q'}{P|Q \xrightarrow{\tau} \nu\tilde{K}(P'|Q')}$$

$\tilde{K} \cap fn(Q) = \emptyset$ means that if an output action causes P to become P' , and the corresponding input action causes Q to become Q' , then P and Q in parallel become P' and Q' in parallel, and the private (bound) process \tilde{K} emitted by P becomes a private (bound) process of $P'|Q'$.

Equivalences: Bisimulation usually identifies processes with the same external behavior. Higher-order bisimulation identifies higher-order processes if their interactions with the environment are the same and if their internal processes are bisimilar.

The case study $\text{HO}\pi$ specification is given below:
 $\text{ROUTE} =$

$$\begin{aligned} & \nu(out, out_{AH}, out_A, ack, service) \\ & \text{OFFICE}(out, mapchannel, collect) \\ & \mid \text{EMPLOYER}(out, out_{AH}, out_A) \\ & \mid \text{MOBILEAGENT}(out_A, service, ack) \\ & \mid \text{MONITOR}(out_{AH}, ack) \end{aligned}$$

$\text{OFFICE}(out) =$

$$\begin{aligned} & \nu(mapchannel, collect, agent) \overline{out}(mapchannel, collect) \\ & . mapchannel(agent). \text{OFFICE}(out, mapchannel, collect) \end{aligned}$$

$\text{EMPLOYER}(out, out_{AH}, out_A) =$

$$\begin{aligned} & \nu(channel, service) out(channel, service) \\ & . \overline{out_{AH}}(channel) \\ & . \overline{out_A}(channel, service) \\ & . \text{EMPLOYER}(out, out_{AH}, out_A) \end{aligned}$$

$\text{MONITOR}(out_{AH}, ack) =$

$$\begin{aligned} & \nu x out_{AH}(x) . \bar{x}(\text{MOBILEAGENT}) . ack \\ & . \text{MONITOR}(out_{AH}, ack) \end{aligned}$$

$\text{MOBILEAGENT}(out_A, service, ack) =$

$$\begin{aligned} & \nu(ch, serv) out_A(ch, serv) . \overline{ack} \\ & . \text{MOBILEAGENT}(out_A, service, ack) \end{aligned}$$

We specified as follows:

1. ROUTE : describes our whole system with all the components.
2. OFFICE : represents service-hosts, which make available different services (jobs) and notify their availability to the EMPLOYER service.
3. MOBILEAGENT : represents our mobile agents that are able to migrate to an OFFICE in order to

apply a job. The job is: *collect* a database information and add it to its "bag".

4. *EMPLOYER* plays a kind of reference book of all the tasks or jobs which are available on the local network.
5. *MONITOR* is the waiting platform for the mobile agents.

We use gates like *out*, *out_A*, *out_{AH}* to specify the communication channel between the agents, employers, offices and monitor. In this specification the mobility is described by the channel parameters.

3. IMPLEMENTATION PART

The specification and validation platform architecture is functionalities directed. The Figure 3. below emphasizes not only the functionalities but also the dependencies of data. The blocks represent the functional entities, whereas the ochre forms are strategic data.

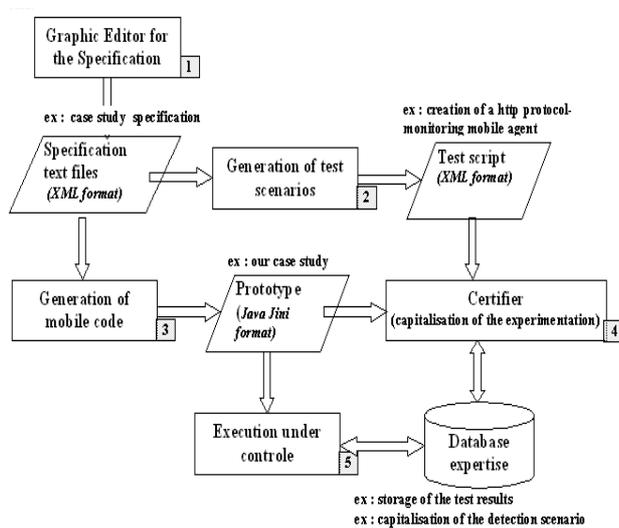


Figure 3. from Specification to Implementation

Our *HoPiTool* consist theoretically of five parts:

- The graphic editor for the specification. The objective is to represent by respecting the Unified Modelling Language (UML) notation, a specification where mobility is an essential characteristic. We propose five visualization windows for our editor:

- an XML visualization of our specification, where the XML file will be validate by the given HO π DTD
- a π -calculus visualization
- an HTML visualization in order to be able to show the documentation of each mobile agent
- an UML visualization like collaboration diagrams.
- a Java/Jini source code visualization

From these charts, files of textual description are generated with the XML format (under control that the diagrams are sufficiently complete). The internationalization (we use the given methods by Java) of the graphic interface aspects is an essential constraint. To developed global software, one of the great commandments of internationalization is to separate text, labels, messages, and other locale-sensitive objects from the core source code. This helps to maintain a single source code base for all language versions of our product. It also facilitates translations because all localizable resources are identified and isolated.

The major idea is to have a rigorous framework to control two important stages: the generation of test scenario and the generation of code.

- The generation of test scenarios
From XML files describing a π -calculus specification, which means an ensemble of mobile agents communicating through the services that they specified, the goal of this functionality is to generate test scenarios respecting the criterions given by the specification. These criterions are compared to the specification not operational, but denotational. The objective is to build tests which are used by the Certifier entity. Also, this test-set must answer to constraints of coherence, that means to satisfy a common goal such as: "execute all the sequences". In other words our first goal is to build structural "white box" and not functional tests called more commonly "black box". Each output file describes a script of tests and also the essential measurement points for the work executed by the Certifier.
- The generation of the mobile code
This functionality is initiated by the graphic interface and it allows to obtain a code prototype from valid HO π specifications. From XML files describing a π -calculus specification a whole of source files is generated in order to obtain a prototype. The aim is to gain ability to do simulations and prototype validations through the tests which we defined before.

More, HTML documents describing the documentation are generated by our tool, which takes in consideration the comments written within the specification. A specification consists of agents declarations. The whole of agents can be divided into two categories: mobile and non mobile. This constraint can be detected when an agent is send over a communication channel to an another agent. The mobile aspects take shape through the use of a technical framework called Java Intelligent Network Interface (Jini) from Sun. The fundamental element is the utilization of Remote Method Invocation (RMI), which allows us to consider the network like object oriented. The services and clients in a Jini architecture can be integrated easier within the object interface. Jini disposes also of identifications and collect services, that is a import future when we want to develop a mobile agent system. The security questions will be treat through Java Cryptography Architecture (JCA) Norm from Sun.

- The certifier
This entity takes a Java prototype as entry but also a whole of test-scripts. Every test is apply over the prototype in order to determine if the prototype validates this test. We can distinguish two cases: the prototype satisfies the test and this simulation will enrich the data base or the prototype doesn't satisfy the test, an anomaly is lifted and a log file is saved. The log files are used in a "post-mortem" analysis phase, which is the basis of the Software Validation and Verification Report (SVVR). The analysis is not a part of our project.
- Execution under control
Every execution of a generated prototype is realized under control. Every anomaly compared with the acquired experience is done through a call-back alert from the sites where the conflict was detected.

Our more important technical choices are the Java 1.4.2 Swing Framework, input, output in XML (with external validation in form of a DTD file), Java Intelligent Network Interface Jini 1.2., Remote Method Invocation (RMI), Oracle 9i. We use Extensible Markup Language (XML) to define and describe the higher order specification of our mobile agent. Our *HOPiTool* reads and generates Jini code from the XML file.

In our tool the XML file is read and parsed by a Simple API for XML (SAX) parser and an instance of *AgentFactory* class is created for each input XML file. A Document Type Definition (DTD) document describes how these XML files should be structured and the agent factory needs to have some knowledge of this DTD in order to know how to handle. Our *AgentFactory*

corresponds therefore to a Java code generator from an XML input file, which formally describes a π -calculus language specification.

We decided to use SAX instead of Document Object Model (DOM) because SAX requires little memory, it does not construct an internal representation (tree structure) of the XML data. Instead, it simply sends data to the application as it is read.

We defined a basic Document Type Definition (DTD) that describes the higher order π -calculus syntax and it is used to validate the XML file of our case study. This DTD states that an agent XML file contains a series of agents which play together in a mobile system. Each agent has a name, a number of arguments and a definition of his form and his objective. Based on the DTD we edit an XML file, which defines our case study, the world system, introduced before. We have defined each agent with the correspondents attributes and arguments. Based on this XML description of our mobile system and with help of our *HOPiTool* we are able to generate Jini code for a further communication of the agents in a network.

We use intensively design pattern such that: *AbstractFactory* pattern, *Composite* pattern, *Template* pattern and so on. This way of coding is an insurance for a better productively and maintainable code. We have defined four packages: *hopitool.gui*, *hopitool.lang*, *hopitool.parsing* and *hopitool.unification*. The *hopitool.lang* package provides an interface that describes the requirements for all the construction of Higher Order π -calculus language. Several classes like: *Call*, *Sequence*, *Parallel*, *Send*, *Receive*, *Choice*, *Restriction*, *Zero* represent the instruction set in HO- π -Calculus and the communication gates are represented by the class *Channel*. The *PiCalculusHandler* class contains the semantics of the Higher Order π -calculus language. The communication and the call of agents use parameters and expressions. The semantics of this data transfer is based on an unification algorithm [4]. In the current version of our tool we use a first order unification algorithm because all the agent names are considered as bound variables.

The second package *hopitool.unification* contains several classes for a *Template* design pattern or the unification concept. The *AbstractTerm* class represents the root class of a *Composite* design pattern. This pattern is applied for modelling a higher order π -calculus term. Its structure is always finite but its depth and its arity are not constant. We implement a finite first order unification to check the association between two terms. This operation is very useful for a *Call* statement and also for every communication, it means for *Send* and *Receive* instructions.

The *Template* design pattern is also used because it

allows us to delay a more complex algorithm for a higher order unification. This kind of concept is essential when an agent asks for a unknown service, but where its signature is already known by this agent. That algorithm is under development because, some constraints have to be checked before to compute this algorithm.

Further the class *Term* represents a non terminal node of a term in an *AbstractTerm* instances. It is used to model all functions or agents in an Higher Order π -calculus term. It is used for the unification step of terms. The class *Variable* represents a general variable in a specification. Instances of that class are built during the analysis of the input XML file. It concerns : all the terms which appear in a Higher Order π -calculus expression and all the Higher Order π -calculus parameters of local declared names. It is used for the unification step of terms.

The class diagram for the creation of all the objects are given below. The *Factory* classes are responsible for the creation of this classes, and the behavior is given by the class itself.

The *hopitool.parsing* package contains all the classes used for the analysis and the management of the symbol table. Some classes are used to a validating check. This classes represents the data structure of some lines of the parsing table. It contains all the data which describe this information in the specification language.

The *hopitool.gui* package contains all the viewers which belong to the graphical part of the tools. The graphical user interface (called GUI) has several features: each window created with *HOPiTool* has the five tabs presented in the first part of the "Implementation Part". Some dialog boxes are used for diagnostics such as a dialog for each previous tab, a help manual for developer, a start guide for the specifier. All entities are internationalized and some external text property files are already prepared for English, French, German and Spanish. This package is divided into several sub packages which corresponds to the views. Also, it is easier to add another one. In the current version, the GUI is used to display some information about the specification. When the user click on a tab the update of the corresponding view is computed. A given specification is observable only in a *HOPiTool* window. In the next version each view will support interactions from the user. All modifications have to be propagated on to the other views.

4. CONCLUSION

This work underlines the connection between the constraints of specification of higher order π -calculus and the mobile generation of code for the communicating systems. Our approach is validated by the construction of a proto-

type respecting the same constraints as in the specification. However, the *HOPiTool* represented here, is only a part of our implementation and only the "GUI", "generation of mobile code" modules are yet implemented. The generated Jini code is in our case study the generation of for java classes: *OFFICE* class, *EMPLOYER* class, a *MOBILEAGENT* class and a *MONITOR* class. These classes can communicate with each other in a network. An example for a framework for agent-based system in Jini is given by S. Li in [3] and it is called *Paradigma*. With help of this framework we can implement our Jini environment.

Also we are working on a test generator, which creates test sequences from initial π -calculus specification. Our intention is to obtain a test set, which will validate the generated code from *HOPiTool*. The test generator is configured with several criterions, which are essential to stop recursive unfolding in the agent definition.

The validation engine will be the final step of our study: it is an observer of the effects of the tests on the Jini prototype, generated by *HOPiTool*. Our intention is therefore to create an environment that allows the implementation of a mobile system of agents in Jini proceeding from a correct specification of the system given by the higher order π -calculus specification.

5. References

- [1] Arnold, K.; A. Wollrath, B. O'Sullivan, R. Scheifler and J. Waldo. 1999 "The Jini Specification", Addison-Wesley.
- [2] Barbu, A. and F. Mourlin. 2003 "Higher Order π -Calculus Specification for a Mobile Agent in Jini". In W. Dosch, editor, *4th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'03)*. pages 250-256, ACIS.
- [3] Li, S. and al. 2000 "Professional Jini" published by WROX Press Inc. ISBN 1861003552
- [4] Lloyd, J. W. 1987 "Foundations of logic programming". *Symbolic Computation - Artificial Intelligence*. Springer Verlag.
- [5] Milner, R. 1999 "Communicating and Mobile Systems: the π -Calculus". *Cambridge University Press*.
- [6] Milner, R.; J. Parrow and D. Walker. 1993 "A calculus of mobile processes, part I/II". *Journal of Information and Computation*, 100:1-77.
- [7] Sangiorgi, D. 1992 "From π -Calculus to Higher-Order π -Calculus - and back", In Proc. *TAPSOFT '93*, LNCS 668, Springer Verlag.

DEVELOPMENT AND PERFORMANCE OF A MASSIVELY PARALLEL REGIONAL SPECTRAL MODEL

Yifeng Cui
San Diego Supercomputer Center
University of California, San Diego
9500 Gilman Drive, MC0505
La Jolla, CA 92093-0505, U.S.A
E-mail: yfcui@sdsc.edu

Hann-Ming Henry Juang
National Center for
Environment Prediction
World Weather Bureau,
5200 Auth Road
Camp Springs, MD 20746, U.S.A

Giridhar Chukkapalli
San Diego Supercomputer Center
University of California, San Diego
9500 Gilman Drive, MC0505
La Jolla, CA 92093-0505, U.S.A

Masao Kanamitsu
Scripps Institution of Oceanography
University of California, San Diego
9500 Gilman Drive, MC0224
La Jolla, 92093-0224, U.S.A

KEYWORDS

Regional Spectral Model; RSM; domain decomposition; parallelization; message passing.

ABSTRACT

The Regional Spectral Model (RSM) is a nested primitive equation spectral model used by U.S. operational centers and international research communities to perform weather forecasts and climate prediction on a regional scale. In this paper, we present the development of an efficient parallel RSM with a message passing paradigm. Our model employs robust and efficient 1-D and 2-D decomposition strategies and incorporates promising parallel algorithms to deal with complicated perturbation architecture and ensure portability using hybrid MPI and openMP. We also achieve bit reproducibility when our parallel RSM is compared to the sequential code. Performance tests were performed on an IBM SP, Compaq, NEC SX-6 and the Earth Simulator and our results show good scalability at over a thousand processors.

INTRODUCTION

The Regional Spectral Model (RSM) is a regional scale climate model developed by the National Oceanic and Atmospheric Administration's National Center for Environment Prediction (NCEP) for the study of regional scale climate and conduct seasonal climate predictions (Juang and Kanamitsu 1994; Juang et al. 1997). The model is nested within NCEP's global spectral model (GSM), and the global model provides the RSM with its required initial and boundary conditions. The RSM requires large compute resources to provide high-resolution scale climate predictions focused on limited areas. For the past decade, RSM has run on high-end workstations, on serial or shared

memory parallel systems, and scalar or vector processor systems. However, the need to run larger and longer scenarios at high resolutions with more sophisticated physics exceeds the capabilities of the original single-node computer architectures. Increased availability of distributed-memory parallel architectures provides the motivation for this work, initiated in 2002, to implement a message-passing version of the algorithm. The current parallel version, completed in beginning 2003, supports the message-passing interface for implementation on scalable, distributed-memory computers.

The RSM is a nested primitive equation spectral model on a stereographic projection and uses sine-cosine series as horizontal basis functions, consisting of a low-resolution global spectral model and a high-resolution regional spectral model. Spectral models have proven to be effective in achieving high-order accuracy when compared with grid-point models (Juang and Kanamitsu 1994). The sequential implementation of RSM uses nesting technique to interpolate coarse-grid global data to fine-grid local data. The perturbation technique is used to force the solution on the regional scale simulations to the global scale coarse-grid solution. The complex parallel implementation of the RSM is designed to be highly flexible, so that the code can be run on a range of architectures, including distributed shared memory as well as distributed vector processors.

Spectral transform methods have important computational advantages (Bourke 1972), but are in many respects the most difficult to parallelize efficiently because of their highly nonlocal communication patterns (Foster et al. 1995). In this paper, we provide a comprehensive description of the design of the parallel RSM, and an evaluation of its performance on IBM

Power3, Power4, NEC SX-6 as well as on Earth Simulator platforms. We particularly emphasize the flexible 1D and 2D data decomposition, and offer a strategy to maintain code structure while improving performance. Throughout our discussion the Regional Spectral Model will be referred to as RSM and the Global Spectral Model as GSM.

REGIONAL SPECTRAL MODEL

The RSM is introduced to predict deviations from the global model, and to improve the forecast of poorly represented global-scale waves in the regional domain (Juang and Kanamitsu 1994). The RSM's prognostic variables are expressed as perturbations from its global counterparts. The RSM uses the same primitive hydrostatic system of virtual temperature, humidity, surface pressure and mass continuity prognostic equations on terrain following sigma coordinates as global spectral model GSM does. The philosophy behind this design is to use the regional model to generate regional scale features forced by the large scale. Thus, the large scale in the regional domain needs to be preserved. Consistency between the two models insures that the change of the large-scale motion in the regional domain is minimized.

The numerical method is described as follows. The difference between the regional model and the global model fields are decomposed into sine and cosine functions (as opposed to spherical harmonic functions in the global model). The sine and cosine functions conveniently satisfy the zero and symmetric lateral boundary conditions appropriate for the differences. The space derivatives are computed as a sum of the derivatives computed from global spherical coefficients and from the sine and cosine functions of the regional model.

The RSM has a high-order accuracy of spectral computations, and a time-dependent perturbation method, which distinguishes it from other regional spectral models (Juang et al. 1997). The spectral transformation of the RSM is a two-dimensional cosine series for perturbation of pressure, divergence, temperature, and mixing ratio, and a two-dimensional sine series for perturbation of vorticity. In the horizontal, the regional model uses double sine-cosine series with wall boundary conditions as base functions, while the global model uses spherical harmonics as base functions. In the vertical, the regional model uses exactly the same finite-difference formulation as in the global model.

One-dimensional Fast Fourier Transform is used in x direction while simple fourier summation is performed in y-direction, intentionally avoiding the use of the two-dimensional Fast Fourier Transform. This approach has

the advantage of reducing the memory requirement and is the best fit for distributed machine adaptation. Also, this method is analogous to the global spherical transform, and thus the program structure of the two models is very similar.

The RSM program system is managed by Concurrent Versions System (CVS) and controlled by the configure and Makefile system. The modeling system is composed of three components: libraries, source code, and run scripts. The *library* contains model libraries, utilities, and climatological/constant fields, machine dependent and resolution independent sources. The library needs only to be made once. The *source code* is used to create executables, defines model resolution and options, creates model resolution dependent constants, and compiles source codes and creates run executables. The *run scripts* are for running the model and producing the outputs.

The model output files include two restart files: one contains fields on sigma surfaces, and the other contains fields on ground surface including diagnostic files such as surface fluxes and precipitation.

IMPLEMENTATION OF RSM ON MESSAGE PASSING

The parallel RSM maintains the implementation structure used in the GSM, but also makes use of promising parallel algorithms. This concept was designed by Juang in 2002. A future paper will discuss the concept in more details.

The parallel implementation builds on the existing shared-memory model. The basic strategy of the implementation of RSM is to make a minimum of changes to the 500,000 lines sequential code while a scalable and portable parallel version of RSM can obtain bit reproducible results. The coding design emphasizes the flexibility and readability. For flexibility, the parallel RSM is designed to run in hybrid mode, so that the code can be used for a range of architecture including inner loop vectorization, outer loop multi-threads (openMP) and multi-nodes (MPI). For readability and easy maintenance, the MPI implementation is designed to be as consistent as possible with the parent Global Spectral Model.

To achieve these goals, the single program multiple data (SPMD) programming paradigm is used, such that each processor performs all compositions for only one subdomain. The model data structure is shrunk in the north/south dimension, using only as much memory as needed on each processor. The Fortran 90 data structure is implemented to simplify the data sharing among all routines through few passing arguments. In order to manage the necessary code change, the C pre-

processor, such as `#define` and `#ifdef` etc, are used to handle dependencies associated with shared- vs. distributed memory coding constructs without degrading the performance. For load balancing, local symmetric distribution of the grids is used and the computation in grids is partitioned so that each processor has approximately the same amount of work to do in each phase of the computation.

The perturbation method requires large memory and significant computation time to deal with base field over the entire regional domain. The parallel RSM employs a number of Fortran 90 features, such as dynamic memory and pointers, to achieve a more flexible and run-time configuration model suitable for distributed memory parallel computers. In particular, a dynamic memory management scheme is employed for all large model arrays to take full advantage of the large aggregate memory on distributed-memory platforms. An example of the coding design is:

```

#ifdef MP
  allocate (syn(igrd12_,levs_))
  ... ..
  deallocate (syn)
#else
  if(km.eq.1) then
    allocate (synpk1(igrd12p_jgrd12_))
    ... ..
    deallocate (syn1)
  else if(km.eq.levs_) then
    allocate (synpk2(igrd12p_,levsp_,jgrd12_))
    ... ..
    deallocate (synpk2)
  endif
#endif

```

where the C preprocessor is set for sequential or parallel (one or multiple vertical levels) implementation. Individual parallel tasks are assigned a range of model latitudes for which they are responsible.

In parallel RSM, both 1-D and 2-D decomposition are built for flexibility, which enables runs on any number of processors. 1-D can be used on any number of processors, but only up to the number of the smallest dimension among all directions and all space. The 2-D decomposition data transposition strategy utilizes a 2-D model data structures, meaning a single dimension of data structure in memory for each processor. This method has been applied successfully in several parallel atmospheric models (Foster and Worley 1997; Barros et al. 1995; Skalin and Bjorge 1997), including the Global Spectral Model. 2-D can be used up to the number of product of two smallest dimensions in all directions and all spaces, except with any prime number of processors (Juang and Kanamitsu 2001). In the current design, the total numbers of working

processors $NODES$ equals $Nrow$ times $Ncol$ where $Nrow$ and $Ncol$ are any integers in X and Y directions. If $NODES$ are prime integer, we set $Ncol=1$ and $Nrow=NODES$. In this case, 2-D decomposition becomes 1-D.

For a spectral model that uses transform method as in RSM, dependent variables and their tendencies take the form of spectral coefficients and grid point values at different stages of the computations. Each stage is characterized by full dimension in one of the three-dimensions. For example, when the Fourier transform is performed in an east-west direction, the full array in the east-west direction needs to be available on each processor, but the size in the north-south and vertical directions can be arbitrary. On the other hand, when the Legendre transform is performed, each processor needs the full array in north-south direction, but the array in east-west and vertical directions can be arbitrary. Thus, the parallelization of the spectral model requires rearrangement of arrays from one configuration to the other (for example, full array in x to full array in y). This rearrangement, also called the transpose method, requires communication between processors and is programmed using MPI routines. The transpose method is used to minimize the code modification - the original serial is left unchanged, the transpose routines are inserted between the transforms in east-west direction and in the north-south direction.

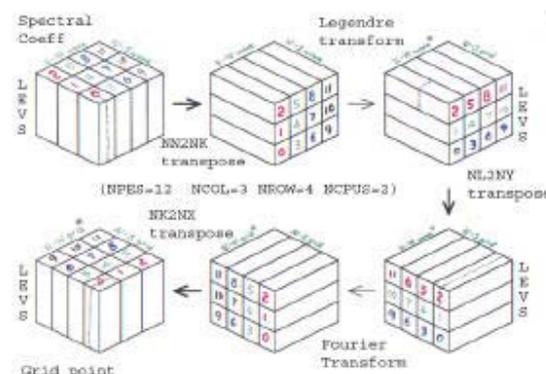


Figure 1: Parallel RSM Data Transposition Strategy, conceived by H.-M. H. Juang.

The entire transposition process is illustrated in Fig. 1. The general sequences of the forecast model after data are read in and transformed to spectral space include: transform from spectral to grid-point space, compute all non-linear model dynamics and model physics, transform back from grid-point to spectral space, and perform linear computation such as semi-implicit and time filter in spectral space. It requires only one transpose from the Legendre transform array configuration to the Fourier transform configuration for 1-D decomposition. All processors communicate with

each other. In contrast to this, 2D decomposition requires three transposes in forward direction, namely in advance of Legendre transform that has a north-south dependency, in advance of Fourier transform that has east-west dependency and in advance of nonlinear computation that has vertical dependency. In this case, communications only take place within a subgroup of processors. After the dynamics and physical computation with vertical dependency are complete, the data is transformed back with three more transposes in the reverse way.

The parallel code includes both the data model and the message passing model. The input/output (I/O) is the part of the code where the SPMD paradigm is not employed. Also, the global to regional spherical transform is not parallelized in the current effort. This is primarily because of very limited execution time compared with the regional forecast computation, as was indicated in the profiler performance analysis. The global coding is generally executed independently on each node.

Bit reproducibility was a goal of this parallel effort. This implies that on the testing computer, the parallel code be run on a variable number of processors and the sequential code should give identical results. To debug the message-passing version of RSM, we used a parallel debugging tool called Totalview, developed at Etnus, to compare serial and parallel version output and ensure bit-by-bit reproducibility. This produces a high level of confidence at each coding step. The final version of the parallel RSM produces results identical to the original sequential code on the same machine, independent of the number of processors on which it ran.

PARALLEL PERFORMANCE

The parallel RSM is portable, and only minimal modifications are needed to run the code on different platforms. The code has already been run on an IBM SP, Compaq Alpha, NEC SX-6 and the Earth Simulator. The performance numbers listed below are collected from an IBM SP and SX-6. The testing horizontal resolutions are at T62L28r12885 and at T62L28r512335 (global model resolution of triangular truncation 62 with 28 levels). T62 has sufficient resolution to give good large-scale predictions. Our test IBM SP platforms were the Power3 Blue Horizon and Power4 DataStar, both located at San Diego Supercomputer Center. The Power4 DataStar is a 7.9 teraflops IBM SP system which consists of 176 8-way 1.5Ghz p655 nodes and 7 32-way 1.8 Ghz p690 nodes, with total shared memory of 3.2 terabytes. The Power3 Blue Horizon is a 1.7 teraflop system with 144 8way SMP nodes. Each SMP node has 4 gigabyte of memory shared among its eight 375 Mhz Power3 processors. The vector machine NEC-6 is a

single cabinet, 8 cpu node with 64 GB of symmetric shared memory. Each cpu is a single-chip 8-way vector processor. The vector units operate at 500 Mhz. Peak performance is 8 GFLOPS/processor with 8 pipes given. The 8 vector register per CPU are 256 elements long and CPU-memory bandwidth is 32GB/sec/CPU (Baring 2003).

We performed our experiments in two stages. In the first stage, single node tests were performed for code optimization. In the second stage, we tested and analyzed scalability on equivalent platforms.

Optimization

On IBM SPs, we use xprofiler and the IBM SP hardware counter utilities HPM (Hardware Performance Monitor), to analyze the single-node performance and final options are tuned as:

```
FORT_FLAGS="-O3 -qfixed -qrealize=8 -qnosave -
qmaxmem=-1 -bmaxdata:0x80000000 -
bmaxstack:0x10000000"
```

```
LOAD_FLAGS="-O3 -qfixed -qrealize=8 -qnosave -
qmaxmem=-1 -bmaxdata:0x80000000 -
bmaxstack:0x10000000"
```

The total performance of the IBM SP3/SP4 is about 109/531 Mflop/s on a single processor at 128x85x28 resolution, resulting in a 7%/8% of peak performance.

On ARSC NEC-6, we set up some environments such as I/O variable F_FILEINF, profiling variable F_FTRACE and performance data variable F_PROGINF to analyze the code performance. Some exploration of the compiler options was made to achieve optimal performance. The compiler options are tuned as:

```
FORT_FLAGS="-Chopt -ftrace -f3 -float0 -ew -llpack_64 -
lblas_64 -lfft_64 -I./MODS"
```

```
LOAD_FLAGS="-Chopt -ftrace -f3 -float0 -ew -llpack_64 -
lblas_64 -lfft_64 -I./MODS"
```

For a few subroutines, they abort under the highest optimization (-Chopt) but not under a lower optimization (-Cvopt). Some modest modifications such as inlining are sufficient to achieve 35% of peak performance on average (2806Gflops/p), and 80% of peak performance in the best case for Legendre transform on the resolution at 720x715x28. The vector ratio achieves over 96%.

Scalability

The wall clock time of the parallel code runs on different number of processors are shown in Figure 2 for IBM SP3/SP4. Here, a time step of 30 seconds is used and

the simulation period is 6h. The solid lines represent the measurement on Power3, the dashed lines on Power4; the hollow points reflect the 128x85x28 resolution, the solid points the 512x335x28 resolution. The figure shows that the low-resolution model performance saturates quickly as the number of processors increase. For the high-resolution model, it scales to 1024 processors and the model efficiency does not diminish quickly with an increase in the number of processors. The smoothness of the curve is affected at certain points by a mismatch of the domain decomposition with the number of mesh points (Wehner et al. 1995), as well as a transition from using small number of nodes to larger number of nodes. The speedup and efficiency on various processor

configurations are summarized in Figure 3. The fine-resolution runs (512x335) take significant longer time on the dynamics and physics computation. At 1024 processors, the efficiency is reduced to 39%. There are several factors contributing to the falloff in performance. In addition to the remaining serial codes for the treatment of lateral RSM boundaries, we use the 'master-worker' strategy - the master processor distributes the work to the available processors and is responsible for reading and writing the history and checkpoint data on an hourly-averaged basis. As a result, I/O may account for 20-30% of the total elapsed time. Also, with the increased number of processors, the message communication cost is increased, and the load imbalance becomes noticeable, which will be discussed later.

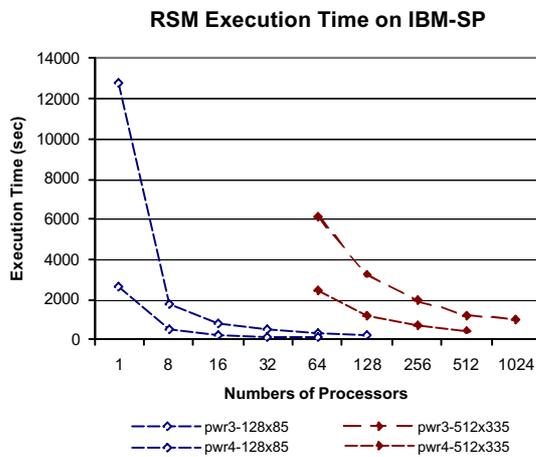


Figure 2: RSM Execution Time on IBM SPs

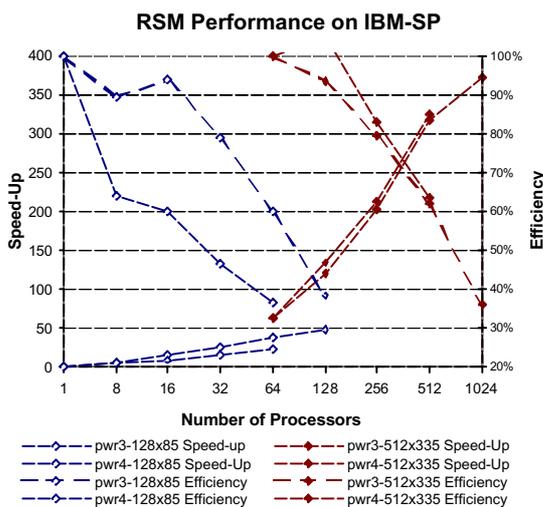


Figure 3: RSM Speedup and Efficiency on IBM SPs

The performance measured with 64-processors at 512x335x28 resolution is 137/357 Mflop/s on IBM SP3/SP4. Performance gradually degrades from this level to 60 Mflop/s on 1024 processors, which is fairly good when compared with other applications. The per-processor performance achieved on microprocessor-based parallel computers is often disappointing: a small fraction of peak. The reasons for this are not yet fully understood, but seem to be a combination of code structure (data structures and loop structures designed for vector machines often do not perform well on RISC microprocessors) and high memory bandwidth requirements (Drake and Foster 1995). Another reason for this is the parallel partitioning strategy designed in climate modeling. The spectral transform method imposes the most severe constraint on selecting a parallel decomposition strategy (Hack et al. 1995). The measurements above are based on system-independent libraries including Fast Fourier Transform (FFT) for portability consideration. We recently ported the code to use vendor-supplied subroutine libraries ESSL FFT, which increased performance by about 15%.

Communication Overhead

The parallel model has inter-processor communication overhead not present in the sequential model. The communication overhead in the parallel version of the RSM at 512x335x28 was approximately 34% on the 64 processor run, but increased to 52% on the 128 processor run. As mentioned above, the spectral model has the highly nonlocal communication patterns. The results indicate that the communication cost is the most responsible for sublinear performance scaling of the code on IBM SP system.

Load Imbalance

With an increasing number of processors the load imbalance becomes noticeable. The load imbalance accounted for about 12% of the total cost on the 16-

processor runHere load imbalance is calculated as follows:

$$L = \frac{T_{\max} - T_{\text{avg}}}{T_{\max}}$$

where T_{\max} and T_{avg} are the maximum and average execution times of MPI tests respectively. The load balance of 1.0 means all processors take exactly the same amount of time. Unlike the global spectral model that uses triangular truncation, the regional spectral model doesn't have a markedly uneven subdomain distribution in the 2-D decomposition. The load imbalance is mainly due to the different tasks of the processors such as the calculation in the physical parameterization (e.g. shortwave radiative transfer) or I/O.

Machine Comparisons

Our tests show that the one-dimensional decomposition is more efficient than the two-decomposition on SX-6 vector processors due to increased vector lengths. This is consistent with previous work on the Fujitsu VPP5000 (Juang and Kanamitsu 1997). However, this does not apply for cache-based IBM-SP. On the IBM SP, the 2-D decomposition is significantly better than 1-D decomposition. The tests on the Power3 show that on a small number of processors (8 to 16), the run based on 2-D decomposition saves approximately 20% of the execution time; on a large number of processors (such as 128), it saves approximately 40% of execution time.

Table 1: Performance Comparison on SPs and SX-6

	Power3	Power4	SX-6
1 cpu			
Execution time (sec)	902	193	94
Mflips/Mflops	132	593	1002
Sustained Performance*	8.8%	8.7%	12.5%
4 cpus			
Execution time (sec)	252	84	57
Mflips/Mflops	136	345	708
Sustained Performance*	9.1%	5.1%	8.9%
Efficiency	90%	57%	41%

* Percent of theoretical Peak Performance

A rough comparison of the performance on different machines is summarized in Table 1. A more detailed comparison, including the Earth Simulator, will be addressed later in another paper. Here, we test on the small resolution at 128x85x28 and set a time step of 360 seconds. The Power4 is more than 4 times faster than the Power3, but it sustains a much lower fraction of peak due to its relatively poor ratio of memory bandwidth to peak performance. Results demonstrate that the SX-6 achieves high sustained performance and significantly outperforms the superscalar designs of the

Power3 and Power4. In addition to its vector architecture and the shared-memory structure, the SX-6 has high memory bandwidth and low memory latency. The SX-6 vector unit lacks data cache. Instead of relying on data locality to reduce memory overhead, memory latencies are masked by overlapping pipelined vector operations with memory fetches (Oliker et al. 2003). Limited scaling on the SX-6 is mainly due to the reduced length of big vector loops as the number of processors increases. More work is being done to improve its scalability and efficiency on vector architecture.

CONCLUDING REMARKS

The production parallel Regional Spectral Model is designed to execute on massively parallel computer systems. The model employs a flexible 1D and 2D algorithmic decomposition, which has been implemented for both shared-memory and distributed-memory parallel architectures. With the parallel version of RSM, overall computational performance is quite respectable for distributed-memory implementation on IBM SP machine over number of nodes from a few to 1024 processors. Since the completion of the parallel implementation, the parallel RSM has been widely adopted by the international research community. In particular, the National Center for Environment Prediction (NCEP) has incorporated the parallel implementation into their operational RSM version for daily operational weather forecast and climate prediction.

Although the parallel RSM code scales up to 1024 processors with fine-resolution, the parallel speedup with RSM begins to show a significant degradation as the number of processors employed begins to approach the maximum number of available parallel processes, as seen with other climate models. The communication overhead and load imbalance are among the reasons of this degradation. A significant weakness in the current version of the parallel RSM is the treatment of I/O. Further implementation of parallel I/O techniques is needed to achieve sufficient I/O bandwidth to the parallel file systems. In addition, the model initialization step involving global to regional spectral transform has been done sequentially due to the small fraction of time used by this step.

During the course of this work, the RSM was ported to the Earth Simulator (ES) to prepare for a 50-year downscaling of atmospheric reanalysis over the continental US. While RSM performance is still being optimized on the Earth Simulator, we have already observed 1.6 Gflops preprocessor performance at the preliminary test of 10-km resolution at 1024x651x28 on ES, which is 20% of the peak performance. The ES is a NEC supercomputer that implements high-speed vector

processors in full scale and consists of 640 nodes connected by a 640x640 single-stage crossbar switch. The experience on ES will be very important in enabling meaningful comparisons between the capabilities of the Earth Simulator and IBM SP, and can guide future system upgrades to support RSM and similar applications including distributed simulations.

ACKNOWLEDGEMENTS

This work was supported by the National Partnership for Advanced Computational Infrastructure (NPACI) at the San Diego Supercomputer Center, NOAA Climate Dynamics and Experimental Prediction program, California State, and Climate Modeling Center at National Center for Environmental Prediction. We used the IBM SP3 Blue Horizon and SP4 Datastar systems at the San Diego Supercomputer Center, the NEC SX-6 system at the Alaska Arctic Region Supercomputing Center, and the NEC Earth Simulator of Japan.

REFERENCES

- Baring, T.L.. 2003. "SX-6 Compare and Contrast". *Techn. Report*. Arctive Regional Supercomputer Center, University of Alaska Fairbanks (May).
- Barros, S.R.M.; D. Dent; L Isaksen; G. Robinson; G. Mozdynski, and F. Wollenweber. 1995. "The IFS model: A parallel production weather code". *Parallel Computing*, No.21 (Oct), 1621-1638.
- Bourke, W.. 1972. "An efficient, one-level, primitive-equation spectral model". *Monthly Weather Review*, No.102, 687-701.
- Drake, J. and I. Foster. 1995. "Introduction to the special issue on parallel computing in climate and weather modeling". *Parallel Computing*, No.21 (Oct), 1539-1544.
- Foster, I.; B. Toonen and P.H. Worley. 1995. "Performance of Parallel Computers for Spectral Atmospheric Models". Tech. Report ORNL/TM-12986, Oak Ridge National Laboratory. Oak Ridge, Tenn., (Apr).
- Foster, I. and P.H. Worley. 1997. "Parallel algorithms for the spectral transform method". *SIAM J. Sci. Comput*, No.18(2), 806-837.
- Hack, J.J.; J.M. Rosinski; D.L. Williamson; B.A. Boville and J.E. Truesdale. 1995. "Computational design of the NCAR community climate model". *Parallel Computing*, No.21 (Oct), 1545-1569.
- Juang, H.-M. H. and M. Kanamitsu. 1994. "The NMC Nested Regional Spectral Model". *Monthly Weather Review*, No.122, 3-26.
- Juang, H.-M. H.; S.-Y. Hong and M. Kanamitsu. 1997. "The NCEP Regional Spectral Model: An Update". *Bulletin of the American Meteorology Society*, No.78, 2125-2143.
- Juang, H.-M. and M. Kanamitsu. 2001. "The Computational Performance of the NCEP Seasonal Forecast Model on Fujitsu VPP5000". *Ninth*

ECMWF Workshop on the use of high performance computing in meteorology (Nov 13-17). World Scientific under the title: Developments in teracomputing. Ed. by W. Zwiefelhofer and N Kreitz. Singapore, ISBN 981-02-4761-3.

- Oliker L.; A. Canning; J. Carter; J. Shalf and D. Skinner. "Evaluation of Cache-based Superscalar and Cacheless Vector Architectures for Scientific Computations". *Supercomputing 2003* (Phoenix, Arizona, Nov 15-21).
- Skalin, R. and D. Bjorge. 1997. "Implementation and performance of a parallel version of the HIRLAM limited area atmospheric model". *Parallel Computing*, No.23 (Dec), 2161-2172.
- Wehner, M.F.; A.A. Mirin; P.G. Eltgroth; W.P. Dannevik; C.R. Machoso; J.D. Farrara and J.A. Spahr. 1995. "Performance of a distributed memory finite difference atmospheric general circulation model". *Parallel Computing*, No.21 (Oct), 1655-1675.

AUTHOR BIOGRAPHIES

CUI, YIFENG received his Ph.D. in Hydrology from University of Freiburg. He also holds a bachelor degree in Meteorology from Nanjing Institute of Meteorology, and a master degree in Hydroclimatology from Hohai University in 1987, China. He worked for a couple of years for Environment Canada before moving in 2000 to the University of California, San Diego, where he now works on parallel performance with the Scientific Computing Department at the San Diego Supercomputer Center.. His e-mail address is: yfcui@sdsc.edu.

HANN-MING HENRY JUANG is a research meteorologist at the National Center for Climate Prediction, NOAA. His e-mail address is: henry.juang@noaa.gov.

GIRIDHAR CHUKKAPALLI received his Ph.D. in Mechanical Engineering from University of Toronto, Canada. He has an extensive background in CFD algorithms (finite element/ finite difference, spectral schemes). He is currently a senior computational scientist with the Scientific Computing Department at the San Diego Supercomputer Center, and has extensive experience with both shared memory and message passing programming. His e-mail address is: giri@sdsc.edu.

MASAO KANAMITSU is a research meteorologist at Scripps Institution of Oceanography, University of California, San Diego. His e-mail address is mkanamitsu@ucsd.edu.

Performance Analysis of Continuous Cell-DEVS models

Gabriel Wainer
Dept. of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa, ON. K1S 5B6. Canada.
gwainer@sce.carleton.ca

KEYWORDS

DEVS, Cell-DEVS, Quantized DEVS, CD++.

ABSTRACT

The Cell-DEVS formalism allows describing complex cellular models. When using very large models with continuous variables, the execution performance degrades. We experimented with different quantization techniques to reduce the number of messages generated by the simulator. We present two different strategies for the automatic update of the quantum sizes in different cells, and we discuss the use of quantized DEVS with hysteresis applied to cellular models. We obtained important reductions in the error involved, while maintaining the high performance of quantized DEVS models.

INTRODUCTION

In the last 20 years, cellular computing became popular as a tool for complex systems analysis. Cellular Automata (CA) [1] are organized as n -dimensional infinite lattices in which each element holds a state variable and a very simple computing function. These functions are local to each cell, and they execute synchronously using the state values of the present cell and neighbors. The Cell-DEVS formalism [2] permits describing cellular models as discrete event systems based on the DEVS formalism [3]. A real system modeled with DEVS can be described as a composite of behavioral (atomic) and structural (coupled) submodels, which can be integrated into a hierarchy. In Cell-DEVS, each cell is seen as a DEVS atomic model, and a procedure for coupling cells is defined based on the neighborhood relationship. Explicit timing delays permit expressing complex timing conditions. The hierarchical nature of DEVS also permits the integration of these cellular models with others defined using different formalisms, resulting in enhanced facilities for modeling complex systems.

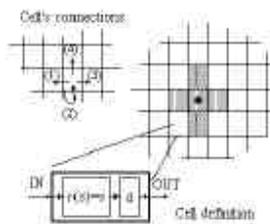


Figure 1. Description of a Cell-DEVS Model.

The CD++ toolkit [4] allows simulating DEVS and Cell-DEVS models, and it has been used to execute a

variety of models (traffic, forest fires, biological systems and experiments in physics). As the complexity of the models grows, large data sets are generated, increasing execution. The situation is worse for models with continuous state variables.

Different efforts that showed how to simulate continuous DEVS models efficiently include quantized DEVS (Q-DEVS) [5], Q-DEVS with hysteresis [6], and Cell-DEVS with Dynamic Quantization [7]. When using quantized DEVS, a state value will be only informed to its neighbors if the cell's value crosses a threshold (called the *quantum*). This operation reduces substantially the frequency of message updates, while potentially incurring into error. DEVS with Hysteresis has strong stability, convergence and error bound properties. Cell-DEVS with Dynamic Quantization reduces the error by improving the precision of the most active cells.

We used different models as a workbench, including a model of the electrical activity of the heart tissue [8], a model of watershed formation [9], and a Flow Injection Analysis (FIA) system [10], which studies the automated analysis of liquid samples. We analyzed different metrics (error, execution time and number of messages), and we could determine how the dynamic quantization techniques improve the amount of error introduced by the quantizers.

DEVS MODELING

A DEVS atomic model can be defined as: $AM = \langle X, Y, S, \delta_{xt}, \delta_{int}, \lambda, ta \rangle$. A DEVS model in state $s \in S$ will remain in that state for a period as defined by $ta(s)$. When $ta(s)$ expires, an internal transition runs: the model outputs the value $\lambda(s)$, and changes to the state $\delta_{int}(s)$. A state transition can also happen upon reception of an external event: δ_{ext} is activated with the input value, the current state and the elapsed time.

Coupled models, are defined as: $CM = \langle X, Y, D, \{Mi\}, \{Ii\}, \{Zij\} \rangle$. They consist of a set of basic models (Mi , atomic or coupled) connected through the models' interfaces. Component identifications are stored into an index (D). A translation function (Zij) is defined by using an index of influencees created for each model (Ii). The function defines which outputs of model Mi are connected to inputs in model Mj .

Cell-DEVS allows the creation of cellular models, in which each cell is defined as a DEVS atomic model. A Cell-DEVS atomic model is defined as: $TDC = \langle X, Y, S, \theta, N, d, \delta_{int}, \delta_{ext}, \tau, \lambda, D \rangle$. A cell uses a set of input values N to compute its future state, which is obtained by applying the local function τ . A delay function is δ -

sociated with each cell, after which, the new state value is transmitted to the neighbor cells. After the basic behavior for a cell is defined, a complete cell space can be built as a coupled Cell-DEVS: $GCC = \langle Xlist, Ylist, X, Y, n, \{t_1, \dots, t_n\}, N, C, B, Z \rangle$. A coupled Cell-DEVS is composed of an array of atomic cells (C), each of which is connected to the cells in the neighborhood (N). The border cells (B) can be programmed with a different behavior than the rest of the space. The Z function defines the internal coupling of cells in the model. Xlist and Ylist define the coupling with external models.

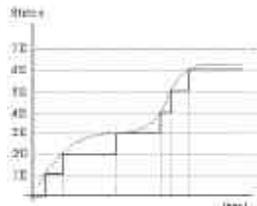


Figure 2. Quantized DEVS [5].

Performance of models with continuous variables is reduced due to the large number of messages interchanged by the simulation engines. The theory of quantized DEVS [5] represents continuous signals by the crossings of an equal spaced set of boundaries (defined by a *quantum* size), as showed in Figure 2. This operation reduces substantially the frequency of message updates, while potentially incurring into error. DEVS Quantized systems with hysteresis [6] improved these results (models showed to have strong stability, convergence and error bound properties). Let $D = \{d_1, \dots, d_m\}$ be a set of real numbers where $d_{i-1} < d_i$, $x \in \Omega$ is a continuous trajectory where: $x: \mathbb{R} \rightarrow \mathbb{R}$ and $b: \Omega \times \mathbb{R} \rightarrow \Omega$ is a mapping where $q = b(x, t_0)$ that satisfies:

$$q(t) \begin{cases} d_m & \text{if } t = t_0 \\ d_{i+1} & \text{if } x(t) = d_{i+1} \wedge q(t^-) = d_i \wedge i < r \\ d_{i-1} & \text{if } x(t) = d_{i-1} \wedge q(t^-) = d_i \wedge i > r \\ q(t^-) & \text{otherwise} \end{cases}$$

$$m \begin{cases} 0 & \text{if } x(t_0) < d_0 \\ r & \text{if } x(t_0) > d_r \\ j & \text{if } d_j \leq x(t_0) < d_{j+1} \end{cases}$$

Here, the hysteresis width is E and the parameters d_0 and d_r are the lower and upper saturation values. In [6], the authors proved that when the hysteresis width is set equal to the quantum size, we obtain the smallest possible error.

Cell-DEVS with Dynamic Quantization [7] tries to reduce the error by improving the precision of the cells. An active cell can appear as quiescent due to the selection of a quantum size covering the activity area, and if the quantum size is reduced, a smaller error will be obtained. Simultaneously, if we increase the quantum size in the cells with steep update functions, the execution time can be improved at a low cost in terms of the error introduced. Two different strategies were proposed to adjust the quantum size. Let q be the base quantum, r the adjustment ratio for the dynamic quantum, and $d(t)$

the quantum value used in time t . If $v = \text{Last Threshold Value}$, $v' = \text{new value}$, and $q(0) = q$, then:

$$\begin{aligned} \text{Strategy 1} & \rightarrow \text{regionChange}(v, v', d) \Rightarrow d = q * (1 - \text{ratio}); \\ & \text{regionChange}(v, v', d) \Rightarrow d = q * (1 + \text{ratio}); \\ \text{Strategy 2} & \text{regionChange}(v, v', d) \Rightarrow d = q * (1 - \text{ratio}); \\ & \neg \text{regionChange}(v, v', d) \Rightarrow d = q * (1 + \text{ratio}); \end{aligned}$$

where $\text{regionChange}(v, v', q) = (v = \phi \mid q = 0 \mid (q \neq 0 \wedge [v/q] \neq [v'/q]))$. Strategy 1 tries to reduce the quantum size if the result of updating the cell's value does not cross the threshold (otherwise, the quantum increases). This technique reduces the quantum size for cells with high update rates, and increases it for cells with low update rates. Strategy 2 reduces the quantum size every time a threshold is crossed (otherwise, it increases). This strategy reduces the number of messages involved in the simulation at a cost of a higher error.

We tested these techniques on CD++, which was originally built as an implementation of DEVS and Cell-DEVS theories. Cell-DEVS models are defined using a built-in language. The behavior of a cell is defined using rules with the form: VALUE DELAY {CONDITION}. If the CONDITION is satisfied, the state of the cell will change to the VALUE, and this new state value will be spread to the neighboring cells after the DELAY. Common operators are included: boolean, comparison and arithmetic; trigonometric, roots, power, rounding and truncation, module, logarithm, absolute value, minimum, maximum, etc. [4].

DEFINING COMPLEX CELL-DEVS MODELS

We used different complex models as a workbench, including a heart tissue model [8], a watershed [9], and a Flow Injection Analysis (FIA) system [10]. These examples, are the most complex cases of a larger pool of models we executed, and they represent different categories of interest. The heart tissue model permits analyzing systems in which the behavior of all of the cells is alike. The watershed model, instead, permits analyzing systems that tend to a steady state after a transient period. Finally, the FIA model shows a case of a model in which the behavior of different cells does not follow a predefined pattern.

A model of the heart tissue behavior

The heart muscle is excitable, and its cells respond to external stimuli by contracting the muscular cells. Hodgkin and Huxley [11] originally characterized the behavior of this cell membrane. They showed that if a stimulus is too weak, the muscle does not respond; instead, if the voltage received is adequate, they contract at maximum capacity. Whereas solving the equations representing this behavior using numerical methods for one cell is feasible, the use of this model in a realistic reproduction of the heart tissue (probably consisting of millions of cells) can be computationally expensive.

Different authors tried to simplify the complexity of the equations; for instance [12], presented the use of CA (with simple rules for the model's behavior, but at a cost in precision) and in [8] we presented the use of Cell-DEVS to build a discrete variable model of heart tissue

conduction. Our Cell-DEVS model executes the Hodgkin-Huxley model in each of the cells, as follows.

```
[heart]
dim : (50,50) delay : transport
border : nowraped
neighbors : (-1,-1) (0,-1) (-1,1) (0,-1) (0,0)
(0,1) (1,-1) (1,1) (0,1)
localtransition : heart-rule-AP

[heart-rule-AP]
rule : { AP(0,0) } 1 { ((-1,0) > 0 or
(0,-1)>0 or (-1,-1)>0) and (0,0)=-83.0) }
```

Figure 3. Heart tissue model: Cell-DEVS definition.

We first define the size of the cell space (50x50 cells), the kind of delay, and the neighborhood shape (in this case, all the adjacent cells). Then, we define the local computing function, called *heart-rule-AP*. The rule will be evaluated only if the cell is resting and a positive voltage is detected in the cell's neighborhood. This rule will trigger the update of the cell state using the Hodgkin-Huxley equations in [11].

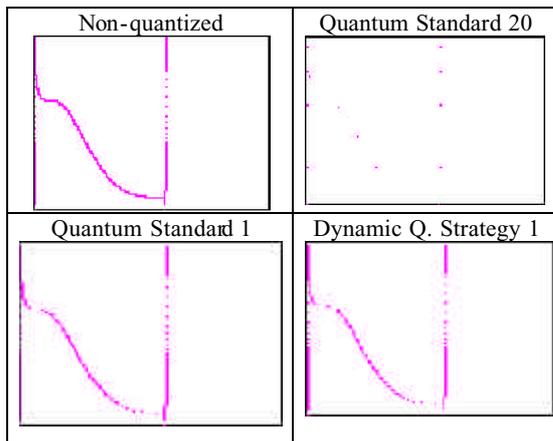


Figure 4. Sample execution of the Heart tissue model.

A Watershed model

Watersheds are regions defined by the shape of the land surface, which store up water because of rain, ice melting and rivers. In [13], the authors defined a hydrology model in which they identified several verticals layers composing a watershed: air, vegetation, water surface, land surface and stones. The model was divided in equal portions of land (cells), permitting an analyzing the water distribution and the influence of the topology.

Figure 5 shows a description of this model. When the rain is absorbed by the vegetation, the rest is received by the surface. Depending on the topology, the cells can also receive/send, water from/to the neighbors. Part of the water received is lost due to the filtration over the land and stones. The accumulated water on a period depends on: the quantity of effective water (rain), the quantity of water dumped from the neighbor cells (effective rain plus the water received from the neighbors minus water sent to the neighbors) minus the water filtered by stones and the soil.

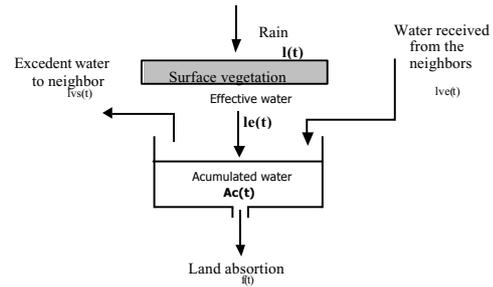


Figure 5. Hydrology model [13].

We can see the execution results of this model in Figure 6. We first show the initial state, representing the slope of the terrain before raining (each cell is 1x1m). The remaining figures show the execution results after intense rain (0.0022 mm/s) after 10 minutes of simulated time.

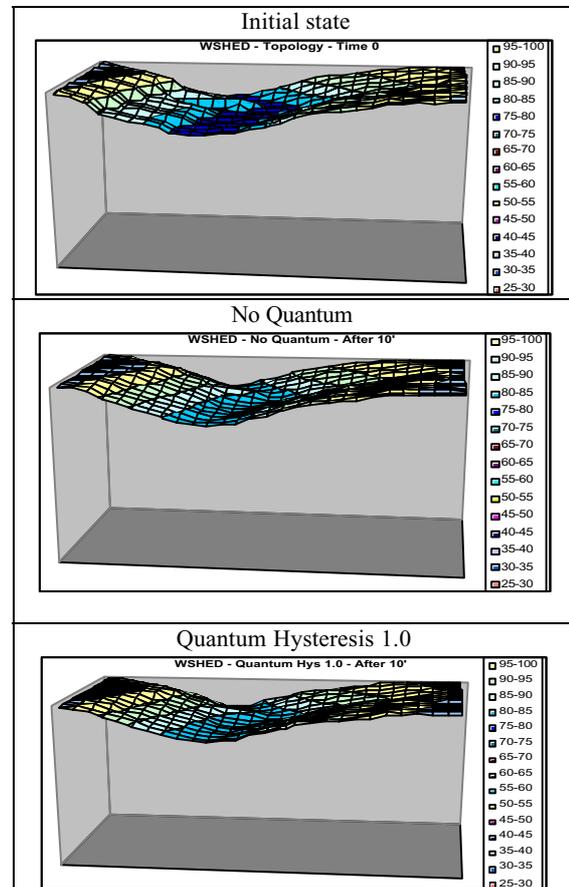


Figure 6. Watershed simulation results.

These results were obtaining by running the hydrology equations in CD++, using a mechanism similar to the one presented for the heart tissue model. These rules represent that the present water of the cell, and the rain are added. Then, we consider how much water must be passed to the neighbors, and how much water is received from the inverse neighborhood. The different layers are represented as planes in a three dimensional model.

```

[Watershed]
dim: (30,30,2) border: nowrapped
delay: transport localtransition: Hydrology
neighbors : (-1,0,0) (0,-1,0) (0,0,0) (0,1,0)
(1,0,0) (-1,0,1) (0,-1,1) (0,0,1) (1,0,1) (0,1,1)

[Hydrology]
rule : {0.0022+(0,0,0) - if((-1,0,0)!=?) and
((0,0,1)+(0,0,0)>((-1,0,1)+(-1,0,0)),((0,0,0)+
(0,0,1)-(-1,0,0)-(-1,0,1))/1000)*(0,0,0)
/1000),0)-if((1,0,0)!=?) and((0,0,1)+(0,0,0))>
((1,0,1)+(1,0,0)),((0,0,0)+(0,0,1)-(1,0,0)-
(1,0,1))/1000)*(0,0,0)/1000),0) - if(((0,-1,0)
!= ?) and ((0,0,1)+(0,0,0))>((0,-1,1)+(0,-1,0)
),((0,0,0)+(0,0,1)-(0,-1,0)-(0,-1,1))/1000)*
(0,0,0)/1000),0)-if((0,1,0)!=?) and ((0,0,1)
+(0,0,0))>((0,1,1)+(0,1,0)),((0,0,0)+(0,0,1)-
(0,1,0)-(0,1,1))/1000)*(0,0,0)/1000),0)+ if(
(-1,0,0)!=?) and ((-1,0,1)+(-1,0,0))>((0,0,1)+
(0,0,0)),((-1,0,0)+(-1,0,1)-(0,0,0)-(0,0,1))*
(-1,0,0)/1000),0) + if((1,0,0) != ?) and
((1,0,1) + (1,0,0))>((0,0,1)+(0,0,0)),((1,0,0)
+(1,0,1)-(0,0,0)-(0,0,1))*(1,0,0)/1000),0)+
if((0,-1,0)!= ?) and ((0,-1,1)+(0,-1,0)) >
((0,0,1)+(0,0,0)),((0,-1,0)+(0,-1,1)-(0,0,0)-
(0,0,1))*(0,-1,0)/1000),0)+if((0,1,0)!=?) and
((0,1,1)+(0,1,0))>(0,0,1)+(0,0,0)),((0,1,0)
+ (0,1,1)-(0,0,0)-(0,0,1))*(0,1,0)/1000),0) }
1000 { cellpos(2)=0 }

```

Figure 7. Hydrology model in CD++.

A Flow Injection Analysis Model

Flow-injection methods are used for automated analysis of liquid samples. In a flow injection analyzer, a small fixed volume of a liquid sample is injected as a discrete zone using an injection device into a liquid carrier. Because of convection at the beginning, and axial and radial diffusion later, this sample is progressively dispersed into the carrier as it is transported along the tube, which can be sensed by flow-through sensors.

In [9] we built a Cell-DEVS model describing the integrated conductivity in detail. The model studied a 0.025 cm radius tube, a 10.75 cm loop and a 9,25 reactor coil. We assumed the total tube length of the tube to be of 20 cm and defined a cell space of 25 rows and 200 columns.

```

[fia]
width : 200 height : 25 delay : inertial
border : nowrapped
neighbors : (-1,-1) (-1,0) (-1,1) (0,-1) (0,0)
(0,1) (1,-1) (1,0) (1,1)
localtransition : transport

[transport]
rule : {(0,-1)} {0.1/(22.578*(1-
power(cellPos(0)*.001+.0005,2)/.000625))*1000}
{cellPos(1)!=0}
rule : {0.8} {.1/(22.578*(1-power(
cellPos(0)*.001+.0005,2)/.000625))*1000}
{cellPos(1)=0}

[diffusion]
rule : {((-1,0)+(0,0)+(1,0))/3} 1 {cellPos(0)!=
0 AND cellPos(0)!=24}
rule : {((-1,0)+(0,0))/2} 1 {cellPos(0)!=0 AND
cellPos(0) = 24}
rule : {((0,0)+(1,0))/2} 1 {cellPos(0)=0 AND
cellPos(0) != 24}

```

Figure 8. A model of FIA in CD++.

The value of each cell represents the concentration of nitric acid in the carrier. To deal with convective transport and radial diffusion at the same time, the model reacts in two phases: transport and diffusion.

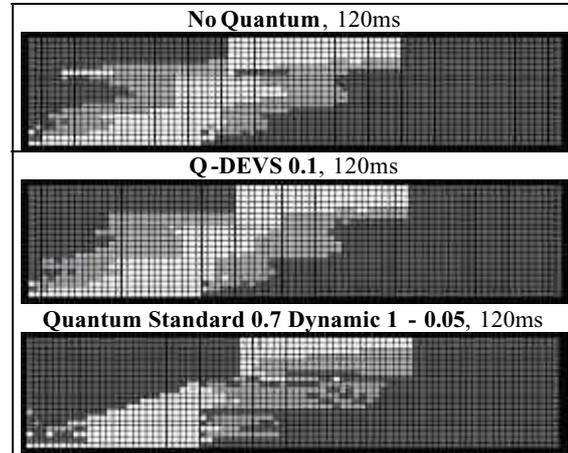


Figure 9. FIA Result Simulation Examples.

PERFORMANCE ANALYSIS

We executed simulations of the models presented in the previous section, analyzing two main metrics: execution time (number of messages involved in the simulation), and error. We executed a large number of tests in different categories, including:

- Non-quantized (noted as DEVS in the figures).
- Standard quantization (noted as Q-DEVS).
- Quantization with hysteresis (noted as H).
- Dynamic quantization, strategy 1.
- Dynamic quantization, strategy 2.

Different combinations of the previous categories with different quantum sizes and update ratios were used. The error is obtained by comparing the values in quantized versus non-quantized cases.

Figure 10 presents the cumulative error obtained with different strategies for the heart tissue model. The figure compares the different strategies (using different update ratios for the Dynamic DEVS strategies 1 and 2). The results obtained with standard and hysteresis quantum overlap, because the results of Hysteresis quantum differ from the standard when direction changes are present (and, as seen in Figure 4, there is only one).

The lowest error was obtained with dynamic quantum Str1 with ratio 0.9. Str1 results were better than the Str2 and standard Q-DEVS (the larger the ratio, the better the result), as expected. The quantum size is adjusted very quickly, which reduces the amount of error obtained. Similarly, in the non-linear section of the function, the quantum is quickly adjusted to a smaller size. Str2 error is large because each update the quantum size increases.

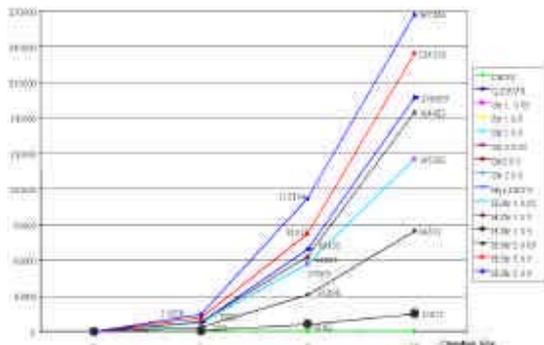


Figure 10. Heart tissue model: cumulative Error.

The number of messages is reduced up to a 99.95%. These empirical results verify the theoretical conclusions presented in [4], and reproduce the general shape of the message reduction found in [7]. Nevertheless, the amount of error involved was highly reduced.

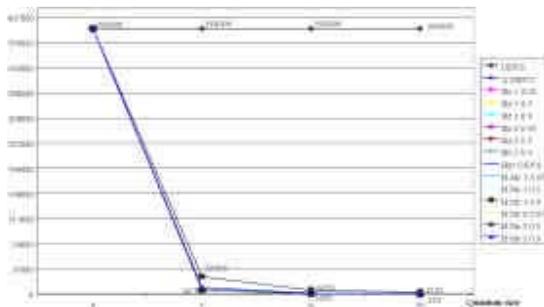


Figure 11. Heart tissue model: No. of messages.

Adaptive quantization improves the error when compared with Q-DEVS. Str2 expands the quantum size every time a threshold is not crossed, increasing the associated error. Str1 obtained the least improvement when compared with the rest, nevertheless, its overall execution time improves 6.57% vs. 1.58% for $Q=1$, and 0.8% vs. 0.06% for $Q=20$.

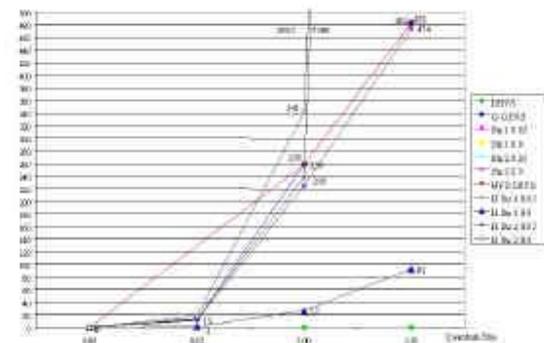


Figure 12. Watershed model Cumulative Error.

The Watershed model was tested using the land topology presented in Figure 5. In Figure 12, we present the cumulative error obtained when running this model.

These results repeated the pattern obtained for the heart tissue model: no difference between standard and hysteresis quantization (the watershed function has an

increasing linear function for the topology and rain conditions chosen), the lowest error was obtained with Str1 with ratio 0.9, and Str1 results were better than Str2 and Q-DEVS. The order of the different strategies is maintained, while the total error is smaller.

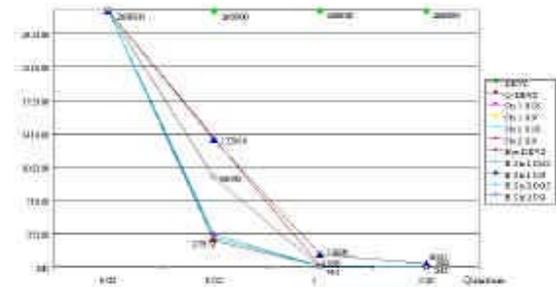


Figure 13. Q-DEVS Watershed model Output Messages

The best execution time was for Q-DEVS and Str1 with an update ratio of 0.05 (larger update ratios adjust the values quicker, reducing the error while increasing the number of messages). When using $q=0.05$, Q-DEVS provides better results. As each cell increases approximately 0.07 units in each update, changing the quantum size makes it oscillate around the function value, resulting in an increase in the total simulation messages. This does not occur with fixed quantum size. Likewise, once the quantum size varies, the dynamic quantum strategies have lower message interchange. A low update ratio improves the number of messages involved, while increasing the error. Paying a small cost in the extra execution overhead, we were able to reduce the error involved (up to 75%). Str2 reduces the number of messages using higher rates when compared to Str1, but incurring in a higher amount of error. If we consider, for instance, $q=1$ with Str1 and ratio 0.9, the amount of error introduced is minimum and the number of messages has been highly reduced. If we consider now $q = 3.5$, the error obtained with Str1 is better, while the number of messages involved is comparable.

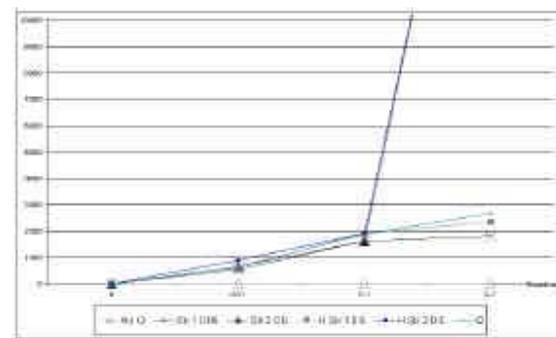


Figure 14. FIA model Cumulative Error.

In the FIA model, we observed a different error pattern: hysteresis quantum provided a more stable behavior. Str1 with a small update ratio improves the overall error because of the model behavior, presented in Figure

15. The diffusion combined with transport affect the results in cells closer/farther from the sample.

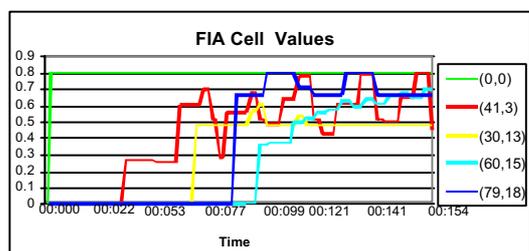


Figure 15. Individual cell behavior on the FIA model.

Here, Str2 with a larger update ratio always diverges. With Str1, a small update ratio improves the results (because it adjusts better to the different values).

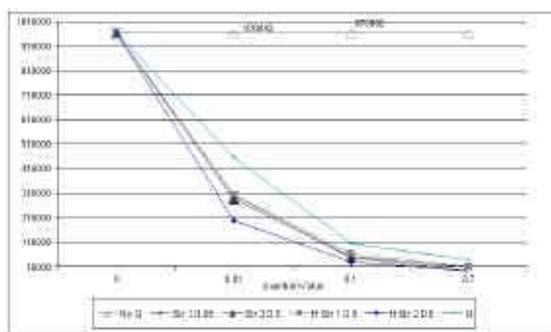


Figure 16. FIA model: number of messages.

The simulations with the highest error rate have provided the best execution times, as expected. Nevertheless, the amount of error obtained has highly reduced at the cost of little overhead.

CONCLUSION

We presented the analysis of quantization techniques for the execution of continuous variable Cell-DEVS models. We presented two different strategies for automatic updating of the quantum sizes in different cells. We obtained important reductions in the error obtained, while maintaining the high speed of quantized DEVS models. We used different complex models as a workbench, representing different categories of interest.

In every case, the lowest error was obtained with the dynamic quantum strategy 1. According to the model, updating the dynamic quantum size with higher/lower ratios improved the simulation results. In every case, the number of messages was reduced with quantization (up to 99.95% of reduction). Likewise, we could see that the introduction of hysteresis quantum permits to obtain a more controlled behavior, even for applications with cells executing with a non-linear pattern.

REFERENCES

[1] WOLFRAM, S "A new kind of science". Wolfram Media, Inc. 2002.

- [2] WAINER, G.; GIAMBIASI, N. "Timed Cell-DEVS: modelling and simulation of cell spaces". In "Discrete Event Modeling & Simulation: Enabling Future Technologies", Springer-Verlag. 2001.
- [3] ZEIGLER, B.; KIM, T.; PRAEHOFER, H. "Theory of Modeling and Simulation". Academic Press. 2000.
- [4] WAINER, G. "CD++: a toolkit to define discrete-event models". In Software, Practice and Experience. Wiley. Vol. 32, No.3. November 2002.
- [5] ZEIGLER, B.P. "DEVS Theory of Quantization". DARPA Contract N6133997K-0007: ECE Dept., The University of Arizona, Tucson, AZ. 1998.
- [6] KOFMAN, E.; JUNCO S "Quantized State Systems. A DEVS Approach for Continuous Systems Simulation". Transactions of the SCS. 18(3). 2001.
- [7] WAINER, G.; ZEIGLER, B. "Experimental results of Timed Cell-DEVS quantization". In Proceedings of AIS'2000. Tucson, Arizona. U.S.A. 2000.
- [8] N. GIAMBIASI, G. WAINER. "Using G-DEVS and Cell-DEVS to model complex continuous systems". Accepted for publication in Simulation: Transactions of the SCS. 2003.
- [9] AMEGHINO, J.; WAINER, G. "Application of the Cell-DEVS paradigm using CD++". In Proceedings of the 32nd SCS Summer Computer Simulation Conference. Vancouver, Canada. 2000.
- [10] TROCCOLI, A.; AMEGHINO, J.; IÑON, F.; WAINER, G. "A flow injection model using Cell-DEVS". Proceedings of the 35th IEEE/SCS Annual Simulation Symposium. San Diego, CA. 2002.
- [11] HODGKIN, A.; HUXLEY, A. "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve" Journal of Physiology 117: 500-544. 1952.
- [12] SAXBERG, B.; COHEN, R. "Cellular Automata Models of Cardiac Conduction". In Theory of Heart. Springer-Verlag. 1991.
- [13] MOON, Y.; ZEIGLER, B.; BALL, G.; GUERTIN, D., "DEVS representation of spatially distributed systems: validity, complexity reduction". IEEE Transactions on Systems, Man and Cybernetics. pp. 288-296. 1996.

GABRIEL WAINER received the M.Sc. (1993) and Ph.D. degrees (1998, with highest honors) of the Universidad de Buenos Aires, Argentina, and Université d'Aix-Marseille III, France. He is Assistant Professor in the Dept. of Systems and Computer Engineering, Carleton University (Ottawa, ON, Canada). He was Assistant Professor at the Computer Sciences Dept. of the Universidad de Buenos Aires, and a visiting research scholar at the University of Arizona and LSIS, CNRS, France. He is author of a book on real-time systems and another on Discrete-Event simulation and more than 70 research articles. He is Associate Editor of the Transactions of the SCS. He is Associate Director of the Ottawa Center of The McLeod Institute of Simulation Sciences and a the chair of the SISO DEVS standardization Study Group.

Modeling and Simulations of Biomedical Data Networks

Pauli Lallo
Signals and Electrotechnical School
PL 5 11120, Riihimäki, Finland
pauli.lallo@kolumbus.fi, Finland

KEYWORDS

Modeling, simulation, transactions, adaptive modem, waveforms.

ABSTRACT

This paper presents a simulation research of a data network and data transmission for biomedical and telemedicine purposes. The information exchange, adaptive data waveforms for wireless channels and traffic simulations of data flow is the substance of this investigation. Simulations have been made with a Comnet 3 network traffic simulator package.

First section is a motivation section showing examples of innovative use of data networks and emerging new terminologies. Biomedical information technologies combined with data network, collaboration, grid, adhoc and virtual organization are discussed.

Second section is the basis of the study. It deals with customer data communication needs. The communication channels can be radio channels or wired lines (analog or digital). In the latter case digitized analog voice (voice coding) is transmitted by Voice over IP (VoIP). The channel bandwidth of the network is usually a bottleneck of performance. Thus the transmission capacity of the network is an important target of the simulation investigation. We investigate the need of the channel bandwidth in bits per second by simulating data transmission delays in connectionless data networks. Delay simulation results of some standard wireless data communications systems are compared with adaptive data waveform results.

In third section modeling and simulation results of the network design are investigated. Data transmissions are evaluated by simulations of biomedical or telemedicine traffic using modeled data transmission networks. The quality of service (QOS) in digital data networks is measured with bit error rate (BER) or time delay. BER of a communication channel describes QOS of the modulation method used. It depends on the signal-to-noise (S/N) ratio of the channel.

In last sections we propose and discuss the adaptive data communication design principles of the networks. We evaluate the proposal modulation selection system and the use of secure adaptive waveforms. We design end-to-end security on the physical OSI reference model level 1 and at the same time we improve throughput of data traffic of our network. The investigation of the sensitivity of a network structure, the message flow fluctuations and the growth process are important. We present principles for network design and channel capacity planning. The channel bandwidth and the S/N are well known parameters in network design by Shannon's formula. With this metric we get different limits for wireless channel traffic (bit rates), transactions in the system and data network performance.

INTRODUCTION

New emerging technologies combined with innovative use of data networks are studied in many countries. Three motivating examples for military use from USA present information technology areas (USA DoD 2000):

- Advance distributed learning (ADL).
- Smart Sensor Web (SSW).
- The biomedical technology area.

New information technologies present the opportunity to make significant improvements in training and education effectiveness. The DoD enterprise of education and training offers a key opportunity to reduce costs in these domains.

Smart Sensor Web (SSW) is a recent focus inspired by extraordinary technological advances in sensors and microelectronics and by the emergence of the Internet as a real-time communication tool. The near future will see a proliferation of sensors and associated processors available for battlefield use. The overall vision for SSW is an intelligent, secure, web-centric distribution and fusion of sensor information that provides greatly enhanced situational awareness, on demand, to Warfighters at lower echelons.

The biomedical technology area is focused to yield essential technology in support of the DoD mission to provide health support and services to U.S. Armed Forces. Most national and international medical investment is focused on public health problems of the general population. Military medical study is concerned with developing technologies in order to preserve combatant health and optimal mission capabilities despite extraordinary battle and non-battle threats to their wellbeing. Preservation of individual health sustains warfighting capabilities.

The Ohio State University had started a prototype implementation and presented a paper of grid support for collaborative clinical and biomedical research studies (Hastings et al. 2002). Throughout many areas in science and commerce, there is an increasing recognition of the need to support the information service needs posed by overlapping, often ad-hoc collections of work groups and organizations. They used the name "virtual organization" (Foster et al. 2002) to refer to an ad-hoc collection of work groups and organizations. Most work groups and organizations already make use of their own information systems. The term Grid (Chervenak et al. 1999, Foster et al. 1999) refers to hardware and software infrastructures and frameworks that provide access to heterogeneous collections of computational resources across multiple institutions and multiple domains. The underlying premise of Grid computing is to harness this wide-area network of resources into a distributed computation and data management system.

A lot of information technologies are available for telemedicine (xDSL, OFDM, GPRS, 3G) but not discussed here. Our research is focused on

- Biomedical information transactions as the source of the telemedicine data traffic,
- Quality of the service of present data networks (delays, BER, S/N) achieved with standard data transmission methods,
- Security design of mobile wireless communication channels and
- Adaptive modulation methods (waveforms) as a new approach.

The results of this study may be applied to the military sensor communication (data fusion) and other secure ad-hoc communication needs. Our research method is modeling and simulations with a data traffic simulation package Comnet 3 (Caci 1995).

INFORMATION EXCHANGE

In this section we present simulation results and discuss old and present information exchange possibilities, nature of data traffic, mathematical distributions used in our modeling and our adaptive communication system for use in telemedicine data channels.

Need for Information Exchange

The basis of this study is in the customer data communication needs. The need, nature and benefits of the biomedical information exchange is described in references (Hasting et al. 2002, Lees et al. 2002, Andrade et al. 2001-2002):

- Analysis of medical images requires the ability to query, retrieve, and process large amounts of image data. In a collaborative environment these studies could be aggregated from multiple participating institutions.
- In collaborative environments, data replication across storage systems and caching of data hot spots close to groups of clients can significantly increase throughput and decrease response times seen by clients (Andrade et al. 2001-2002). Data replication and caching would alleviate the bottleneck that will be created by a single data source.
- Application dependent processing of large volumes of data is a critical component in many fields of medicine.
- The regional health-telematics-network, interconnects healthcare facilities in the region, supporting the development of the regional healthcare information infrastructure.
- Internet technology provides secure multilingual access to medical records in a cardiology department.
- The scalable and modular architecture can support interconnectivity between medical records of cardiology patients in departments within the same and different institutions.

Modeling and Simulation

We have modeled and simulated message transmissions of source instruments and data transmission of a biomedical or

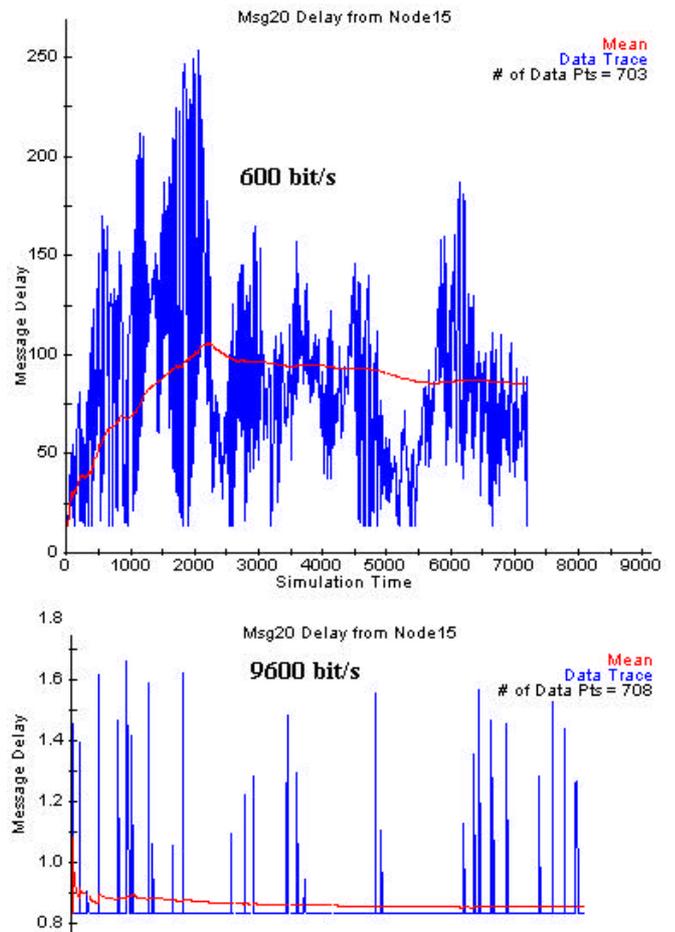


Figure 1: Message Delays in Seconds

telemedicine network model. The network model is based on Comnet 3 (Caci 1995), Figures 1-4. The simulation model, traffic parameters and results are discussed in more detail in the next sections.

Nature of Data Traffic

The nature of data traffic was seen in Figure 1. The main difference between data and voice communication is traffic bursts in data transmission. Figure 1 shows burst data traffic simulation results in a backbone network structure of Figure 4. Wireless band-limited data rates of the channels are evaluated based on the present day realism in military and commercial equipments. Delays depend heavily on the data transmission lines used in simulations as the results of mean delays indicate:

- <0.5 s delay with 9600 bit/s and
- <100 s delay with 600 bit/s.

Many heavy unpredictable variations in traffic and delay times are seen in the low bit rate case. The modeling of the data traffic has been found difficult in many references. We used the well-known exponential distribution for traffic generation with 10 sec arrival times, Figure 3. In Figure 2 we had a fixed 1000 byte mean message size. Delay times with the 600 bit/s lines are not acceptable but 9600 bit/s

might be acceptable in many real world applications. This example demonstrates with 600 bit/s rate the radio channels of a common analog military message system and with 9.6 kbit/s rate cellular data channels widely used in GSM technology.

Bit Rates

In Figure 2 we have simulated data transmission over the standard telecommunication channels in an example network using different wireless or wired technologies and standard bit rates in band-limited channels :

- Radio channels 600, 2400 and 9600 bit/s.
- Adaptive radio channels 22500 bit/s.
- Wired ISDN channels 64 kbit/s.

The lowest bit rate 600 bit/s is used in a conventional analogue military tactical vhf radio messaging system. The low bit rate channels 2400 bit/s are common in all new digital military hf/vhf radios. This data bit rate is also available with newest frequency hopping waveforms, which can secure our data transmission. Cellular data transmission (GSM) has been used since the beginning of mobile Internet connections with 9600 bit/s data channels. Adaptive radio channel is made with our prototype modem added to a conventional tactical radio. The system was field-tested in 2000 (Lallo 2001). Wired ISDN technology has a fixed 64 kbit/s bit rate. The simulated message delays with these channels are presented in Figure 2. Interesting is the standard deviation of delay as an indicator of QOS.

We are not discussing here the present ADSL or other wide-band technologies, which can offer 512 kbit/s or 2 Mbit/s. Although ADSL-technology has rapidly developed in the wired telecommunication networks as the basic access system to the Internet but we concentrate on the mobile wireless communication and mobile band-limited Internet access in this study. Next we will show how critical is the channel bandwidth measured as bit/s.

Message Delay

The quality of data transmission in our modeled network is evaluated with message delay simulations. Figure 2 shows simulated message delays versus bit rate. Table 1 gives also the maximum delays. We can found that 64 kbit/s data rate is good enough in burst data transmission case with the zero delay deviation. Message delay (average and its deviation) is a good measure for quality of service (QOS). In some instrumentation cases a low bit rate may be practical because it saves channel capacity and maybe channel costs. If our instrumentation references are biomedical instruments of a hospital or military sensors in the field there may be a specific QOS limit for a real time system delays. In Figure 2 an important result is that QOS decreases rapidly when the bit rate of the transmission channel is under a certain limit. The limit in this simulation is 2.4 kbit/s or 9.6 kbit/s depending on the particular message delay requirements of the application in concern.

Table 1: Delay Simulation Results (ms)

Bit rate	Average	Std dev	Maximum
600	24797.64	15677.19	92547.3
1200	7911.91	2472.74	18013.3
2400	3552.63	650.25	6479.4
9600	846.24	78.73	1479.4
22500	358.27	17.30	523.9
64000	125.84	0	125.8

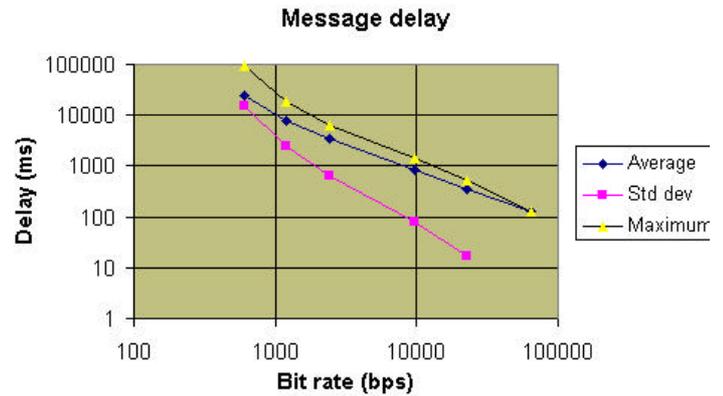


Figure 2: Message Delay vs Channel Bit Rate

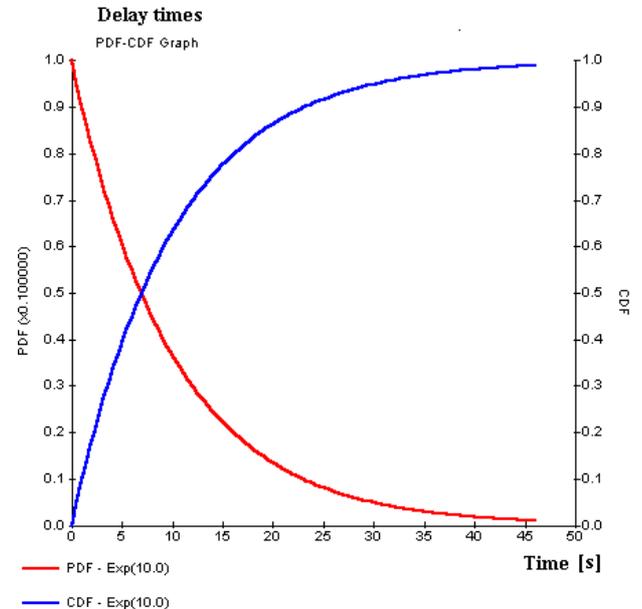


Figure 3: Message Arrival Times (Caci 1995)

2.4 kbit/s value is the maximum value of most tactical radio channels. In cellular world the value is higher 9.6 kbit/s or 14.4 kbit/s with GSM based data transmission or with GPRS technology about 50 kbit/s in practice. With these options we have now good possibilities to use wireless communication for instrumentation. A software modem with adaptive modulation, waveform and bit rate selection is a perfect and secure communication tool for different band-limited channels, noise and multi-path conditions (Lallo 2002).

A practical measure for the traffic quality is the deviation of the message delay found in simulation, Figure 2. We found that 64 kbit/s data channels offer zero delay time deviation, which means that no bursts were found in the data traffic, Table 1.

Security

The indoor instrumentation might be based on new standards like Bluetooth or wireless LANs. Although the security on OSI reference model upper levels in data communications over Internet is solved with tunneling methods there is a lack of the OSI level 1 security in LANs. However, we can have a secure IP-based virtual private network (VPN) between biomedical LANs. There is a VPN standard for tunnel management (RFC 1701/1702). Tunnels can originate and terminate in the enterprise server or in the service provider's access switch. Security problems are in LANs themselves, where we do not have OSI level 1 security. Thus we propose the use of adaptive end-to-end (PC-to-PC) secure waveforms inside LANs for solving this securing problem.

PRINCIPLES OF NETWORK DESIGN

In this section we study adaptive data communications with adaptive waveforms for biomedical and telemedicine networks, Figures 5-7 and Table 2. There are design principles for reducing blocking or delay times:

- Alternate routing.
- Increasing of channel capacities

Alternate routing is used both in telecommunication and Internet networks. Our network simulation model in Figure 4 had two alternate routes in the backbone network model. Adaptive modulation is a method to select wireless channel capacity by waveform selection. In the wireless world it means the selection of the digital modulation method (waveform) according to the channel characteristics. The selection of the error free modulation method is a key issue in the adaptive modem concept. In a mobile wireless environment we have time varying different channel characteristics (bandwidth, S/N) and thus it is advantageous to fit the digital modulation method to each channel condition.

Selection of Waveform

Table 1 shows the simulation results using different channel bandwidths in the backbone network representing different wireless or wired technologies. We see the benefits of adaptive waveform selection data rates, Tables 1-2 and Figure 6.

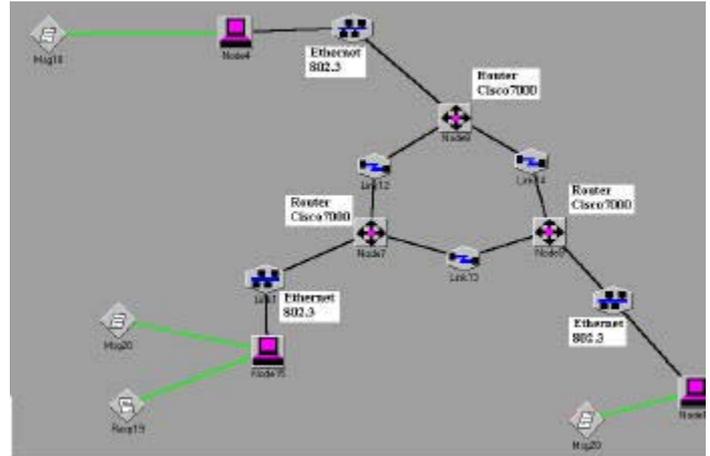


Figure 4: Example Network Model (Comnet 3)

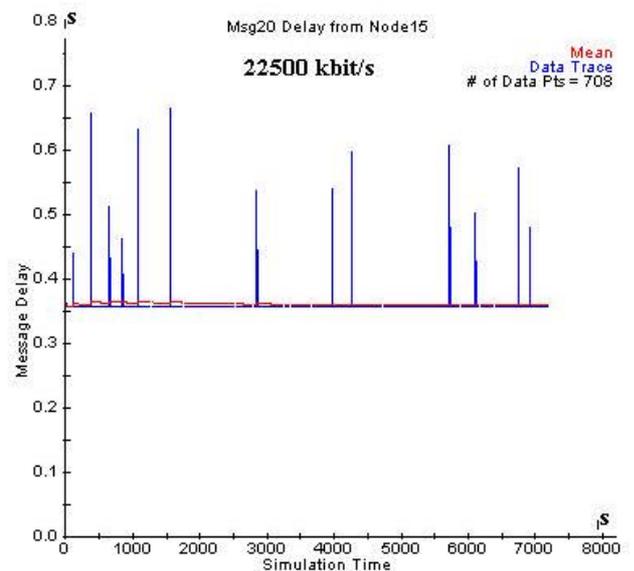


Figure 6: Message Delay (22500 bit/s Backbone)

Usually data network are based on international or industry standards and their data rates are fixed. In the simulation results we see that the selection of a non-standard 22.5 kbit/s adaptive waveform of a prototype modem (Lallo 2001) gives a lower average message delays 0.36 s than a standard 9.6 kbit/s standard cellular data modem 0.85 s, Table 1. This data rate is good enough to give also low deviation.

Quality of Service (QOS)

A good measure of QOS is the standard deviation of the message delay or the maximum delay. Data bursts cause the deviation of the delay times, Figures 1 and 6. Bursts will disappear if we have more bandwidth for example 64 kbit/s, Figure 2. An important principle in network design is the use of delay deviation in monitoring and controlling the

traffic versus the channel bandwidth (capacity in bit/s). Present data traffic and its growth is used as one of the main design factor in network capacity design. Knowing these factors we can predict the channel capacity needed in the future by delay simulations.

Adaptive Multi-Carrier Waveforms

Knowing channel characteristics much better bit rates than with present standard systems discussed earlier are available using if the adaptive multi-carrier data communication is used, Table 2. Table 2 shows bit rate possibilities simulated with an adaptive modem worksheet simulator (Lallo 2001). The adaptive modem adapts to the available bandwidth using a proper number of carriers (channels 1-10 in Table 2). Each carrier uses the best available modulation method (QAM states in Table 2) adapted to the channel characteristics usually the S/N ratio. A software modem should select automatically the best downloadable software algorithm and modulation method and thus maximizes the bit rate in mobile band-limited wireless communication.

Physical Security with Waveforms

We have made a proposal for a secure network model for biomedical or telemedicine communication needs, Figure 5. This proposal includes the use of adaptive modems as a data communication securing tool on OSI level one (physical).

Transaction Rate

We investigated increasing transaction rates in a network using the 22.5 kbit/s adaptive waveform in the backbone channels. We increased gradually the network traffic from message sources having a fixed normal distributed message size (100 kilobyte mean, 10 kilobyte standard deviation) and arrival time as variable (10 sec - 160 sec). These arrival times gave us a practical transaction range of a biomedical institution with 65 000 to 260 000 transaction per year. The results of these simulations are in Figure 7. Delays in Figure 7 have an average (ave) and a standard deviation (std) value. We found that there is a transaction value limit about 250 000 per year after which the backbone network causes increasing delay time. In the evaluation of the simulated result the message size (100 kilobyte used) is a critical measure. If the messages include for example large images (1 Megabyte) the number of transaction limit may drop to 25 000 per year. However, it is possible to work with a low data rate band-limited backbone but with larger delays.

We have to take care about our data traffic culture (mean message size, transactions per day, rush hours etc). We can develop our principles for network design. We can predict the transactions and the future number of traffic sources. Simulations can be done using the system model with parameters according to the predicted situation. Then economical decisions can be made about the communication channel and network capacity. In existing biomedical network structures telephone network and wired Ethernet 802.3 LAN are very often used, Figure 5.

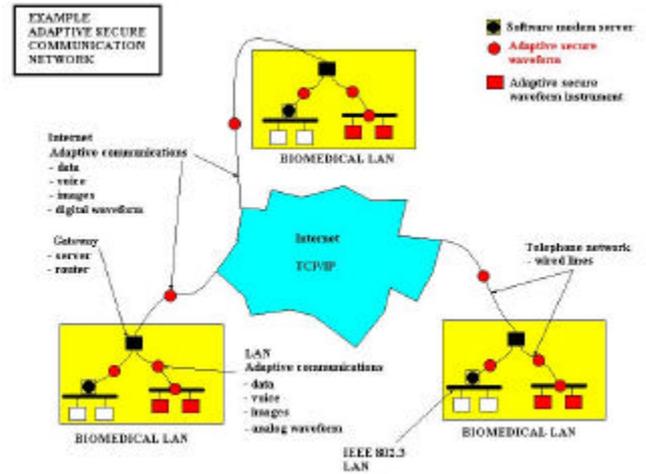


Figure 5: Adaptive Secure Network Model

Table 2: Adaptive Modem Bit Rates

Symbol Rate	QAM	Channels	MFC-code frequencies	Bit/s
1000	1*16	1	(1 1)	4000
1000	1*16	2	(5 2)	8000
2000	1*8	2	(4 2)	12000
2000	1*16	2	(5 2)	16000
2000	1*64	2	(24 2)	24000
3000	1*64	2	(17 2)	36000
3000	1*64	3	(65 2)	54000
3000	1*64	4	(513 2)	72000
3000	1*64	5	(514 3)	90000
3000	1*64	6	(285 4)	108000
3000	1*64	7	(803 4)	126000
3000	1*64	8	(640 5)	144000
3000	1*64	9	(572 6)	162000
3000	1*64	10	(541 7)	180000
3000	1*128	10	(685 8)	210000
3000	1*256	10	(836 9)	240000

In Figure 5 we have a proposal system for data transmission waveforms made with software algorithms (adaptive modem). A server in each LAN has an adaptive modem cluster (software package) that recognizes each individual instrument carrier frequency and thus the corresponding instrument. The adaptive modems are band-limited and they can be used inside the voice frequency band adapted to any type of communication channels (military radio, cellular phone, wired telephone network etc). We can connect secure waveforms over an analogue channel or over a digital voice channel (VoIP), where analog voice is coded into digital form. Thus the channels in Internet or in Public Switched Telephone Network (PSTN) are available for secure transmission on OSI level 1, Figure 5.

Passing adaptive waveforms over Internet or over telephone network to another LAN server we can build our own VPN tunnel channel with OSI level one securing. The securing is

made in the adaptive modem with frequency hopping in the voice frequency band. This can be done with the adaptive modem software algorithm in the data modulation process. Theory is presented in Milcom 2002 Tutorial (Lallo 2001-2002).

Simulation of adaptive communication methods was not possible a couple of years ago but now it is due to the present processor power and computer programming systems. Details of this securing are based on the discrete Fourier theory but are not the subject of this paper.

DISCUSSION

The channel bandwidth and the S/N are related to the channel capacity by Shannon's formula. With this metric we design wireless channel capacities (bit rates). We had earlier made a worksheet simulation process to compare modulation methods. With a selected adaptive software modem algorithm we can design an adaptive multi-carrier data communication system. We can select the proper modulation method (carriers, QAM states, secure coding, etc) and most suitable bit rate for different channels (S/N, bandwidth, etc). By traffic simulations we can check QOS of the network and we can design optimal channel capacities and select modulation methods for telemedicine data transmission networks. Using data traffic simulations we have noticed message delays and by evaluating the deviation of the delay we found design limits for data transactions. Simulation with different message sizes can give the traffic capacity limit of the backbone network (Internet, tcp/ip).

Bit errors in the received data signal begin rapidly to increase at a waveform specific S/N-limit. Data transmission needs usually a good bit error rate (BER) associated with a high S/N value. In solving this problem we can use adaptive waveforms.

SUMMARY

The subject is quite extensive. We have investigated and discussed one model of data networks with limited simulated examples, network design principles and the use of adaptive selection of the modulation method (waveform). The adaptive multi-carrier secure data communication system is an approach that can give wireless security with band-limited frequency hopping on the lowest OSI level and optimal throughput for different channels used with biomedical applications in telemedicine networks.

REFERENCES

- Andrade H.; T. Kurc; A. Sussman; and J. Saltz J. 2001. "Efficient Execution of Multiple Query Workloads in Data Analysis Applications". *Proceedings of SC2001*. Denver, Colorado, USA.
- Andrade H.; T. Kurc; A. Sussman; and J. Saltz J. 2002. "Active Proxy -G: Optimizing the Query Execution Process in the Grid". In *Supercomputing 2002*.

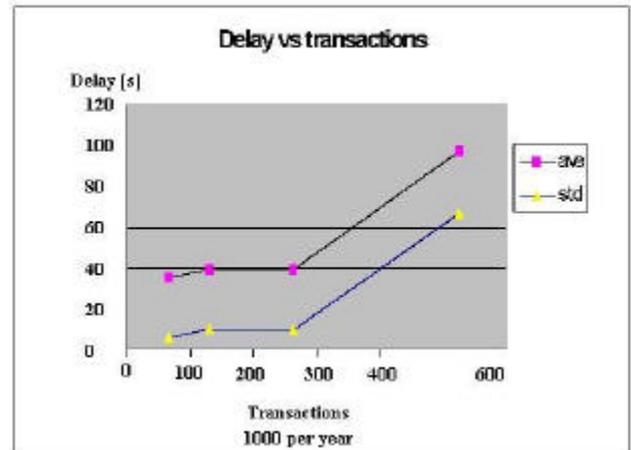


Figure 7 Message Delay vs Transactions

- Andrade H.; T. Kurc; A. Sussman; and J. Saltz J. 2002. "Multiple Query Optimization for Data Analysis Applications on Clusters of SMPs". In *IEEE International Symposium on Cluster Computing and the Grid, CCGrid2002*. Berlin, Germany.
- Andrade H.; T. Kurc; A. Sussman; and J. Saltz J. 2002. "Multiple Query Optimization Support for the Virtual Microscope". *Archives of Pathology/ and Laboratory Medicine*.
- Andrade H.; T. Kurc; A. Sussman; and J. Saltz J. 2000. "Scheduling Multiple Data Visualization Query Workloads on a Shared Memory Machine". *International Parallel and Distributed Processing Symposium, IPDPS 2002*. Fort Lauderdale, Florida.
- CACI Products Company, 1995. "COMNET III Release 1.1i. Successor to COMNET II.5 (R) and L-NET II.5 (TM)."
- Chervenak, A.; I. Foster; C. Kesselman; C. Salisbury; and S. Tuecke, S. 1999. "The Data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets."
- Foster, I. and Kesselman, C. 1999. "The GRID: Blueprint for a New Computing Infrastructure." Morgan-Kaufmann.
- Foster, I.; C. Kesselman; J.M. Nick; and S. Tuecke. 2002. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration."
- Hastings, S.; M. Gray; T. Kurc; U. Catalyurek; T. Gingrich; D. Young; and J. Saltz. 2002. "Grid Support for Collaborative Clinical and Biomedical Research Studies (Extended Abstract)", In The Advancing Pathology Informatics, Imaging, and the Internet conference (APIII 2002). Biomedical Informatics Department, The Ohio State University, Columbus, OH.
- Lallo, P.R.U., 2001. "Adaptive Software Modem Technology". *MILCOM 2001*, McLean, VA, USA.
- Lallo, P.R.U. 2002. "Tutorial T6, Adaptive Data Modulation Methods for Wireless Data Communication". *MILCOM 2002*, Anaheim, CA, USA.
- Lees P.; C. Chronaki; F. Chiarugi; E. Tsiknakis; and S. Orphanoudakis. 2002. "Interconnectivity of Cardiology Patient Databases Using Internet Technology Center for Medical Informatics and Health Telematics Applications". In *The Advancing Pathology Informatics, Imaging, and the Internet conference, ICS-FORTH*, Heraklion, Greece, APIII 2002.
- USA DoD, 2000, Small Business Innovation Research Program (SBIR) for Fiscal Year 2000.

AUTHOR BIBLIOGRAPHY

PAULI R. U. LALLO born in 1944 in Joensuu, Finland went in 1963 to the Technical University of Helsinki, where he studied telecommunications and applied electronics. He obtained M.Sc 1969 and Tech.Lic in 1975. First he has worked in Post and Telecommunication Administration (PLH) in Finland a couple of years, in Adult Education Organizations 11 years and in a small Developing Company. Between 1970-1975 and since 1988 he works in Finnish Defence Forces (FDF). His present rank is LTC in engineering and he is the Chief Scientist of the Signals and Electrotechnical School since 1.1.2003 (former Signals School).

EPOCH TASK SCHEDULING IN DISTRIBUTED SERVER SYSTEMS

Helen D. Karatza
Department of Informatics
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
Email: karatza@csd.auth.gr

KEYWORDS

Simulation, Distributed Systems, Scheduling, Performance.

ABSTRACT

In this work we study a special type of task scheduling referred to as epoch scheduling in a distributed server (processor) system. With this policy, processor queues are rearranged only at the end of predefined intervals. The time interval between successive queue rearrangements is called an *epoch*. The objective is to examine if epoch scheduling can perform close to STF method and achieve fairer service than that of STF. A simulation model is used to address performance issues associated with epoch scheduling. Simulated results indicate that epoch scheduling is a good method to use.

INTRODUCTION

Scheduling in distributed server systems has been a major research goal for many years. However, it is not always possible to efficiently execute parallel jobs. To do so, it is necessary to divide programs into tasks, assign the tasks to processors and then schedule them on distributed processors.

Many research papers exist in this research area. For example, few of them are (Abawajy and Dandamudi 2003; Dandamudi 1994; Dandamudi 2003; Harchol-Balter, et al. 2002; Harchol-Balter et al. 2003; Gong and Williamson 2003; Karatza 2000a; Karatza 2000b; Karatza 2002; Karatza and Hilzer 2003; McCann and Zahorjan 1995; Nikolopoulos and Polychronopoulos 2003; Sabin et al. 2003; Weissman, et al. 2003), and many others.

Most research into distributed system scheduling policies has focused on improving system throughput where scheduling overhead is assumed to be negligible. However, scheduling overhead can seriously degrade performance.

FCFS (First Come First Served) is the simplest scheduling method and it is fair to individual jobs but often it results in sub-optimal performance. This method results in no overhead. Many scheduling algorithms have been proposed that achieve higher performance by taking into account information about individual requests.

It is well known that STF (Shortest Task First or Shortest Time First) usually performs best but it has two disadvantages: 1) It involves a considerable amount of overhead because processor queues are rearranged each time new tasks are added. 2) It is possible to starve a task if its service time is large in comparison to the mean service time.

In this work we study epoch scheduling. With this policy, processor queues are rearranged only at the end of predefined intervals. The time interval between successive queue rearrangements is called an epoch. At the end of an epoch, the scheduler recalculates the priorities of all tasks in the system queues using the STF criterion. This type of epoch scheduling is different from epoch scheduling that is studied in (McCann and Zahorjan 1995). In their paper, only policies that provide co-scheduling are considered. Also, in the same paper all nodes are reallocated to jobs at each reallocation point.

In our work we consider that a parallel program has a simple fork-join structure. We do not consider co-scheduling. Rearrangement of queues takes place at the end of predefined intervals, instead of node reallocation to jobs. Epoch scheduling has been also studied in (Karatza 2001; Karatza 2003). The differences between this paper and each of those two papers are the following: In (Karatza 2001) a closed queuing network model is considered with a fixed number of jobs, while in this paper we consider an open queuing network model with various job arrival rates. In (Karatza 2003) the jobs are not parallel and therefore the queues contain jobs, while in this paper each job is parallel and consists of independent tasks which are assigned to queues. Therefore, (Karatza 2003) examines job scheduling, while this paper examines task scheduling.

The results of this study apply to both loosely coupled multiprocessor systems and networks of workstations connected to support parallel applications.

The objective is to study whether we can find an epoch that can perform close to STF but minimizes as much as possible the disadvantages of STF. That is we are interested to find an epoch which combines good performance and that minimizes the number of queue rearrangements and achieves fairer service than that of STF. Performance is examined for different epoch sizes.

Various workloads are examined. To our knowledge, such an analysis of epoch scheduling has not appeared in the research literature for this type of system operating with our workload models.

The technique used to evaluate the performance of the scheduling disciplines is experimentation using a synthetic workload simulation. In studies like this, it is usually necessary to use synthetic workloads because real workloads cannot be simulated efficiently enough and real systems with actual workloads are not available for experimentation. Also, useful analytic models are difficult to derive because subtleties that exist between various disciplines are difficult to model and because the workload model is quite complex.

This paper is an experimental study in that the results are obtained from simulation studies instead of from the measurements of real systems. Nevertheless, the results presented are of practical value. Although we do not derive absolute performance values for specific systems and workloads, we do study the relative performance of the different algorithms across a broad range of workloads and analyze how changes in the workload can affect performance.

For simple systems, performance models can be mathematically analyzed using queuing theory to provide performance measures. However, in the system presented in this paper, fork-join programs and scheduling policies with different complexities are involved. For such complex systems, analytical modelling typically requires additional simplifying assumptions that might have unforeseeable influence on the results. Therefore, research efforts are devoted to finding approximate methods to develop tractable models in special cases, and in conducting simulations.

The precise analysis of fork-join queuing models is a well known intractable problem. For example, (Kumar and Shorey 1993) derived upper and lower bounds for the mean response time when jobs have a linear fork-join structure.

We chose simulations because it is possible to simulate the system in a direct manner, thus lending credibility to the results. Detailed simulation models help determine performance bottlenecks in architecture and assist in refining the system configuration.

The structure of this paper is as follows: Next section specifies system and workload models, it describes scheduling strategies, and it presents the metrics employed while assessing performance of the scheduling policies. Model implementation and input parameters are described in the section after, where also experimental results and performance analysis are presented. The last section contains conclusions and suggestions for further research.

MODEL AND METHODOLOGY

System and Workload Models

This paper uses a simulation model to address scheduling performance issues. An open queuing network model of a distributed server system is considered. $P = 16$ homogeneous and independent processors are available, each serving its own queue. A high-speed network connects the distributed nodes. The configuration of the model is shown in Figure 1.

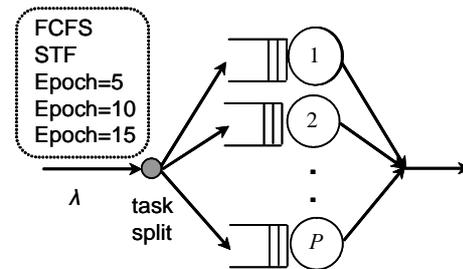


Figure 1: The Queuing Network Model

An important aspect of distributed system design is workload sharing among the processors. This includes partitioning the arriving jobs into tasks that can be executed in parallel, assigning the tasks to processors and scheduling the task execution on each processor. The workload considered here is characterized by three parameters:

- The distribution of job arrival.
- The distribution of the number of tasks per job.
- The distribution of task execution time.

We assume that there is no correlation between the different parameters. For example, a job with a small number of tasks may have a longer execution time.

Job inter-arrival times are exponential random variables with a mean of $1/\lambda$.

Jobs consist of a set of $n \geq 1$ tasks that can be run in parallel. The number of tasks that a job consists of is this job's *degree of parallelism*. It is assumed that the tasks are uniformly distributed in the range of $[1..P]$. We have chosen the uniform distribution because it is one of the distributions that are used for this kind of models. Each task is randomly assigned to a processor queue. Tasks are processed according to the current scheduling method. No migration or pre-emption is permitted.

On completing execution, a task waits at the join point for sibling tasks of the same job to complete execution. Therefore, task synchronization is required, and that synchronization can seriously degrade parallel performance. The price paid for increased parallelism is a synchronization delay that occurs when tasks wait for siblings to finish execution.

The number of tasks of a job j is represented as $t(j)$. Due to the probabilistic assignment of tasks to processor queues, more than one tasks of the same job may be assigned to the same processor. Therefore, if $p(j)$ represents the number of processors required by job j , then the following relation holds:

$$p(j) \leq t(j) \leq P$$

Job service demands are exponentially distributed with a mean of $1/\mu$.

Notation used in this paper appears in Table 1.

Job Scheduling Policies

In this work we examine only non-pre-emptive scheduling policies. We assume that the scheduler has perfect information when making decisions, i.e. it knows the execution time of tasks. Next we describe the scheduling strategies employed in this work.

First-Come-First-Served (FCFS). With this strategy, tasks are assigned to a queue in the order of their arrival. This policy is the simplest to implement.

Shortest Task (Time) First (STF). This policy assumes that a priori knowledge about a task is available in form of service demand. When such knowledge is available, tasks in the processor queues are ordered in a decreasing order of service demand.

Epoch - x: With this policy, processor queues are rearranged only at the end of predefined time intervals called epochs. The size of an epoch is x . At the end of an epoch, the scheduler recalculates the priorities of all tasks in the system queues using the STF criterion.

We study the performance of the Epoch - x policy in relation to STF method for different epoch sizes. The goal is to obtain: 1) performance comparable to that of STF, 2) large decrease in the number of queue rearrangements (small overhead), 3) large decrease in the maximum job response time (fairness in individual job service).

It should be noted that a priori information is not often available and only an estimate of task execution time can be obtained. However, it has been reported in the literature (for example in (Dandamudi 1994)) that simulation results have shown that estimation error in processor service times can marginally affect system performance.

Performance Metrics

Parameters used in simulation computations (presented later) are shown in Table 1.

Table 1: Notations

λ	Mean arrival rate of jobs
μ	Mean processor service rate
E	Estimation error in service time
U	Mean processor utilization
RT	Mean Response Time of jobs
MRT	Maximum Response Time of jobs
$Synch$	Task synchronization time
NQR	Number of Queue Rearrangements
$RT\ Ratio$	The ratio of RT when SRT or Epoch- x method is employed versus RT of the FCFS policy
$MRT\ Ratio$	The ratio of MRT when SRT or Epoch- x method is employed versus MRT of the FCFS policy
$Synch\ Ratio$	The ratio of task synchronization time when SRT or Epoch- x method is employed versus $Synch$ of the FCFS policy
$NQR\ Ratio$	The ratio of NQR when the Epoch- x method is employed versus NQR of the STF policy

RT represents overall performance, while MRT expresses fairness in individual job service.

SIMULATION RESULTS AND DISCUSSION

Model Implementation and Input Parameters

The queuing network model described above is implemented with discrete event simulation (Law and Kelton 1991) using the independent replication method. For every mean value, a 95% confidence interval is evaluated. All confidence intervals are less than 5% of the mean values.

We have chosen mean processor service time

$$1/\mu = 1,$$

which means mean service rate per processor $\mu = 1$. Since the processors average 8.5 tasks per job, when all processors are busy, an average of 1.882 jobs are served each unit of time. This implies that the mean job inter-arrival time must be larger than $1/1.882 = 0.531$ in order that the system will not be saturated. For this reason we examined the following mean inter-arrival times:

$$1/\lambda = 0.6, 0.625, 0.650, 0.675, 0.7,$$

which means mean arrival rate:

$$\lambda = 1.67, 1.6, 1.54, 1.48, 1.43.$$

Epoch length x was taken as 5, 10, 15. We chose epoch length 5 as a starting point for the experiments because the mean processor service time is equal to 1, and also because with this epoch size the number of queue rearrangements were smaller than in the STF case. Therefore we expected that larger epoch sizes would result in even smaller NQR .

Performance Analysis

Figures 2-8 present the performance metrics versus $1/\lambda$. Mean processor utilization U is 0.89, 0.85, 0.82, 0.79, 0.76, for $1/\lambda = 0.6, 0.625, 0.650, 0.675, 0.7$ respectively (Figure 2).

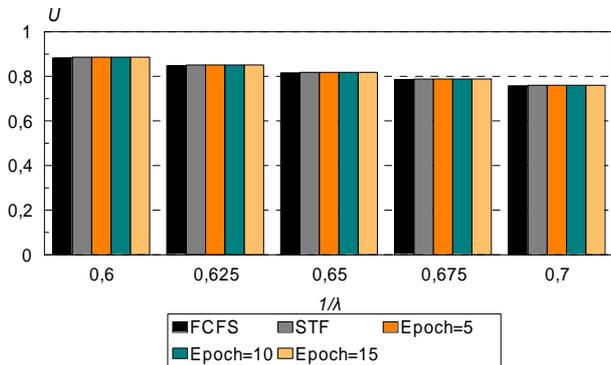


Figure 2. U versus $1/\lambda$

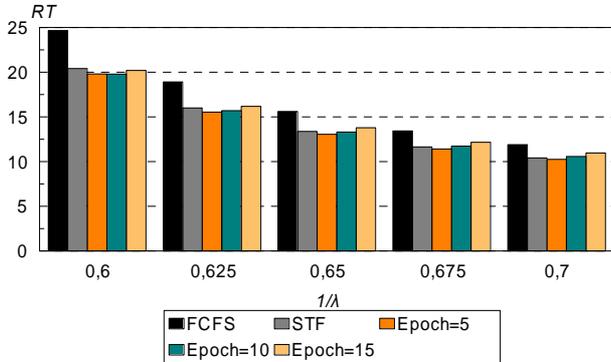


Figure 3. RT versus $1/\lambda$

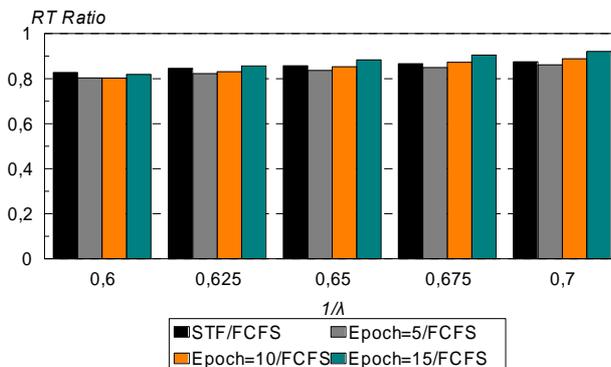


Figure 4. RT ratio versus $1/\lambda$

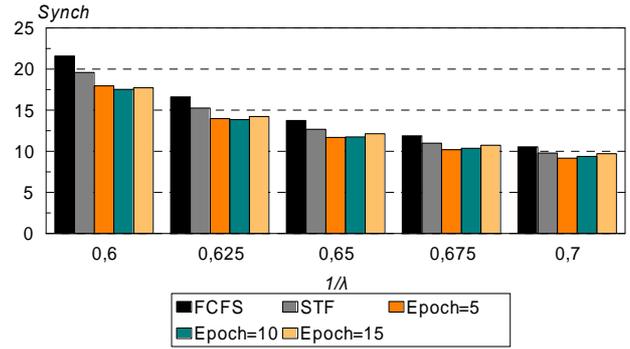


Figure 5. $Synch$ versus $1/\lambda$

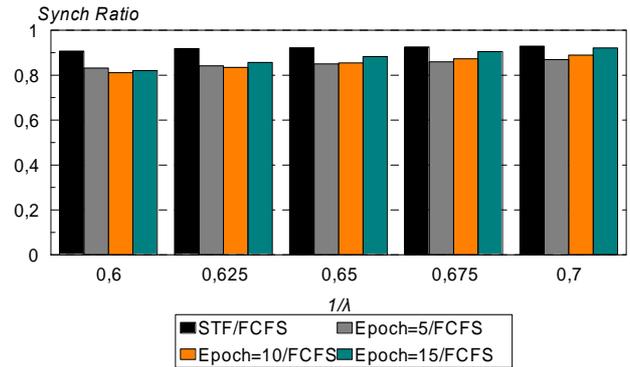


Figure 6. $Synch$ ratio versus $1/\lambda$

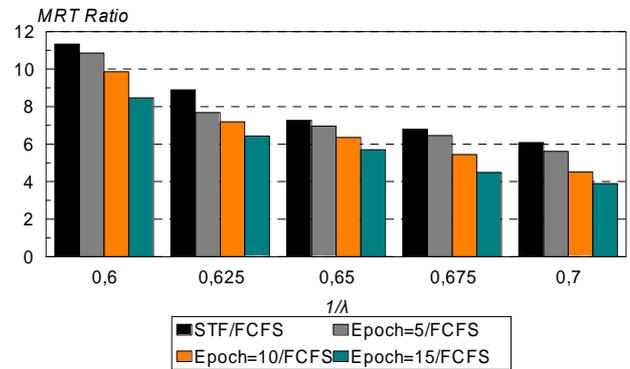


Figure 7. MRT ratio versus $1/\lambda$

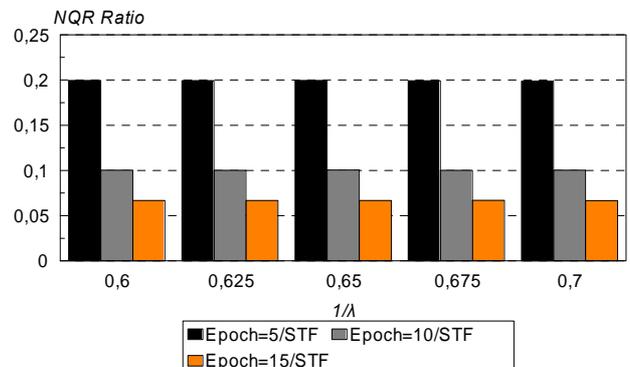


Figure 8. NQR ratio versus $1/\lambda$

In Figure 3 it is shown that from all methods examined FCFS method yields the largest response time. This is the reason in all cases RT ratio is smaller than one (Figure 4). RT ratio increases with increasing mean inter-arrival time which means that the superiority of the STF and Epoch- x methods over FCFS is larger at larger loads (that is at smaller mean inter-arrival time). This is because fewer jobs are in the queues when $1/\lambda$ is large than when is small, so there are then fewer opportunities to exploit the advantages of the STF and Epoch- x methods.

RT generally increases with increasing epoch size, but the increase is not significant. For this reason RT ratio slightly increases with increasing epoch size.

In all cases examined, epoch scheduling for epoch size 5 yields slightly smaller RT than STF. Epoch-10 yields RT which is very close to the RT of the STF case. Epoch=15 performs either close (at larger loads) or a little worst (at smaller) loads than the STF method (Figures 3, and 4). However, even with this epoch size the difference in performance between epoch scheduling and STF is not significant.

Figure 5 shows that with all scheduling methods task synchronization time is decreasing with decreasing load. This is because it is more probable at large loads than at small loads for the first task of a job that finishes execution to wait for a long time some sibling tasks that are still waiting in processor queues.

It is also shown in Figure 5 that the largest $Synch$ incurs with the FCFS method. This is because it is more probable when the FCFS policy is employed than when one of the other scheduling strategies are used, some small tasks to wait behind some large tasks in processor queues. A consequence of this may be long synchronization delay of sibling tasks. On the other hand, the STF method causes delays to large tasks, but the simulation results reveal that these delays influence $Synch$ in a smaller degree than the delays caused by the FCFS policy. $Synch$ is larger in the STF case than in the epoch scheduling case, as in the later case priority is given to small tasks on a periodic basis only.

In Figure 6 it is shown that the difference in $Synch$ between FCFS and each of the remaining methods is generally larger at larger loads. This is because regarding synchronization delay of sibling tasks, the advantages of STF and epoch scheduling over FCFS are better exploited when there are many job tasks in the system than when there are few.

In Figure 7 it is shown that in all cases the STF method yields the largest MRT , while the smallest MRT is produced by the FCFS method. In all cases, epoch scheduling yields smaller maximum response time as compared to the STF method. Therefore, in all cases

epoch scheduling is fairer than STF. Furthermore, Figure 7 shows that in all cases epoch scheduling is fairer when epoch size is large than when is small.

MRT ratio is decreasing with increasing mean inter-arrival time. This is due to the fact that when STF or epoch scheduling is employed, more job tasks go out of order for longer time when the load is large than when is small. Therefore, larger MRT can be produced at large loads than at small ones.

Figure 8 presents the NQR ratio. It is shown that for all λ the decrease in the number of queue rearrangements due to epoch scheduling is very high. Therefore significant reduction in the number of queue rearrangements can be achieved when epoch scheduling is employed instead of STF. In the same Figure is also shown that for each epoch size NQR ratio is nearly the same for all values of λ . The NQR ratio is about 0.2, 0.10, and 0.07 in the Epoch=5, 10, and 15 cases respectively. For each λ , NQR ratio decreases with increasing epoch size which means that NQR is smaller when epoch size is large than when is small.

In order to study the impact of service time estimation error on the performance of the STF and Epoch- x methods, additional simulation experiments were conducted. In those experiments task execution time estimated was assumed to be uniformly distributed within $\pm E\%$ of the exact value. We set estimation error at $\pm 10\%$. The results showed that the estimation error did not significantly affect performance. This is in accordance with other results in the literature related to estimation of service time (Dandamudi 1994). For this reason in this paper we present the results for the exact service times only. This means that we consider estimation error set at $\pm 0\%$.

CONCLUSIONS AND RECOMMENDED FUTURE RESEARCH

This paper studies task scheduling policies in a distributed server system. Simulation is used to generate results used to compare different configurations.

Simulation results reveal that epoch task scheduling is a good policy to choose. For the epoch lengths that we examined this policy performs very close to STF. Furthermore and more importantly, it involves much less overhead and is also fairer than STF.

It is also shown that all of the epoch lengths have merit. For all loads that we examined large epochs result in fairer service for individual jobs, and involve less overhead than short epochs. On the other hand, response time is shorter with short epochs. However, for the epoch lengths that we examined, response time does not differ significantly at different epochs.

This paper represents a case study where the number of tasks per job is bounded by the number of distributed

servers in the system (the uniform distribution is used). As a future research we plan to consider the exponential distribution for the number of tasks per job, so that we can study the performance of epoch scheduling when the number of job tasks can be larger than the number of processors.

REFERENCES

- Abawajy, J.H. and S. Dandamudi. 2003. "Scheduling Parallel Jobs with CPU and I/O Resource Requirements in Cluster Computing Systems". In *Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (Mascots'03)* (Orlando, FL, Oct.12-14). IEEE Computer Society, Los Alamitos, CA, 336-343.
- Dandamudi, S. 1994. "Performance Implications of Task Routing and Task Scheduling Strategies for Multiprocessor Systems". In *Proceedings of the IEEE-Euromicro Conference on Massively Parallel Computing Systems* (Ischia, Italy, May 2-6). IEEE Computer Society, Los Alamitos, CA, 348-353.
- Dandamudi, S. 2003. *Hierarchical Scheduling in Parallel and Cluster Systems*. 1st edn. Kluwer Academic/Plenum Publishers, New York.
- Harchol-Balter, M.; K. Sigman; and A. Wierman. 2002. "Asymptotic Convergence of Scheduling Policies with Respect to Slowdown". In *Proceedings of IFIP Performance 2002* (Rome, Italy, Sept.22-27). *Performance Evaluation* 49, Elsevier B.V., Amsterdam, The Netherlands, 241-256.
- Harchol-Balter, M.; B. Schroeder; N. Bansal; and M. Agrawal. 2003. "Size-based Scheduling to Improve Web Performance". *ACM Transactions on Computer Systems* 21, No.2, Association for Computing Machinery, New York, N.Y., 1-27.
- Gong, M. and C. Williamson. 2003. "Quantifying the Properties of SRPT Scheduling". In *Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (Mascots'03)* (Orlando, FL, Oct.12-14). IEEE Computer Society, Los Alamitos, CA, 126-135.
- Karatza, H.D. 2000a. "Scheduling Strategies for Multitasking in a Distributed System". in *Proceedings of the 33rd Annual Simulation Symposium* (Washington, D.C., Apr.16-20). IEEE Computer Society, Los Alamitos, CA, pp. 83-90.
- Karatza, H.D. 2000b. "A Comparative Analysis of Scheduling Policies in a Distributed System using Simulation". *International Journal of Simulation: Systems, Science & Technology* 1, No.1-2 (Dec.), UK Simulation Society, 12-20.
- Karatza, H.D. 2001. "Epoch Scheduling in a Distributed System". In *Proceedings of the Eurosim 2001 Congress*, (Delft, Netherlands, Jun. 26-19). Eurosim, Delft, Netherlands, 1-6.
- Karatza, H.D. 2002. "Task Scheduling Performance in Distributed Systems with Time Varying Workload". *Neural, Parallel & Scientific Computations* 10, No. 3, Dynamic Publishers, Atlanta, GA, 325-338.
- Karatza, H.D. 2003. "A Comparison of Load Sharing and Job Scheduling in a Network of Workstations". *International Journal of Simulation: Systems, Science Technology* 4, No. 3&4, UK Simulation Society, Nottingham, UK, 4-11.
- Karatza, H.D. and R.C. Hilzer. 2003. "Parallel Job Scheduling in Distributed Systems". *Simulation: Transactions of the Society for Modeling and Simulation International* 79, No.5 (May), Sage Publications, Thousand Oaks, CA, 287-298.
- Kumar A. and R. Shorey. 1993. "Performance Analysis and Scheduling of Stochastic Fork-Join Jobs in a Multicomputer System". *IEEE Transactions on Parallel and Distributed Systems* 4, No.10, IEEE Piscataway, NJ, 1147-1162.
- McCann, C. and J. Zahorjan. 1995. "Scheduling Memory Constrained Jobs on Distributed Memory Parallel Computers". In *Proceedings of the 1995 ACM Sigmetrics Conference* (Ottawa, Canada, May 15-19). The Association for Computing Machinery, New York, 208-219.
- Law A. and D. Kelton. 1991. *Simulation Modelling and Analysis*. 2nd Ed., McGraw-Hill, New York, USA.
- Nikolopoulos, D.S. and C.D. Polychronopoulos. 2003. "Adaptive Scheduling Under Memory Constraints on Non-Dedicated Computational Farms". *Future Generation Computer Systems* 19, Elsevier, Amsterdam, 505-519.
- Sabin, G.; R. Kettimuthu; A. Rajan; and P. Sadayappan. 2003. "Scheduling of Parallel Jobs in a Heterogeneous Multi-Site Environment". In *Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science* 2862, D. Feitelson, L. Rudolph, and W. Schwiegelshohn (Eds.). Springer-Verlag, Berlin Heidelberg, 87-104.
- Weissman, J.B.; L.R. Abburi; and D. England. 2003. "Integrated Scheduling: the Best of Both Worlds". *Journal of Parallel and Distributed Computing* 63, Elsevier Science, New York, USA, 649-668.

AUTHOR BIOGRAPHY



HELEN D. KARATZA is an Associate Professor in the Department of Informatics at the Aristotle University of Thessaloniki, Greece. Her research interests include Computer Systems Performance Evaluation, Parallel and Distributed Systems Scheduling, and Simulation. Dr. Karatza is an Associate Editor of *SIMULATION: Transactions of the Society for Modeling and Simulation International*, and a member of the Editorial Board of the *International Journal of Simulation: Systems, Science & Technology*, and *Simulation Modelling Practice and Theory Journal*. Her email address is: karatza@csd.uth.gr and her Web-page can be found at <http://www.csd.auth.gr/~karatza>.

INTELLIGENT SYSTEMS

THE ANALYSIS OF NETWORK MANAGERS' BEHAVIOUR USING A SELF-ORGANISING NEURAL NETWORK

Helen Donelan and Colin Pattinson
Computer Communications Research Group
Leeds Metropolitan University
Beckett Park, Leeds, LS6 3QS, UK
E-mail: H.Donelan@leedsmet.ac.uk

Dominic Palmer-Brown and Sin Wee Lee
Computational Intelligence Research Group
Leeds Metropolitan University
Beckett Park, Leeds, LS6 3QS, UK

KEYWORDS

Network Management, Neural Networks, Data Analysis.

ABSTRACT

This paper presents a novel method for the analysis and interpretation of data that describes the interaction between trainee network managers and a simulated network management system. A simulation based approach to the task of efficiently training network managers, through the use of a simulated network, was originally presented by Pattinson (2000). The motivation was to provide a tool for exposing trainee network managers to a life like situation, where both normal network operation and 'fault' scenarios could be simulated in order to train the network manager. The data logged by this system describes the detailed interaction between trainee network manager and simulated network. The work presented here provides an analysis of this interaction data that enables an assessment of the capabilities of the network manager as well as an understanding of how the network management tasks are being approached. A neural network architecture (Lee et al. 2002) is adapted and implemented in order to perform an exploratory data analysis of the interaction data. The neural network architecture employs a novel form of continuous self-organisation to discover key features, and thus provide new insights into the data.

INTRODUCTION

A simulation based approach to the task of efficiently training network managers through the use of a simulated network was originally presented by (Pattinson 2000). The motivation was to provide trainee network managers with realistic, 'hands-on' experience without disrupting a live network. The approach makes use of a production-standard network management platform, interacting with processes (model agents) representing network entities. This simulation tool has been successfully used in the training of network managers. Tasks are set such as exploration exercises to identify active components of a network and the control of simulated 'fault' conditions. The trainee needs to quickly establish how to approach a given task, which devices need to be interrogated and what parameters need to be monitored

in order to obtain information on the status of the network. The trainee selects commands that to the best of their knowledge represent the most appropriate course of action required to manage the network and data describing their actions is computer logged. It is this data that is being analysed here through the use of a neural network (NN) in order to assess the effectiveness of the network manager from how the tasks are being approached. The data includes a description of the commands issued by the trainee, the device within the network that the command is directed at, any associated variables, and a date and time stamp. The commands are used to request current values of, or set up processes to monitor, various parameters and are divided into groups defined by the layer (or networking protocol) that they apply to.

The computer logged data contains both qualitative and quantitative data and therefore provides a challenge in how to approach its processing, analysis and evaluation. Qualitative data tends to be voluminous and inconsistent, and has many problems associated with its analysis, as discussed by Miles and Huberan (1994). The analysis of such data typically involves sifting through the data and noting regularly occurring relationships between variables. The tasks of coding, isolating and interpreting patterns of interest can be long and arduous and relies on the analyst being able to comprehensively identify all patterns that are of interest. The concept of using a NN in the analysis means that the task of identifying patterns within the data is relinquished to the NN. This presents an opportunity to find hidden patterns and establish relationships between variables that are extremely difficult, if not impossible to discover by human eye or thought process and that therefore offer new information about the data. An unsupervised technique has been adopted as this requires no a priori knowledge about the data. Classes or categories are formed by the NN according to attributes of the data and it is then necessary to uncover what properties determine how the data has been classified.

Neural Networks have previously been applied to data types that face similar challenges to those faced here. For example Shalvi and DeClariss (1998) successfully demonstrated the use of self-organising maps, an unsupervised NN approach, to cluster medical data. Medical data typically requires a large amount of pre-

processing in order to extract the useful information and tends to be numeric and textual interspersed. In these ways the data is very similar to that being examined here. There have been many projects involving NNs for user data analysis and pattern discovery. For example, Zhang et al (2001) used NNs for learning relations between textual data to aid the construction of hypertext computer assisted learning material; and Mullier et al (2002) used them for identifying hypermedia browsing patterns.

THE NEURAL NETWORK

Developments of Adaptive Resonance Theory(ART)

In recent years, several variations of the original ART (Grossberg 1976) have been introduced. ART1 (Carpenter and Grossberg 1987a) self-organises recognition categories for arbitrary sequences of binary input sequences; ART2, operates on either binary or analogue inputs (Carpenter and Grossberg 1987b). Further development has seen the creation of ART2-A (Carpenter et al. 1991a), which is 2 or 3 orders of magnitude faster than ART2. Fuzzy ART (Carpenter et al. 1991b), incorporated computations from fuzzy set theory. Extensions to ART networks to allow supervised learning, supervised multi-layer, and self-growing systems (Palmer-Brown 1992; Tan 1997) have also been introduced.

Performance-guided ART (P-ART) Architecture

The P-ART network is a modular, multi-layered architecture as shown in Fig. 1.

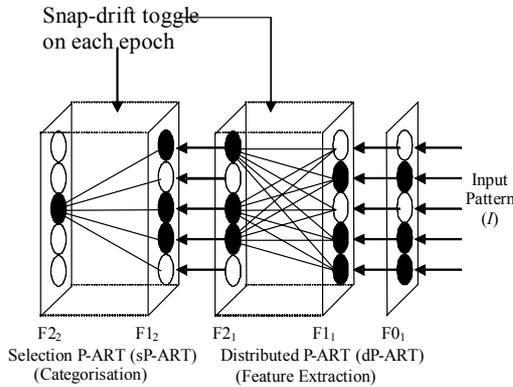


Figure 1: P-ART Network

It is composed of 2 modules, a Distributed P-ART (dP-ART) network, and a Selection P-ART (sP-ART). The $F1_1 \leftrightarrow F2_1$ connections of the dP-ART network and $F1_2 \leftrightarrow F2_2$ of the sP-ART are interconnected through weighted bottom-up and top-down connections that can be modified during the learning stage. For clarity, only the connections from the F1 layer to the active (winning) F2 node in each P-ART module are shown. On presentation of an input pattern at the input layer $F0_1$, the dP-ART will learn to group the input patterns according to their general features using the novel

learning principles of the snap-drift algorithm recently developed (Lee et al 2002; 2003). The new version used here is fully self-organising, toggling between snap and drift learning modes on successive epochs.

The 'Snap-Drift' Algorithm

In an environment where new patterns are introduced over time, the learning utilises a novel snap-drift algorithm based on fast, convergent, minimalist learning (snap) and cautious learning (drift). Snap is based on a modified form of ART; and drift is based on Learning Vector Quantization (LVQ) (Kohonen 1990). In general terms, the snap-drift algorithm can be stated as:

$$w = \alpha(\text{Fast_Learning_ART}) + \sigma(\text{LVQ})$$

In this paper, α and σ are toggled between (0,1) and (1,0) at the end of each epoch. The point of this is to perform two complementary forms of feature discovery within one system. The ART style learning acquires features characterized by the intersection of patterns, whereas LVQ performs clustering, discovering features that are averaged across patterns.

The Distributed P-ART (dP-ART) Learning

On presentation of an input pattern, the bottom-up activation is calculated. Then the D $F2_1$ nodes with the highest bottom-up activation are selected:

$$T_J = \max\{T_J \mid J = 1, 2, \dots, M\}$$

D is set to 3 in this application. The three $F2_1$ nodes learn according to:

$$w_{ji}^{(new)} = \alpha(I \cap w_{ji}^{(old)}) + \sigma(w_{ji}^{(old)} + \beta(I - w_{ji}^{(old)}))$$

where w_{ji} = top-down weights vectors; I = binary input vectors, and β = the drift speed constant = 0.5. When $\alpha = 1$, w updates simply to:

$$w_{ji}^{(new)} = (I \cap w_{ji}^{(old)})$$

This invokes fast minimalist learning, causing the top-down weights to reach their new asymptote on each input presentation:

$$w_J \rightarrow I \cap w_J^{(old)}$$

In contrast, when $\sigma = 1$:

$$w_{ji}^{(new)} = (w_{ji}^{(old)} + \beta(I - w_{ji}^{(old)}))$$

This causes a simple form of clustering or LVQ at a speed determined by β . Overall the learning is a combination of the two forms of adaptation as the

mode is toggled between snap and drift. The novel bottom-up learning of the P-ART is a normalised version of the top-down learning:

$$w_{ji}^{(new)} = \frac{w_{ij}^{(new)}}{|w_{ij}^{(new)}|}$$

where $w_{ij}^{(new)}$ = top-down weights of the network after learning.

METHODOLOGY

Although the specific aim here is to provide a novel interpretation of the interaction data described, a general aim is to devise a methodology that can be applied to any ‘interaction’ or ‘user behaviour’ data, and minimises the pre-and post-processing through the definition of a structured approach. There are several considerations that need to be given to the method and the stages of transformation the data must undergo before it is suitable as an input to a NN. A generalised methodology has been developed and realised in the form of a set of procedures that embrace the following processes: (i) Pre-processing: conversion of the ‘raw user data’ into a form suitable for input to the NN; (ii) Selection of optimal NN parameters; (iii) Post-processing: manipulation of the results in order to provide a novel and intelligent analysis. Stages (i) and (ii) are summarised in relation to the network management data below and Stage (iii) is discussed in the results section.

Data Collection and Initial Assessment

There are some fundamental issues regarding the data that need to be addressed initially such as: How much data is available? Is there a constant output of data? Is there sufficient data to adequately train and test the network? The performance of a NN is dependent on the training data. The training data must be representative of the task being learnt and tends to be chosen through trial and error before finding an acceptable training data set (Callan 1999). The production of the network management data is ongoing but for the purposes of this paper a sample has been used to provide an initial evaluation of the data. 55 datasets are used, where a single dataset represents a trainee undertaking a single network management task.

Structure and Context

The next consideration in pre-processing the data is how to *structure* the encoded data in order to ensure all appropriate variables contribute towards the NN’s decision making and is in a form suitable for input to a NN. The data is initially primary encoded into a list of consecutive events, where an *event* comprises all the information required to describe what is taking place at a given instant in time. The aim is to ensure that each event comprises the same amount of information in order to introduce some structure to the data. The

network management data is already in a format that lists events, it simply needs condensing into the relevant information. This takes the form: $\{Action - Node Description - Duration\}$. *Action* describes the command issued by the student, *Node Description* refers to which device the command relates to and whether it is active or inactive and *Duration* is the time taken to issue the command. Determining the structure of an event is the first level of encoding and the resulting event is called the primary encoded data. This process reduces the feature space and simplifies the data therefore care must be taken to ensure relevant information is not excluded.

Another consideration is how to interpret the context of the data. Although the data is in a serial format, relationships between consecutive and subsequent events may be an important factor that needs to be considered. When the NN receives an input vector, it compares it to previously stored input patterns and then either puts it into the class that most closely matches it, or if no such class exists, creates a new one. Any patterns that are being identified within the data are so across input vectors. Therefore the length and quantity of data within each input vector is extremely important. The investigation requires several stages where each stage can be described in terms of the length of the input vector, which is a multiple of events (1xEvent, 2xEvent etc.). The first stage is simply concerned with the occurrence of individual events, i.e. an input vector is equal to 1 event. Whilst this provides information on the significance of each of the different events within a network management session, no information is afforded on the context of events. The second stage of investigations tackles the contextual aspect of the data. An input vector presented to the NN comprises 2 or more consecutive events. Consecutive input vectors comprise overlapping events as illustrated in the following equation, where i_n represents the n -th input vector and E_n the n -th event. $[E_n E_{n+1}]$ represents the concatenation of 2 consecutive events.

$$i_{n-1} = [E_{n-1} E_n], i_n = [E_n E_{n+1}], i_{n+1} = [E_{n+1} E_{n+2}], \text{ etc.}$$

Coding the Network Management Data

Once the structure of an event has been established, each component that makes up the event can be individually coded and the coded components concatenated to form the overall input vector. It is necessary to know how many different values each of these components may take to facilitate the implementation of an appropriate coding scheme. Examples of the coding scheme for the *Action* component are illustrated in Table 1 in terms of the possible actions, their primary encoded form, and the final coded format (‘O’ represents binary ‘1’ and ‘X’ binary ‘0’). The first segment of the codeword defines which category or layer the action belongs to, such as IP or TCP. The second segment distinguishes between the different actions within a category.

Table 1: Encoded *Action* Component of an Event

	Name (Primary Encoding)	Secondary Encoding	
1	System Info(SI)	OXXXXXXXXX	XXXXXXXXXX
2	IF Status(IF St)	XOXXXXXXXXX	OXXXXXXXXX
3	IF Parameter(IF P)	XOXXXXXXXXX	XOXXXXXXXXX
4	IF Usage(IF Usq)	XOXXXXXXXXX	XXOXXXXX
5	IF Error(IF Err)	XOXXXXXXXXX	XXXOXXXX
6	IF Quality(IF Q)	XOXXXXXXXXX	XXXOXXXX
7	IP Stats(IP Stat)	XXOXXXXX	XXXXXOX
8	IP Addresses(IP A)	XXOXXXXX	XXXXXOX
....
22	Start Monitor(Smon)	XXXXXXXXXO	XXXXXOX
23	Help (Help)	XXXXXXXXXO	XXXXXOX

The second component of an event is the *Node Description*. Again, the codeword is segmented. The first part indicates whether the node is active (O) or inactive (X) and the second part is used to distinguish between the active nodes.

Table 2: Encoded *Node Description* Component of an Event

Model	Primary Encoding		Secondary Encoding
192.9.200.1	1	1	O : OXXXXXXXXXXXXXXXXX
192.9.200.2	1	2	O : XOXXXXXXXXXXXXXXXXX
192.9.200.3	1	3	O : XXOXXXXXXXXXXXXXXXXX
192.9.200.4	1	4	O : XXXOXXXXXXXXXXXXXXXXX
.....
194.9.177.66	1	14	O : XXXXXXXXXXXXXXXXXXXXO
Inactive node	0	0	X : XXXXXXXXXXXXXXXXXXXX

Finally, the third component is the *Duration*. This is useful as it gives an indication of the time taken to execute a command. A coarse coding scheme (Eurich et al. 1997) is used, where neighbouring sub-divisions are allocated codewords that differ from each other by 1 bit position.

Table 3: Encoded *Duration* Component of an Event

Duration	Primary Encoding	Secondary Encoding
0:00:00	t_0	XXXXXX
0:01:08	t_1	XXXXXO
0:02:17	t_2	XXXOOO
0:03:25	t_3	XXXOX
.....
0:35:21	t_{31}	OXXXXX

Neural Network Parameters

There are several parameters concerning the dimensions of the NN and the level of categorisation. The NN architecture has been presented in detail elsewhere (Lee et al. 2002). In this paper, the vigilance parameter is set to 0.3, which in effect means that the criterion for allowing a node to respond and learn the input pattern is a 30% match. The number of input neurons to the dP-ART is determined by the length of the input vector. The number of output neurons for the dP-ART and therefore the input neurons the sP-ART, has been set at 500 which has proved large enough to

avoid saturation of all the output neurons of the dP-ART. The number of output neurons of the sP-ART has been set at 150 in order to limit the number of output classes that can be formed to a manageable number and yet ensure that the majority (approximately 95%) of inputs are classified.

RESULTS

The P-ART is implemented in C++. The inputs are binary vectors and the output is a corresponding column of numbers indicating the output class. A total of 2700 input vectors are input to the NN and evaluated here. Actions are referred to by number or primary encoded format. These and their basic functions are summarised in Table 4 below.

Table 4: Summary of Actions

	Primary Encoding	Description of functionality
1	SI	Display information of the system group
2	IF St	Display information of the interfaces
3	IF P	Display interface parameter like speed
4	IF Usq	Display interface statistics
5	IF Err	Display interface statistics
6	IF Q	Show error and discard rate for each interface
7	IP Stat	Display statistics and parameters of IP layer
8	IP A	Show IP addresses used by this device
9	IP R	Display routing table
10	IP ARP	Display other devices its been in contact with
11	TCP Stat	Display statistics and parameters of TCP layer
12	TCP C	Display status of existing TCP connections
13	UDP Stat	Display statistics and parameters of UDP layer
14	UDP L	Display status of existing USP listener
15	ICMP Stat	Display stats and parameters of ICMP layer
16	SNMP Stat	Display stats and parameters of SNMP layer
17	W	Walk through MIB tree and print object values
18	SetP	Set SNMP parameters
19	monV	Monitor an SNMP variable in a stripchart
20	Del	Delete the monitoring process
21	Lmon	Set up monitoring process
22	Smon	Start monitoring process
23	Help	List choice of actions

Firstly, it is necessary to assess the number and size of the output classes that have been formed, i.e. how many output nodes of the NN are used and how many input vectors are assigned to each of these. The results are easily rearranged, grouped and manipulated in order to make different comparisons. Results are reorganized and displayed by output node in order to visualise the input vectors clustered within each output class. The second stage is to establish which of the classes formed are major, significant and minority classes. Major classes are the most commonly used classes, significant classes are smaller but still populated sufficiently have an impact on conclusions, whereas minority classes are those that are only used once or twice and have little impact so are disregarded. Once the results are grouped by output class the key or dominant features of each group are established. A dominant feature is defined as a feature common to over 90% of the input vectors within that class. It may take a single value or a group of values, e.g. action 1 (SI) or actions 2-6 (all IF type actions). A class maybe defined by a single dominant feature, such as an action,

or by several, such as action and node. Where the dominant features are discussed the terminology {*action ; node ; duration*} is adopted. For example { 12 ; 1/- ; - } represents an output class where a majority of the input vectors within this class are action 12 (TCP C) implemented on an unspecific active node.

1xEvent

These results provide information on the significance of events. Table 5 summarises some of the classes that have an action or group of actions as a dominant feature along with their size (combined size where more than one class exists with the same feature).

Table 5: Dominant Features for 1xEvent Results

Action	Number of classes	Class size
1 (SI)	4	813
2 (IF St)	2	49
3 (IF P)	1	47
6 (IF Q)	1	10
8 (IP A)	2	153
9 (IP R)	1	169
10 (IP ARP)	3	181
12 (TCP C)	2	148
21 (Lmon)	1	164
17-23 (monitor actions)	5	146
2-6 (IF actions)	2	200
11-16 (TCP, UDP, ICMP and SNMP)	1	13

Action 1 (SI) is by far the most frequently used action. This is expected as this is the conventional method by which to obtain standard information regarding the network devices. Action 21 (Lmon) is also a common action which is also expected as this enables the monitoring of interface (network card) loads and is a common networking requirement. In respect to types of actions, it is the IP actions that are the most commonly applied, which implies a good use of commands. IP layer actions and action 12 (TCP C) reflect requirements to determine network topology through address structure and are encouraged methods of exploration. The groups of actions that appear as dominant features, include monitoring actions, IF actions and much less commonly, actions 11-16 (TCP, UDP, ICMP and SNMP layer actions). This illustrates a much more occasional use of these types of commands compared to other layers.

Due to the way the NN has grouped certain inputs it is possible to compare classes that feature actions directed at an inactive node with those directed at active nodes. For example, the four output classes associated with action 1 (SI) are summarised in terms of their dominant features and size in Table 6.

Table 6: Major Output Classes Featuring 'SI'

Output Class	Dominant Features	Class Size
5	{ 1 ; 1/9 ; 1-5 }	64
14	{ 1 ; 0/0 ; 1 }	215
16	{ 1 ; 1/- ; 1-7 }	455
28	{ 1 ; 0/0 ; - }	79

Classes 5 and 16 illustrate action 1 applied to an active node within the network and have a combined size of 519. Classes 14 and 28 illustrate the same action but applied to an inactive node, and have a combined size of 294. The latter two classes imply inefficient practice or limited knowledge of the network structure and make up 36% of the total number of occurrences of this command. Similarly, for classes featuring action 8 (IP A) as an individual dominant feature, those featuring this action applied to an inactive node makes up 27% of the total. However for action 10 (IP ARP) the percentage of cases this action is applied to an inactive node, is only 9%.

2xEvent and 3xEvent

Where as the 1xEvent results are useful to determine the frequency of specific events, the 2xEvent and 3xEvent results can be used to identify relationships between consecutive events.

A strong relationship exists between action 1 (SI), and the monitoring actions. Several output classes (combined size of 241) have been created that illustrate this action both preceding and following a monitoring action. Another interesting observation is the formation of several output classes that feature inactive nodes as dominant features in consecutive events. For the 2xEvent results, four output classes are formed that feature an inactive node in both events. One of these is a major class (size 105) and contains instances when the SI command is repeatedly implemented on an inactive node. This behaviour implies an ineffective use of the SI command, both due to its repetition and it being directed at an inactive node. Major classes where inactive nodes appear as dominant features in both events of an episode make up 7% of the overall major classes for the 2xEvent results. Extending this investigation to the 3xEvent results to determine how common it is that three consecutive events feature an inactive node, it is seen that four classes are formed. One is of a significant size (33) and again shows repeated use of the SI command on an inactive node.

As expected from the 1xEvent results, the SI command features most prominently. For the 2xEvent results it is a dominant feature of four out of the fourteen major classes (45%) and is a dominant feature of both events in three of these.

For the 3xEvent results, many of the classes formed are not defined by multiple specific dominant features. This is because although the input vector length has increased the vigilance parameter has remained the same in order to encourage a more generalised clustering of the input patterns. One interesting result is the output classes that highlight events that follow action 1 (SI) – i.e. what the trainee manager does once basic system information has been obtained. The most popular course of action following SI is a repetition of the same action (combined size 165). The network management simulator has a functionality that allows the selection of multiple nodes and the application of a single action to each of the nodes selected. Whilst this feature exists, it is not an efficient method for collating information on the network as redundant information is gathered and therefore has implications on the bandwidth required due to unnecessary network traffic being generated. The output classes discussed here, with the repetition of the command over 3 consecutive events, implies the use of this simulator feature, which in turn implies lack of consideration to the way in which the exploration of the network is conducted. In comparison, the preferred course of action to follow the SI action is the use of IP layer commands to provide a more thorough and yet directed interrogation of network devices. This does appear as a dominant feature, but less frequently than the repeated use of SI (combined size 49).

CONCLUSIONS

A novel method for the analysis of interactions between a network manager and a network management training platform has been presented. The method can be used to uncover hidden patterns in user behaviour and therefore provide novel insights into that behaviour. The output classes formed by the NN can be used to compare instances of good and bad practice and reveal patterns embedded within the data that are difficult to recognize through other methods. The results identified both commonly occurring combinations of events and other interesting, though less common, sequences of events. Whilst a great deal of information has been accumulated in relation to the commands or actions performed by the network manager as well as the nodes within the network that these actions are directed towards, little information has been gleaned on the duration of these events. A reason for this may be that a large proportion of the *durations* fall within the first few increments of the overall range of values (i.e. t_0 and t_1). It may be beneficial to re-address the coding scheme applied to this component of an event for it to be more influential in the forming of the output classes.

A proposed development of the project intends to incorporate a more integrated pre-processing operation through the automatic creation of primary encoded data files at source. It is also intended to develop an on-line feedback system that responds to real-time operation of

the simulated network with critiques in response to the approaches chosen by the trainee network manager.

REFERENCES

- Callan, R. 1999. *The Essence of Neural Networks*. Prentice Hall, Europe.
- Carpenter, G. A. and S. Grossberg. 1987. "A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition Machine," *Computer Vision, Graphics and Image Processing*, Vol. 37, 54-115.
- Carpenter, G. A. and S. Grossberg, 1987. "ART2: Self-Organization of Stable Category Recognition Codes for Analogue Pattern," *Applied Optics*," Vol. 26, 4919 -4930.
- Carpenter, G. A.; S. Grossberg and D.B. Rosen. 1991. "ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition," *Neural Networks*, Vol. 4, 493 - 504.
- Carpenter, G. A.; S. Grossberg, and D. B. Rosen. 1991. "Fuzzy ART: Fast Stable Learning and Categorization of Analogue Pattern by an Adaptive Resonance System," *Neural Networks*, Vol. 4, 759 - 771.
- Eurich, C.W.; H. Schwegler, and R. Woesler, "Coarse Coding: Applications to the Visual System of Salamanders." *Biol. Cybern.*, Vol. 77, 1997, pp. 41-47.
- Grossberg, S. 1976. "Adaptive Pattern Classification and Universal Recoding. I. Parallel Development and Coding of Neural Feature Detectors," *Biol. Cybern.*, Vol. 23, 121 - 134.
- Kohonen, T. 1990. "Improved versions of learning vector quantization", In *Proceedings of Int. Joint Conf. Neural Networks*, Vol. 1. 545-550.
- Lee, S.W.; D. Palmer-Brown; J. Tepper; C. Roadknight. 2002. "Performance-guided Neural Network for Rapidly Self-Organising Active Network Management." In *Soft Computing Systems: Design Management and Applications*, A. Abraham et al (Eds.), IOS Press, 22-31.
- Lee, S. W.; D. Palmer-Brown; J. Tepper and C. M. Roadknight. 2003. "Snap-Drift: Real-time Performance-guided Learning," in *Proceedings of Int. Joint Conf. Neural Networks*, Vol. 2, 1412 – 1416.
- Miles, M.B. and A.M. Huberman. 1994. *Qualitative Data Analysis, Second Edition*. SAGE Publications.
- Mullier, D. J.; D. J. Hobbs and D. J. Moore. 2002. "Identifying and Using Hypermedia Browsing Patterns." *Journal of Educational Multimedia and Hypermedia*, No.11 (1), 31-50.
- Palmer-Brown, D. 1992. "High Speed Learning in a Supervised, Self Growing Net," in *Proceedings of Int. Conf. on Artificial Neural Networks*, Vol. 2, 1159-1162.
- Pattinson, C. 2000. "A simulated network management information base." *Journal of Network and Computer Applications*, 23, 93-701.
- Shalvi, D. and N. DeClaris. 1998. "An Unsupervised Neural Network Approach to Medical Data Mining Techniques." In *Proceedings of IEEE Int. Joint Conf. Neural Networks*, 171-176.
- Tan, A. H. 1997. "Cascade ARTMAP: Integrating Neural Computation and Symbolic Knowledge Processing," *IEEE Trans. Neural Networks*, Vol. 8, No. 2, 237 – 250.
- Zhang, S.; H. Powell and D. Palmer-Brown 2001. "Methods for Concept Extraction using ANNs and Stemming Analysis and Their Portability Across Domains." In *Proceeding of The 2nd Workshop on Natural Language Processing and Neural Network* (Tokyo, Japan), 62 - 79.

A COMPUTATIONAL MODEL OF ACUTE PAIN

Karen Prince, Jackie Campbell, Phil Picton, Scott Turner
Centre for Healthcare Education
University College Northampton
Boughton Green Road, Northampton
England, NN2 7AL
E-mail: k.m.fulstow@btinternet.com

KEYWORDS

Acute pain, Mathematical modelling, Gate control theory, Interneurons, T cells.

ABSTRACT

In 1965 Melzack and Wall proposed the influential gate control theory of pain. This theory postulates that the substantia gelatinosa, located within the spinal cord, acts as a gate control mechanism which can influence the flow of information to the brain and thus impact on the pain experience. Subsequent research has, in general, supported this theory. The theory presented is very explicit and the fact that pain is a poorly understood phenomenon suggests it is an ideal candidate for modelling. Despite this, the utilisation of such techniques has been very limited. This paper successfully replicates the mathematical model presented by Britton and Skevington, and expands on their work to make the model more biologically plausible and provide a basis for further work with this model.

INTRODUCTION

Pain is a personal subjective experience that requires psychological awareness and can occur without tissue damage. By definition it is '*an unpleasant sensory and emotional experience associated with actual or potential tissue damage, or described in terms of such damage*' (Merskey and Bogduk 1994). This definition of pain clearly demonstrates the duality between the physiological and psychological experience. It is probably due to this, that despite much research, pain remains a phenomenon that is poorly understood. Theories of pain all possess inadequacies and limitations and only explain part of the pain process. From these theories and their own observations Melzack and Wall (Melzack and Wall 1965) proposed the influential gate control theory (GCT). With the inclusion of a descending control, it was possible to explain the variation in the pain felt between individuals, and how pain intensity is not in direct relation to tissue damage as cognitive, emotional, social and environmental factors can all influence pain. This explicit theory made it feasible to express in a mathematical model (Britton et al. 1996; Britton and

Skevington 1989). However despite the success of this model this useful and powerful technique has not been utilised further in this field. With more research now available on pain it seems appropriate to extend the model to ascertain if it is still successful. Firstly however, it needed to be replicated. The following will give a brief outline of the processes involved in pain, limiting the review to only those involved with cutaneous stimulation. An overview of the gate control theory (Melzack and Wall 1965) will be presented before considering the mathematical model (Britton et al. 1996; Britton and Skevington 1989)

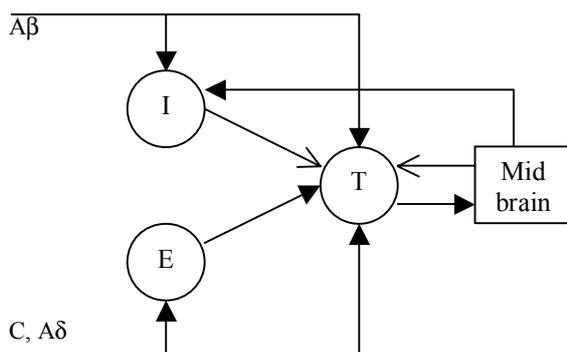
Physiology of Pain

Studies that have used blocking techniques have implied the existence of specialised receptors that respond to tissue damage or to potential tissue damage (Bessou and Perl 1969; Burgess and Perl 1967). The existence of these nociceptors, sensitive to a variety of different kinds of stimulus - thermal, mechanical and chemical - have been confirmed by subsequent studies (see review by Besson and Chaouch 1987). When an adequate stimulus is applied to the skin, the receptors located here convert the physical energy into electrochemical energy. Information is then transmitted in the form of trains of action potentials (similar to pulse trains) via nerve fibres to the dorsal horn of the spinal cord, which then activates the associated transmission neuron, which is the first stage of the pathway to the cortex, and hence into consciousness.

The fibres that convey information from the receptors consist of large myelinated (insulated) A β fibres and small unmyelinated and myelinated (C and A δ respectively) fibres. A β fibres are primarily involved in the transmission of non-painful sensation such as touch. A δ fibres are associated with well-localised sensations of sharp, pricking pain and also subserves pressure, crude touch, and temperature. C fibres are involved in diffuse pain sensations that can be dull, poorly localised and persistent (Ochoa and Torebjork 1989; Torebjork and Ochoa 1980) whilst also subserving temperature. Pain is felt following stimulation of A δ fibres (first pain), this then builds up and intensifies when C fibres are also activated (second pain).

Dependent on their function, these fibres terminate in laminae I, II and V in the dorsal horn, which then activate associated neurons. Of particular interest for pain are the wide dynamic range (WDR) neurons and the nociceptive specific (NS) neurons. Receiving inputs from nociceptive and non-nociceptive fibres the WDR cells can be found in the deeper laminae of the dorsal horn. Conversely the NS neurons can be found in the superficial laminae of the dorsal horn, receiving inputs from C fibre afferents.

It is predominantly the lamina II (substantia gelatinosa (SG)) that Melzack and Wall (Melzack and Wall 1965) attribute with the gate mechanism, modulating the flow of pain information to the brain, and directly affecting the pain experience. Following peripheral stimulation, the afferent impulses from the large ($A\beta$) and small (C, $A\delta$) fibres are received directly at the SG and the first transmission (T) cells. As illustrated by figures 1, the T cells also receive input from the inhibitory (I) and excitatory (E) interneurons that are located deep within the SG. As well as large fibre input, the inhibitory interneuron also receives efferent information from the descending control. In 1965 there was little evidence to verify the existence of such a descending control, however subsequent findings have supported it (see review by Fields and Basbaum 1999). Initially just inhibition was thought to occur, however more recent evidence suggests that there is also a facilitatory action of efferent impulses on nociceptive transmission (Zhuo and Gebhart 1992). On receiving the afferent impulses the T cells perform spatial and temporal summation of all the arriving impulses, which then triggers an action system, and it is this system that is responsible for the experience and behaviour of pain.



Figures 1: The Basic Architecture of the Gate Control Theory used in the Mathematical Model of Pain; the filled arrows denote excitation, the unfilled arrows inhibition

Modelling of pain

The utilisation of computational models in the field of pain has been very limited. In 1981 Minamitani and Hagita (Minamitani and Hagita 1981) produced a neural

network that was able to simulate the conduction mechanisms involved in pain. Haeri et al (Haeri et al. 2003) produced an Artificial Neural Network (ANN) to model steady state behaviour of pain mechanisms and allow prediction of pain given a novel stimulus. Britton and Skevington (Britton et al. 1996; Britton and Skevington 1989) used the explicit nature of the gate control theory to extrapolate the relevant features and translate it into a mathematical model.

The mathematical model (Britton et al. 1996) was subjected to four simulations. The first simulation was constant small fibre input with variable large fibre input. A successful simulation provides support for what has been mainly anecdotal evidence as independent stimulation of small and large fibres is difficult to achieve experimentally. The next simulation Britton and Skevington completed involved small fibre input only. This was to ascertain that as the small fibre input is increased so does the T cell output at a rate slightly greater than linear. Wind-up (Mendell 1966) was also simulated. This involves the repeated stimulation of C fibres resulting in a progressive increase in the T cell response. Evidence suggests that stimulation of glutamate receptors, in particular N-Methyl D Aspartate (NMDA) receptors are involved in the wind-up process and Britton and Skevington were able to illustrate this relationship. The last simulation was that of ramp-off, this occurs when the stimulus is ramped off causing a pulse of pain (Humphries et al. 1993; Humphries et al. 1996). Whilst the model produced expected results in line with the literature, certain assumptions were made in order to simplify the model, as laid out in figure 1. Inputs received by the T cell were from one large fibre ($A\beta$), one small fibre (C), one inhibitory neuron and one excitatory neuron, with no input from $A\delta$ fibre. Neighbouring T cells were assumed to behave in the same way and that any input from $A\delta$ would not affect the results. Assumptions of this nature are common in order to simplify such complex processes whilst leaving the salient features intact, however, they render the model less biologically plausible.

This paper replicates Britton and Skevington's model and expands it further to ascertain if it is still successful. It will explore whether the neighbouring T cells behave in a similar fashion, and seek to resolve if an increase in T cells and associated cells greatly affects the results previously examined.

METHOD

Replication of the Mathematical Model

Using the differential equations given by Britton and Skevington (Britton et al. 1996) the model was translated into MATLAB version 6. The differential equations used were:

$$\tau_i V_i = -(V_i - V_{i0}) + g_{it}(x_i) + g_{mi}(x_m) \quad (1)$$

$$\tau_e V_e = -(V_e - V_{e0}) + g_{se}(x_s, V_e) \quad (2)$$

$$\tau_t V_t = -(V_t - V_{t0}) + g_{st}(x_s) + g_{it}(x_i) + g_{et}(x_e) - g_{it}(x_i) - g_{mt}(x_m) \quad (3)$$

$$\tau_m V_m = -(V_m - V_{m0}) + g_{tm}(x_t) \quad (4)$$

The precise nature of the equations used is as follows:

$$0.7V_i = -(V_i + 70) + 60 \tanh(\theta_i x_i) + 40 \tanh[f_m(V_m)] \quad (5)$$

$$0.7V_e = -(V_e + 70) + 40 \tanh(\theta_{se} x_s) \{1 + 2.9 \tanh[4f_e(V_e)]\} \quad (6)$$

$$0.7V_t = -(V_t + 70) + 40 \tanh[\{1 - \theta_{se}\} x_s] + 40 \tanh[(1 - \theta_i) x_i] + 40 \tanh[f_e(V_e)] - 40 \tanh[f_i(V_i)] - 40 \tanh[f_m(V_m)] \quad (7)$$

$$0.7V_m = -(V_m + 70) + 40 \tanh[f_t(V_t)] \quad (8)$$

The mathematical model calculates the slow potential (V_i) for the inhibitory SG cell (1,5), the excitatory SG cell (2,6), T cell (3,7) and the midbrain (4,8). The firing frequency (x_i) at which the cell fires is a function of their slow potential, given by the function f . The qualitative features of f and g (monotone increasing function) are presumed known and $g \circ f$ is a saturating function, as the firing rate cannot increase indefinitely. In equation (6) $\tanh[4f_e(V_e)]$ is the NMDA component of the equation. The resting potential of a cell is taken to be -70mV . θ_i and θ_{se} denotes the amount of input that passes through the interneurons, whilst $(1 - \theta_{se})$ and $(1 - \theta_i)$ represents the proportion passing through the T cell. The output from the T cell is taken to be in direct relation to the pain experience, such that if the T cell exceeds its firing threshold of -55mV then pain is felt.

The input values of x_s and x_i given for the four simulations performed were:

1. Constant small fibre input, variable large fibre input

$$x_s = 2.0, \quad 0.0 \geq x_i \geq 3.0$$

2. Small fibre input only

$$x_s = 2 \tanh t, \quad x_i = 0.0$$

3. Wind-up simulation

$$x_s = 2.5 \cos^8(2\pi t) \quad x_i = 0.1x_s$$

4. Ramp-off simulation

$$x_s = \begin{cases} 2 & \text{if } 0 \leq t \leq 7, \\ 10(7.2 - t) & \text{if } 7 < t < 7.2, \\ 0 & \text{if } t \geq 7.2, \end{cases}$$

$$x_i = \begin{cases} 1 & \text{if } 0 \leq t \leq 6, \\ 5(6.2 - t) & \text{if } 6 < t < 6.2, \\ 0 & \text{if } t \geq 6.2, \end{cases}$$

Extension of the Model – Two Units

The architecture of Britton and Skevington's model, as described previously and shown in figures 1, assumed that neighbouring T cells behaved in the a similar way. To establish if this assumption was justified and to see if

increasing the number of units would alter the T cell potential the model was extended. The midbrain received input from more than one unit. Each unit consisted of one small and one large fibre input, one inhibitory and one excitatory interneuron and one T cell. If two units were required then equations (5),(6) and (7) were repeated. The output from each T cell was then inputted into the differential equation for the midbrain. Thus the midbrain equation was as follows:

$$0.7V_m = -(V_m + 70) + 40 \tanh[f_i(V_i)] + 40 \tanh[f_{it}(V_{it})] \quad (9)$$

A model consisting of two units was then constructed and the four simulations used previously were performed, recording the theoretical T cell potential to offer a direct comparison.

Multiple Models

The advantage of manually adding the units to the model, as described above, allows differing small and large fibre inputs to be fed into the differing units, which is something that will be explored in future work. The problem with such a model is that it is very time consuming. So a second model was produced that could perform the same simulations using N units.

For this, a function was devised and completed in MATLAB to perform the addition of units to the equation. Again the same values for simulation were used to offer a direct comparison and the number of units were increased in denominations of ten, until 50 units completed, and then the number of units increased by 50 until 200 units had been implemented.

RESULTS

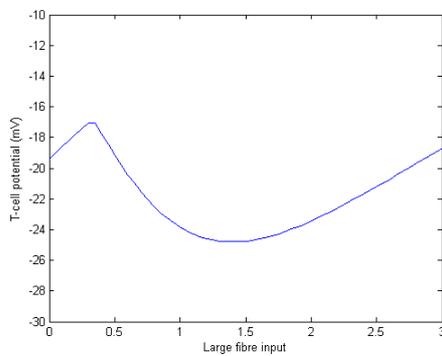
Replication of the Mathematical Model

The differential equations proposed by Britton and Skevington were run on a mathematical program with four simulations being performed; variable large fibre input with a constant small fibre input, small fibre input only, wind up simulation and ramp off simulation. The results produced were in accordance with Britton and Skevington's (see Britton and Skevington 1996). Simulation 1 was repeated six times to record T cell potential over time; large fibre input was presented at 0.5 through to 3.0 with increments of 0.5 with each pass. The results show that when large fibre input is at 0.5 the T cell potential is approximately -20mV . As the input increases, the T cell potential decreases to around -25mV . When the input is run at 3.0, the T cell potential has increased back to -20mV

Two Units

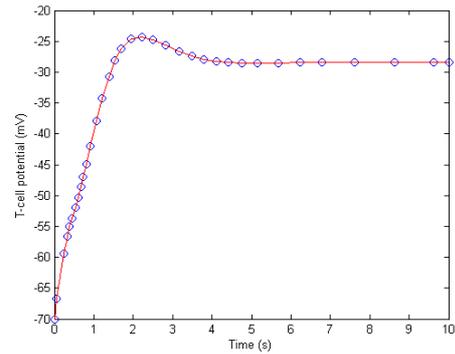
The model was extended to include two units, each consisting of one large and one small fibre input, one inhibitory and one excitatory interneuron and one T cell that was inputted into the midbrain. The T cell potential was then recorded, for each of the four simulations, as it is this that is taken to be indicative of the pain experienced.

When the T cell potential is recorded against the increase in large fibre input with the small fibre stimulation held constant, there is very little difference in the results compared to when there is only one unit present (see figures 2). It also shows that the T cell potential for the two neighbouring units is identical, as would have been predicted with this model.

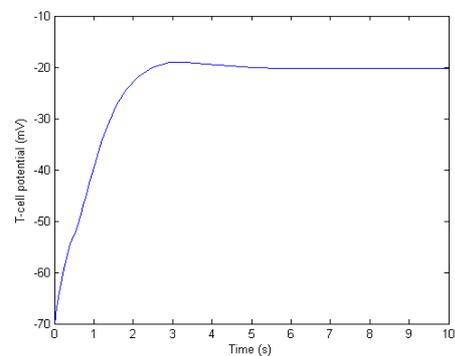


Figures 2: The Theoretical T Cell Potential for Two Units with Variable Large Fibre Input and Constant Small fibre stimulation

When the same simulation was conducted, this time with the T cell potential plotted over time starting with large fibre input at 0.5 through to 3.0 increasing in increments of 0.5 there was a change in results between one and two units. In all cases the theoretical T cell potential was lower when there were two units present. The pattern of the T cell potential also differed; after an initial increase the T cell potential fell slightly before levelling off when two units were implemented (see figures 3). In comparison, when there is only one unit present the T cell potential reaches its highest point and then levels out (see figures 4). This offers further support to the gate control theory as it states that the gate is held in a relatively open position, so that information regarding pain can flow freely. The large fibre input is known to be extremely effective in activating T cells. The ensuing reduction in T cell activity is due to the activation of the inhibitory interneuron by the large fibre input and the descending control. The levelling of the T cell potential occurs where the large and small fibre input counteracts one another.



Figures 3: Theoretical T Cell Potential when Two Units Modelled with Large Fibre Input at 3.0 and Small Fibre Input at 2.0



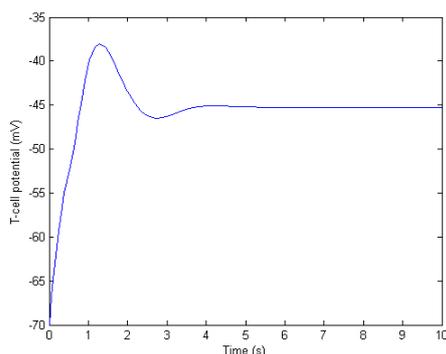
Figures 4: The Theoretical T Cell Potential when One Unit Modelled with Large Fibre Input at 3.0 and Small Fibre input at 2.0

When there is only small fibre input, the T cell potential reaches a peak and then falls off very slightly. The theoretical T cell output is also marginally lower than when compared to the results for only one unit. Inhibition would be expected to be less prevalent as there is no large fibre input. Thus any occurrence of inhibition is due to the descending control, which influences the inhibitory interneuron. There are no notable changes in the results for wind up, with the same pattern and T cell potential reported for two units as with one unit. For ramp off, after the initial rise in T cell potential, there is a more marked decrease before the pulse of pain occurs. The T cell potential for two units is also lower. Given the previous results, this was perhaps not unexpected.

Multiple units

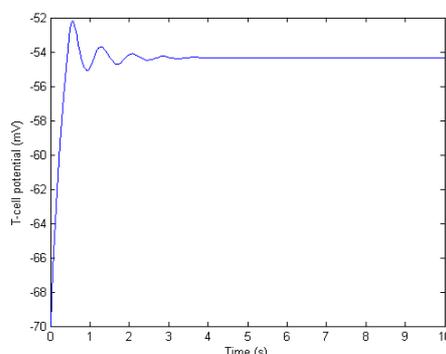
A function was developed in MATLAB to allow N number of units to be selected and the program would return the T cell potential for that number of units. All units received the same input value from the small and large fibres and the four previously used simulations were performed. As with previous results the first stimulation was run with T cell potential recorded over

time. Figures 5 illustrates that when the units are increased to 10 there is a very different pattern that emerges for the T cell potential. After an initial rise followed by a fall in T cell potential, as seen previously, there is a further rise before it levels out. At 200 units the T cell potential rises and falls several times before it levels out (see Figures 6). What also becomes apparent is as the number of units is increased the T cell potential decreases. When there is only one unit the T cell potential reached for simulation 1 when large fibre input is at 3.0 is approximately -20mV , by 200 units it is around -52mV just above the firing threshold of -55mV .



Figures 5: The Theoretical T Cell Potential with 10 Units, Large Fibre Input at 3.0 with Small Fibre Input at 2.0

Figures 6: The Theoretical T Cell Potential with 200



Units, Large Fire Input at 3.0 with Small Fibre Input at 2.0

Similarly, for the remainder of the simulations; small fibre input only, windup and ramp off, a similar pattern emerged in that the T cell potential output decreased as the number of inputs increased. Also the output pattern changed in that in the case of simulation 2, T cell potential after an initial rise, dropped and then rose again before it levelled out. For windup, up to 50 units once the T cell potential had reached its peak the rise and fall in potential remained to the same levels. However, post 50 units although the T cell potential continued to rise and fall to the same levels, what was

observed in between was a lesser rise in T cell potential followed by a slight decrease before increasing further.

DISCUSSION

The primary aims of this paper were to replicate the Britton and Skevington mathematical model for pain and extend this model further to be more biologically plausible. The ability to replicate the model adds to the robustness of it and supports the model proposed. With mathematical software available such as MATLAB it means that the model can be extended and manipulated with relative ease. This has resulted in the production of a model that is now more biologically plausible and yet, although it has produced different results to those perhaps expected, it remains within the essence of Melzack and Wall's paper. Some of the results have also given rise to further questions. In particular we refer to the multiple units and the reduction in the T cell potential when the number of units is significantly increased. This leads to the suggestion that if the number of units were increased further then the T cells may fail to fire, or they might reach a saturation point from which they go no lower. This indicates the importance of the level of input both in terms of the number of small and large fibres activated and their frequency. From these observations a model is currently being produced that has multiple small and large fibre input, enabling further testing to ascertain their impact on the model.

The inclusion of two units produced the results that would be expected in that the neighbouring T cells behaved in the same manner. This was not unexpected as they received the same small and large fibre inputs. What is significant about the model is that it allows different input values to be attached to the small and large fibre inputs to each unit. From this we are currently in process of simulating transcutaneous electrical nerve stimulation (TENS) and acupuncture. TENS works by high frequency, low amplitude stimulation of large peripheral fibres, whilst acupuncture involves low frequency, high amplitude stimulation of small $A\delta$ fibres. Stimulation as described by these methods produces modulation at the gate. As acupuncture involves $A\delta$ fibre activation a time delay needs to be implemented so that $A\delta$ and C fibres (corresponding to first and second pain respectively) can be modelled. A model that is able to test the effectiveness of such pain relief techniques could have significant implications for how pain is treated. There still remains much debate as to how effective these techniques are, the type of pain that it modulates, and at what point it fails to have any benefit (Sluka and Walsh 2003).

Despite its success the model has some very crucial limitations that we will endeavour to explore further. Firstly, the role of the midbrain is very simplified.

Although a single pain centre in the brain has been refuted, imaging studies have made clear that there are many regions of the brain involved in the pain process, making it more similar to consciousness than to primary sensory modalities. A previous neural network model (Minamitani and Hagita 1981) has explored some of the connections of the brain regions involved, and it remains something vital to any comprehensive model of pain. Also the problem of chronic pain needs to be addressed. So far the limited models produced have mainly focused on acute pain, as this is better understood. However, it is chronic pain that causes many to suffer and modelling this could prove vital for research in this area. Neural networks could provide the way forward for this, as plasticity would need to be a key feature. One such suggestion is the use of chaotic neural networks, which are capable of modelling such plasticity (Picton et al. 2001).

CONCLUSION

The gate control theory (Melzack and Wall 1965) was the first to offer a resolution to the duality between the physiological and psychological experience present in the pain phenomenon. The mathematical model successfully gives a good approximation of the inputs involved and how the inputs affect the related cells and activate them to produce pain. However, in order to provide a more comprehensive model to explain acute and chronic pain further work is required to include multiple fibre input, plasticity and a comprehensive descending control. The work completed so far certainly supports the utilisation of modelling in the field of pain.

REFERENCES

- Besson, J.M. and Chaouch, A., Peripheral and Spinal Mechanisms of Nociception, *Physiological Reviews*, 67 (1987) 67-186.
- Bessou, P. and Perl, E.R., Response of cutaneous sensory units with unmyelinated fibres to noxious stimuli, *Journal of Neurophysiology*, 32 (1969) 1025-1043.
- Britton, N.F., Chaplain, M.A.J. and Skevington, S.M., The role of N-methyl-D-aspartate (NMDA) receptors in wind-up: A mathematical model, *Journal of Mathematics Applied in Medicine & Biology*, 13 (1996) 193-205.
- Britton, N.F. and Skevington, S.M., A mathematical model of the gate control theory of pain, *J. theor. Biol.*, 137 (1989) 91-105.
- Burgess, P.R. and Perl, E.R., Myelinated afferent fibres responding specifically to noxious stimulation of the skin, *Journal of Physiology-London*, 190 (1967) 541-562.
- Fields, H.L. and Basbaum, A.I., central nervous system mechanisms of pain modulation. In: P.D. Wall and R. Melzack (Eds.), *Textbook of Pain*, Churchill Livingstone, Edinburgh, 1999, pp. 309-329.
- Haeri, M., Asemani, D. and Gharibzadeh, S., Modeling of pain using artificial neural networks, *Journal of Theoretical Biology*, 220 (2003) 277-284.
- Humphries, S.A., Johnson, M.H. and Long, N.R., an investigation of the gate control theory of pain by ramping-off the experimental pain stimulus of potassium iontophoresis, abstracts of the 7th world congress on pain, IASP, Seattle, 1993, pp. 468.
- Humphries, S.A., Johnson, M.H. and Long, N.R., An investigation of the gate control theory of pain using the experimental pain stimulus of potassium iontophoresis, *Perception & Psychophysics*, 58 (1996) 693-703.
- Melzack, R. and Wall, P.D., Pain Mechanisms: A New Theory, *Science*, 150 (1965) 971-979.
- Mendell, L.M., Physiological Properties of Unmyelinated Fiber Projection to the Spinal Cord, *experimental neur*, 16 (1966) 316-22.
- Merskey, H. and Bogduk, N., Classification of Chronic Pain: Definitions of Chronic Pain Syndromes and Definitions of Pain Terms, International Association for the Study of Pain, Seattle, 1994.
- Minamitani, H. and Hagita, N., A Neural Network Model of Pain Mechanisms Computer Simulation of the Central Neural Activities Essential for the Pain and Touch Sensations, *Ieee Transactions on Systems Man and Cybernetics*, 11 (1981) 481-493.
- Ochoa, J. and Torebjork, E., Sensations Evoked by Intra-neural Microstimulation of C Nociceptor Fibers in Human-Skin Nerves, *Journal of Physiology-London*, 415 (1989) 583-599.
- Picton, P.D., Campbell, J.A. and Turner, S.J., Modelling Chronic Pain: An Initial Survey, Proc. 8th International Conference on Neural Information Processing, Shanghai, 2001.
- Sluka, K.A. and Walsh, D., Transcutaneous electrical nerve stimulation: Basic science mechanisms and clinical effectiveness, *Journal of Pain*, 4 (2003) 109-121.
- Torebjork, H.E. and Ochoa, J.L., Specific Sensations Evoked by Activity in Single Identified Sensory Units in Man, *Acta Physiologica Scandinavica*, 110 (1980) 445-447.
- Zhuo, M. and Gebhart, G.F., Characterization of Descending Facilitation and Inhibition of Spinal Nociceptive Transmission from the Nuclei Reticularis-Gigantocellularis and Gigantocellularis Pars Alpha in the Rat, *Journal of Neurophysiology*, 67 (1992) 1599-1614.

AUTHOR BIOGRAPHY

KAREN PRINCE was born in Peterborough, England and went to the University of Hertfordshire, where she studied cognitive science and obtained her degree in 2001. Currently she is working towards her PhD at University College Northampton. Her email address is k.m.fulstow@btinternet.com.

JACKIE CAMPBELL was born in Northamptonshire and studied physics at Liverpool University and medical physics at the University of Surrey. She worked in pain research at the Pain Relief Foundation and Walton Hospital in Liverpool, and gained her PhD in the electrophysiology of the human spinal cord. She is currently Professor of Neurophysiology and Head of Research and Consultancy in the School of Health at University College Northampton. Her e-mail address is jackie.campbell@northampton.ac.uk

PHIL PICTON was born in South Wales and did his Electronics degree at Swansea University followed by his PhD in Logic Design at Bath University. He was a Lecturer in Electronics at The Open University for 10 years and is now Professor of Intelligent Computer Systems at University College Northampton. His research is in the application of artificial intelligence techniques, particularly neural networks, to a variety of modelling problems. His e-mail address is phil.picton@northampton.ac.uk.

SCOTT TURNER was born in Birmingham, U.K did his electronic degree at University of Hull followed by MSc in Biomedical Instrumentation Engineering at the University of Dundee and PhD from the University of Leicester. Currently a Lecturer at University College Northampton. His email address is scott.turner@northampton.ac.uk.

A MESOSCOPIC APPROACH TO MODELING AND SIMULATION OF PEDESTRIAN TRAFFIC FLOWS

Juri Tolujew and Felix Alcalá
Fraunhofer Institute for Factory Operation and Automation
Sandtorstr. 22, 39106 Magdeburg, Germany
Juri.Tolujew@iff.fraunhofer.de

KEYWORDS

Pedestrian traffic flows, mesoscopic simulation, online simulation, early-warning system

ABSTRACT

The paper discusses the simulation of pedestrian traffic flows in public buildings from a practical point of view: The models developed are intended for use in an operative early-warning system. They must be able to forecast how the situation will evolve in each room of the building for the near future. By evaluating possible measures they can be used as a tool that pro-actively avoids critical system states. A mesoscopic approach to creating these models is described and partially realized. It envisions groups of persons as modeling and simulation objects and thereby distinguishes between physical and logical (virtual) groups. The paper presents novel solutions to the modeling of both types of person groups. An example of a model described in XML and implemented with Java is demonstrated.

INTRODUCTION

Models for pedestrian traffic flows are often categorized by the type of moving objects used to represent the persons to be simulated. On the one hand there is *macroscopic simulation* used to model relatively large numbers of persons that are distributed in an abstract space. Analogous to gases and liquids, physical laws and therefore differential equations can describe their movement (Helbing 1997). In *microscopic simulation* on the other hand every moving object represents exactly one person. Thus even the smallest geometric detail of their movement can be considered. The most important members of the category of microscopic models are models based on cellular automata (Schreckenberg and Sharma 2002) and queuing systems in the form of analytical models or simulation models (Løvås 1994).

The authors of this paper deal with the simulation of pedestrian traffic flows in conjunction with the development of a concept for an *operative early-warning system*. This system monitors and controls pedestrian traffic flows in a public building which is a traffic junction for large quantities of travelers (Hanisch et al. 2003). Since the early-warning system operates *online*, the model must have specific properties:

- The forecast interval is approximately one hour.
- The results of the simulation contain information on the distribution of the *amounts of persons* in predefined rooms of the building.
- The model is started at regular intervals, usually around 5 minutes.
- Since it is based on *direct measurements* of pedestrian traffic flows in each room, the input data is an exact image of the current situation in the building.
- All *real and planned events* are considered in every simulation run. This includes the arrival and departure of passengers as well as actions of the personnel controlling the pedestrian traffic flows.

Critical analysis of the aforementioned categories concluded that none of them can be used for the development of models for an early-warning system. Macroscopic models based on differential equations seem to be too imprecise in this situation since normally they can only be used to describe the average, collective behavior of persons. The two types of microscopic models have a downside of relatively long computing times. Additionally, their level of detail does not match the level of detail at which processes in rooms of a building can be observed and described.

A CLASS OF MESOSCOPIC MODELS

The main idea behind the approach presented in this paper is that *person groups* are regarded as modeling and simulation objects. This approach can be called *mesoscopic modeling and simulation* because it is situated between the microscopic and the macroscopic methodology (Lerner et al. 2000). In this class of modeling, two kinds of person groups can be distinguished: *physical groups* (also real groups) correspond to person quantities actually in particular rooms. Adequate sensors can estimate the number of persons in a physical group and this number is used as input data for the simulation. *Virtual or logical groups* represent person groups whose members show similar behavior because they have the same intention (e.g. to board a specific train). The members of a logical group are normally divided among several physical groups. Consequently, every physical group typically consists of members of several logical groups.

The simulation uses a discrete time with a step size of Δt , which in principle can also be variable. Two models have to be calculated in every time step, one for each kind of groups. Position changes of persons are mapped in the model for physical groups in such a way that new physical groups are formed at the end of each time step Δt . Individual persons' behavior such as choosing a new path is calculated in the model for logical groups and is statistically correlated with reality. The model of logical groups can be updated at the end of time step Δt if a new group is needed or if an existing group can be discarded because it has no more members.

IMPLEMENTATION OF MODELS FOR PEDESTRIAN TRAFFIC FLOWS

Models based on differential equations are usually developed by applying simulators for continuous processes (e.g. MATLAB/Simulink). Any commercial simulator (e.g. Enterprise Dynamics or eM-Plant) is suitable for executing microscopic simulations in the form of queuing systems. Special tools like JCASim are used for microscopic models based on cellular automata.

None of the features of these universal or special simulators constitutes an advantage when working with the category of mesoscopic models presented here. In fact, mesoscopic simulation requires the development of very specific algorithms. They are best implemented using a universal programming language, which leads to relatively high model performance, something extremely important for online simulations.

As discussed below, another aspect of the development of mesoscopic models is the necessary flexibility of the model. This can be achieved by following the principle of data-based model creation and using XML as the modeling language. Among other things, this paper discusses the results of a data-based, Java-based implementation of a mesoscopic model for physical person groups based on the principle of dynamic route segmentation.

EVENT-BASED GENERATION AND MODELING OF LOGICAL GROUPS

A simple example is treated, in which one physical group of persons (passengers on a platform) consists of a simple sum of several logical person groups since all persons to be modeled are located in one room. The arrival and departure of two trains is represented by four events shown in table 1.

Table 1: Description of Four Main Events

Event name	Event time	Event description
Ev1	13 min	arrival of train 1
Ev2	20 min	departure of train 1
Ev3	23 min	arrival of train 2
Ev4	31 min	departure of train 2

In systems where the inflow and outflow of person quantities are caused by means of transportation, most logical groups are formed by concrete events. Table 2, figures 1 and 2 show the eight processes that can be inferred from the events of table 1 (Pax is an abbreviation for the word "passengers"). Groups of departing persons are set up and populated before the corresponding events take place (processes Pr1 and Pr5). Likewise, every new group of arriving persons appears immediately after the occurrence of the corresponding event (processes Pr2 and Pr6).

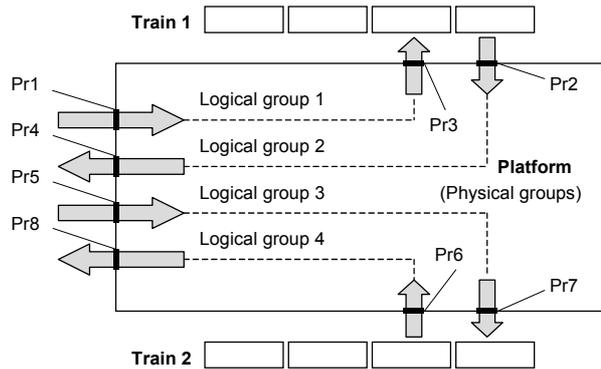


Figure 1: Observation Points for Eight Processes on a Platform

The model of behavior shown in figure 2 for four logical groups is simple but not trivial. For example, it takes into account that while a part of the group "boarding train 1" is already inside the wagon, the rest of this group may still be on its way to the train (the processes Pr1 and Pr3 overlap for two minutes). Moreover, the model contains 20 persons who change trains: 100 persons arrive with train 1 (process Pr2) but only 80 of them leave the platform (process Pr4) because the remaining 20 persons are waiting for train 2. The length of the platform is assumed to be such that the first person who has left a particular train can leave the platform after a three minute walk (offset between the beginnings of Pr2 and Pr4).

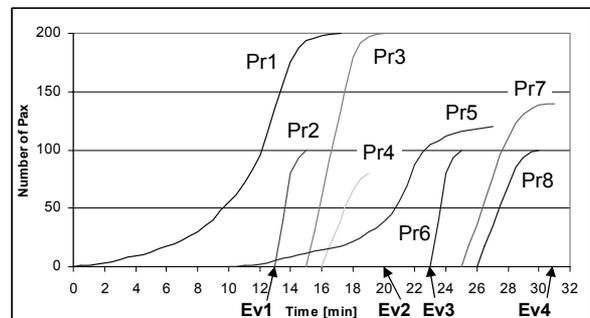


Figure 2: Cumulative Pax Quantities for Eight Processes on a Platform

Table 2: Description of Eight Processes on the Platform

Process name	Start time	End time	Number of pax	Flow direction	Process description
Pr1	0	17	200	input	The pax are coming onto the platform to board train 1
Pr2	13	15	100	input	The pax are deboarding train 1
Pr3	15	20	200	output	The pax are boarding train 1
Pr4	16	19	80	output	The pax who have arrived with train 1 are leaving the platform
Pr5	10	27	120	input	The pax are coming onto the platform to board train 2
Pr6	23	25	100	input	The pax are deboarding train 2
Pr7	25	31	140	output	The pax are boarding train 2
Pr8	26	30	100	output	The pax who have arrived with train 2 are leaving the platform

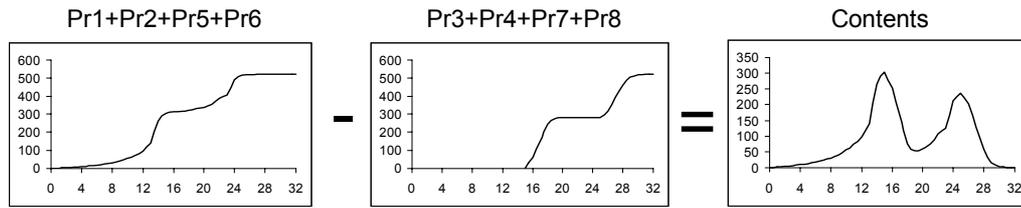


Figure 3: Calculation of the Entire Pax Quantity on the Platform

The characteristic of the process description in figure 2 is the use of cumulative person quantities as indicators. It has been demonstrated in (Tolujew 2003) that these dynamic indicators can be used as operands in arithmetic and algebraic operations. Figure 3 illustrates the last operation, which calculates the number of persons on the platform during the entire duration of the process. The four input and output processes were accumulated beforehand. The dynamic of the person quantity on the platform shown in the diagram “contents” is a precise result directly inferred from the specified processes Pr1 – Pr8.

MESOSCOPIC SIMULATION OF THE MOVEMENT OF PERSON GROUPS

The mesoscopic model consists of four basic components shown in table 3. They can be linked freely, i.e. each component can be reached from and may lead to any other component. Components can also be linked to themselves.

It is assumed that, in one time step, every component is capable of accepting any input quantity. The movement of persons is modeled using the component passageway, a unidirectional connection between two points in the real room. Using passageways as a model for pedestrian movement presupposes three assumptions:

- The time progresses in equidistant time intervals Δt .
- The situation depicted in the passageway is the one at the *end* of each time interval.
- A passageway is composed of several parallel “virtual tracks”, each of which is assigned a specific velocity. Thus, any velocity distribution can be approximated by defining several tracks

with different velocities. The number of tracks depends on the model precision required.

Table 3: Basic Components of the Mesoscopic Model

Component class	Component function	Examples
Source	Fills the model with persons	An arriving train; Main entrance of the building
Sink	Removes persons from the model	A departing train; Door to parking lot
Service station	Delays persons	Ticket counters; Security checkpoints
Passageway	Models distances between components. Persons are delayed until they have covered the distance.	Transfer from platform A to platform B

A track is characterized by the length l of the passageway and the specific velocity v of the persons on this track. All persons on the track advance a distance of $v \cdot \Delta t$ during the time Δt . This distance defines the length of the so-called segments, into which every track is divided. The number of segments in a track is $m = l / (v \cdot \Delta t)$; the result of the division is rounded up to the next integer. As a result, the total length of all m segments ($m \cdot v \cdot \Delta t$) may be longer than the length of the

passageway l . In this case, a part of the last segment (segment number m) is outside the track.

Figure 4 shows an example of a track with three segments ($m=3$). Additionally, a virtual fourth segment is defined (segment number $m+1$) to calculate the output quantity of this track at each time interval. The terms s_1, \dots, s_4 are used to name the length of each segment. The terms n_1, \dots, n_3 represent the number of persons in the respective segment at the end of the time step Δt .

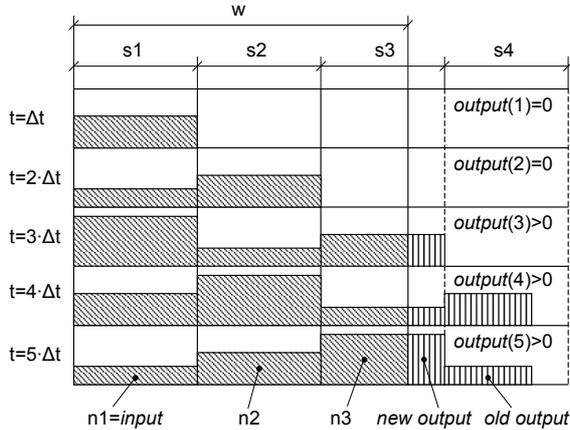


Figure 4: Basic model of one track of a tracked route in the mesoscopic approach using the method of dynamic route segmentation

Figure 4 illustrates the operation of one track during five time steps Δt . During the first time step, n_1 persons enter the track, indicated by the hatched area. The first few representatives of this quantity leave the track during the third time step; their number equals the hatched area behind the boundary of the passageway. During the time units 4 and 5 the output quantity of the track is the sum of two segment quantities (old output and new output).

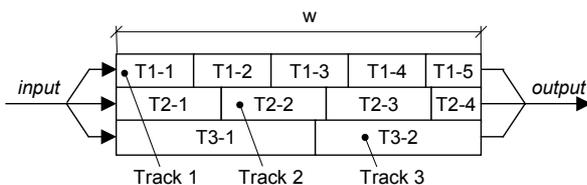


Figure 5: Basic model of a tracked route as a composition of several tracks

As stated before, each passageway consists of several tracks. Figure 5 illustrates the modeling of a passageway consisting of three tracks. The label “T3-1” denotes “Track 3, Segment 1”. The input quantities of the passageway are split between the tracks according to the specified statistical distribution of movement speeds. The output quantities of the entire passageway are the sum of the outputs of the different tracks.

IMPLEMENTATION AND VERIFICATION OF A TEST MODEL

In order to test both the accuracy and performance of our mesoscopic approach, a *prototype simulator for the movement of physical groups* was designed and implemented. It is intended to be used in a control center of a train station or an airport to deliver information on expected system behavior to a decision support system. If the simulation predicts a critical state (e.g. check-in times at counter 4 will grow too long; room A5 will become too full), the decision support system proposes solutions and presents their simulated results to the control center users, who can then decide which alternative is best.

The data needed in order to run the simulation is distributed across several independent data sources:

- The *model structure* must be inferred from a database that contains information about both the building’s structural design and its present state (e.g. malfunctions).
- The *current number of passengers* in each room, line and so on is measured by online sensors.
- *Future events* such as planned arrivals of passengers are stored in the transportation carriers’ booking systems. Further, the building state database contains information on events that influence the model structure, e.g. the temporary shutdown of an elevator due to maintenance work.
- *Past events* may influence the simulation, for example, passengers deboarding a train that has recently entered the train station. Future consequences of past events are calculated by a coupled logical group simulator, which has yet to be implemented.
- The simulator also provides *routing information* for the different person groups since the physical group simulator does not dynamically assign targets to person groups. Until the logical group simulator is implemented, only static routing is possible.
- The external optimizer or decision support system initiates the simulation run and thus provides relevant *simulation control parameters*.

In view of the variety of data sources, data collection is done by independent programs or so-called *agents* that collect the necessary data and provide it to the simulator (Hanisch et al 2004). The strength of this design is its flexibility: The simulator remains unaffected by any change of data sources; only the data collecting agent has to be adjusted.

Data collection is triggered before each simulation run. While XML is the de-facto standard for cross-platform, cross-vendor information exchange, it is also a good format for describing simulation systems (Wiedemann 2002). The prototype uses XML to transfer the results of data collection from the agents to the simulator. XML is also used to describe all simulation control parameters, which leads to a completely data-driven simulation.

To illustrate this technique, consider the (very basic) model presented in figure 6 and its corresponding XML notation in listing 1. Note that no additional information is needed in order to run the simulation and that the results of the simulation can be saved in various data formats such as XML files and Microsoft Excel worksheets. Of course, additional input data (events, the current system state, etc.) can be supplied.

```

<simulation>
  <model>
    <metainfo>[...]</metainfo>
    <elements>
      <element name="entry">
        <source>
          <constOut>60</constOut>
        </source>
      </element>
      <element name="Segmented Way">
        <passageway>
          <length>200</length>
          <speed>1</speed>
          <speed>1.5</speed>
        </passageway>
      </element>
      <element name="exit">
        <sink/>
      </element>
    </elements>
    <links>
      <link from="entry" to="way"/>
      <link from="way" to="exit" weight="0.8"/>
      <link from="way" to="way" weight="0.2"/>
    </links>
    <units>
      <time>min</time>
      <distance>m</distance>
      <speed>m/s</speed>
      <persons>pax</persons>
    </units>
  </model>
  <simulationControl>
    <starttime>0</starttime>
    <endtime>120</endtime>
    <stepsize>1</stepsize>
  </simulationControl>
  <reporting>
    <report type="excel">[...]</report>
    <report type="xml">[...]</report>
  </reporting>
</simulation>

```

Listing 1: XML Source Code of a Simulation Run of the Simple Circle Example

As one can imagine, the XML simulation files for complicated model structures become relatively large. The model the authors used to measure the performance of the prototype consists of 70 service stations, 60 routes and 25,000 passengers, and resulted in 550 lines of XML code. Nevertheless, the simulation setup averaged out at less than one second on an 800 MHz machine. It took about 0.8 seconds to simulate one hour of real process. This time is almost independent of the amount of passengers that are simulated, because in the model passengers are represented by numbers rather than by individual objects. That means that the running time for 1,000 passengers and 100,000 passengers will be as good as equal, if one hour of real process is modeled in both cases. To compare this performance with conventional micro-simulation, an eM-Plant model

was tested on the same machine. It took about three seconds to simulate 2,500 passengers; while a run with all 25,000 passengers usually needed more than 30 seconds.

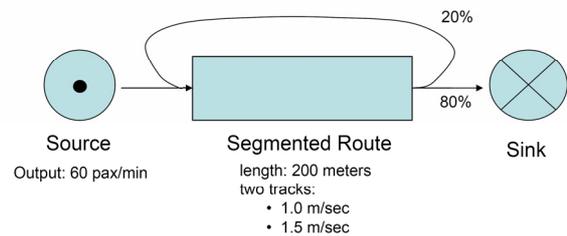


Figure 6: Simulation Model containing a Simple Circle

CONCLUSIONS AND FUTURE WORK

With its rather simple models, the mesoscopic approach presented outperforms classical microscopic simulation systems. Yet, it is fairly precise, allowing it to be used in any domain where both accuracy and speed are important.

One future step to improve the physical group simulator will be implementing the overcrowding of passageways. The main idea is to reduce the movement speed of the persons on an overcrowded route by a factor that is a function of the route's nominal capacity and its current load. Another step will be to implement the concept of rooms since they determine the amount of space available. This will make it possible to calculate the so-called *level of comfort*, indicating whether rooms are too empty or too full.

The most important task is to design and implement a simulator for logical groups and to couple it to the simulator for physical groups. The prototype will then be able to simulate pedestrian traffic flows with a dynamic model that changes during run time. Imagine, for example, the simulation is notified that an escalator is out of order. The simulation time is returned to the corresponding position and the simulation model is simply adjusted to the new conditions. Another example is the simulation of pedestrian traffic flows when a suspicious piece of luggage is found. In this case, some passageways or service stations have to be blocked while other, alternative routes are opened. In our model, the event can be placed anywhere in the future, resulting in immediate changes to the model at the precise time of the incident. As a result, together with its data-based model creation, the mesoscopic approach offers a level of flexibility that at the least is uncommon for classical industrial simulation systems.

REFERENCES

Hanisch, A., Tolujew, J., Richter, K., Schulze, T. "Online Simulation of Pedestrian Flow in Public Buildings". *Proceedings of the 2003 Winter Simulation Conference*, 1635-1641.

- Hanisch, A., Tolujew, J., Meuschke, T., Schulze, T. "Datenkollektion zur online Simulation von Personenströmen". *Simulation und Visualisation 2004*, Proceedings der Tagung in Magdeburg am 3. und 4. März 2004, 27-38.
- Helbing, Dirk: *Verkehrsdynamik: Neue physikalische Modellierungskonzepte*. Springer-Verlag, Berlin Heidelberg New York, 1997.
- Lerner, G., A. Hochstaedter, R. Kates, C. Demir, J. Meier, A. Poschinger. "The interplay of multiple scales in traffic flow: coupling of microscopic, mesoscopic, and macroscopic simulation". *7th ITS World Congress on Intelligent Transport Systems*, Turin, Italy, 2000.
- Løvås, G. "Modeling and simulation of pedestrian traffic flow". *Transportation Res. B*. Vol. 28, No. 6, 1994, 429-443.
- Schreckenberg, M. and S. D. Sharma. *Pedestrian and Evacuation Dynamics*. Springer-Verlag, Berlin Heidelberg New York, 2002.
- Tolujew, J. "Prozessbilder als Grundlage für die Prozessanalyse in Materialflusssystemen." *Simulationstechnik*, 17. ASIM-Symposium in Magdeburg, R. Hohmann (Hrsg.), SCS Int., Ghent 2003, 343-348.
- Wiedemann, T. "XML-Standardschnittstellen für Simulations- und Animationsdaten". *Simulation und Visualisierung 2002*, Proceedings der Tagung in Magdeburg am 28. Februar und 1. März 2002, 59-70.

AUTHOR BIOGRAPHIES



JURI TOLUJEW is a project manager in the Department for Logistics Systems and Networks at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg (Germany). He received the Ph.D. in automation engineering from the University of Riga in 1976. He also received his habil. degree in computer science from the Otto von Guericke University Magdeburg in 2001. His research interests include simulation-based analysis of production and logistics systems, protocol-based methods for the analysis of processes in real and simulated systems as well as mesoscopic approaches to simulation.



FELIX ALCALA, born in Munich, is studying Commercial Information Technology at the Otto von Guericke University Magdeburg (Germany) where he received his intermediate Diplom in October 2003. His area of specialization is simulation. He has been working as a research engineer at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg since January 2004. He can be contacted by e-mail at mail@felixalcala.de; his website is <http://felixalcala.de>.

IMPLEMENTATION OF A TILEWORLD TEST-BED ON A DISTRIBUTED BLACKBOARD SYSTEM

K.W. Choy, A.A. Hopgood, L. Nolle, B.C. O'Neill

The Nottingham Trent University
School of Computing and Technology,
Burton Street,
Nottingham NG1 4BU
United Kingdom

E-Mail: {kw.choy, adrian.hopgood, lars.nolle, brian.oneill}@ntu.ac.uk

KEYWORDS

TileWorld, multi-agent simulator, distributed blackboard system, DARBS

ABSTRACT

Research into multi-agent systems has increased over several years, leading to demand for multi-agent system simulators to investigate the interaction between different agents and their environment. The more complex and intelligent the agents become, the more difficult it is to simulate them. Distributing the simulation across different processors would improve its performance. This paper describes the implementation of a multi-agent test-bed called TileWorld on a distributed blackboard system, DARBS. The slow-down of the simulator on a single processor has been measured as the number of agents increases, and the likely effect of migrating to a multiple-processor system is assessed.

1. INTRODUCTION

Research in multi-agent systems is currently on the increase. Investigating the behaviour of the agents' interaction with each other and with the environment is inevitably part of this research. There is a need for a test-bed to simulate the agents in an artificial environment that can be both dynamic and static. TileWorld (Pollack and Ringuette 1990) is a well-established test-bed for agent systems and MA-TileWorld (Ephrati et al. 1995) is its multi-agent version.

Unfortunately, this type of simulation becomes processor-intensive as the number of agents is increased. An ideal solution to this would be to distribute the agents across different processors. A distributed blackboard system such as DARBS (Distributed Algorithmic and Rule-based Blackboard System) (Nolle et al. 2001) would be a suitable architecture.

1.1 TileWorld Test-bed

From here onwards, MA-TileWorld will simply be referred to as TileWorld. In this test-bed, the world is set up as a two dimensional grid. There are agents, tiles,

holes, and obstacles in the TileWorld. The objective of the agents is to score as many points as possible. They score points by moving around the TileWorld to find and pick up tiles which they put into holes. The agents have a limited view of the TileWorld. The viewing radius is a variable that can be set. For example, in the DARBS TileWorld test-bed, the agents have a viewing radius of 5 cells. Each cell can only be occupied by one agent at a time. Agents cannot move to a cell with an obstacle in it. An example of a 10 × 10 TileWorld is shown in Figure 1.

	1	2	3	4	5	6	7	8	9	10
1										
2	T4	O1			O6	T1				
3										O4
4				A1			H1			T3
5										
6		O5								
7							O2		A2	
8										
9		H2			O3	T2			H3	
10										

Legend

Ax Agent x

Hx Hole x

Ox Obstacle x

Tx Tile x

Figure 1: A 10 × 10 TileWorld

The holes, obstacles, and tiles in the TileWorld can change dynamically, i.e. they can appear and disappear in different locations in the TileWorld. The rate of change is set by a variable, and this can be used to reflect the dynamically changing real-world environment. The TileWorld test-bed has been widely used to test the behaviour and interaction of multiple agents in a dynamic environment (Lees et al. 2003; Kinny et al. 1992). There is also a variant of the original TileWorld test-bed that includes "gas station" objects to top up the resources of the agents (Uhrmacher and Schattenberg 1998). In this variant, the agent's resource-management skill is investigated by making each move consume fuel. Carrying a tile would cause the agent to consume more fuel. Therefore the agent would need to balance the consumption of resources with scoring points.

1.2 DARBS

A blackboard system is an artificial intelligence (AI) technique that is analogous to a team of experts who communicate their ideas by writing them on a blackboard (Engelmore and Morgan 1988). The experts are represented by sets of rules, conventional procedures, neural networks, or other program modules. These modules are termed knowledge sources (KS). The blackboard is an area of global memory containing evolving information. The system's current state of understanding of a problem is stored here as it develops from a set of data towards a conclusion. DARBS is a distributed blackboard system developed at the Open University and the Nottingham Trent University (Nolle et al. 2001). The original non-distributed blackboard system was called ARBS and was run on a single processor machine (Hopgood et al. 1998).

DARBS was selected for this work as it is a research-based distributed blackboard system that does not have a central control module. Commercially available distributed blackboard systems were not considered owing to their cost and the limitations on access to their source code.

In DARBS, the blackboard system is modelled on the client/server model where the blackboard (BB) is the server and the KSs are the clients. Therefore, DARBS can be seen as a distributed blackboard system with the blackboard server running on one PC and other KS clients running on different PCs. This allows different KSs to run in parallel and thus to be truly opportunistic (Engelmore and Morgan 1988). For this reason, DARBS may also be considered as a multi-agent collaborative system (Jennings and Wooldridge 1998). Different KS clients on different PCs can work independently, in keeping with the TileWorld test-bed where each agent can be run as a separate KS client

DARBS does not have a control module and as such uses broadcast messages to inform other KSs that a change in a particular partition of the blackboard has occurred. It is then up to the KS to decide what to do. It may either stop what it is doing and check exactly what has changed or it may finish what it is doing and then check.

1.3 Design criteria

One of the benefits of the blackboard architecture is that there is central storage of information about the current problem. All KSs store their working memory on the blackboard, visibly to other KSs or users. New KSs can then be developed that can make use of this information or the information can be used for debugging purposes. A balance is needed, as too little information on the blackboard defeats the purpose of central storage and too much information would create a great communication overhead. Partitioning the information on the blackboard can help, but again there is a balance.

Too much information on a single partition would slow down the blackboard server in searching for the information, and too many partitions would cause the KS clients to monitor more partitions for changes.

The format of the information stored on the blackboard is also important. The more intelligible the information, the greater the amount of data that is used to store it and the more data the communication channel has to send. Also putting the information in a format that is difficult to search or query would slow down the overall system. With all these criteria in mind, TileWorld was designed and implemented on DARBS.

2. DESIGNING TILEWORLD ON DARBS

The natural way the TileWorld test-bed fits a blackboard system made it ideal for DARBS. In the TileWorld environment all the agents and other objects interact with each other and hence the agents can be directly implemented as KS clients and the world itself can reside on the blackboard. This gave all the agents equal access to view and change the world, thus allowing true parallel agent simulations. The setup of the TileWorld on DARBS is as shown in Figure 2. The *Initiator KS* is the KS that sets up the TileWorld with its parameters and generates the world. The *Display TileWorld KS* displays the TileWorld and its contents in a graphical format so that it is easy for users to view the simulation. The display KS also needs to keep track of the changes in the world and make sure that the graphical representation is as up-to-date as possible. Each *Agent KS* controls their respective agent on the TileWorld and makes changes to the world according to the behaviours that are coded in them.

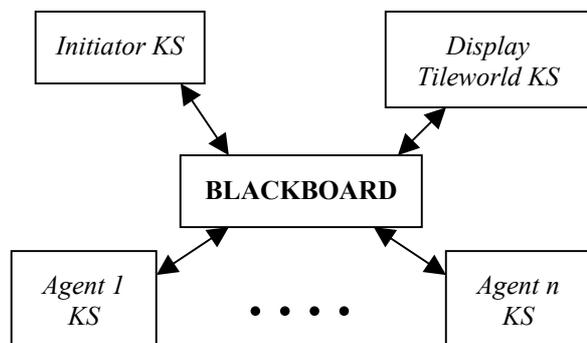


Figure 2: TileWorld on DARBS

To simplify the design, the objects in the TileWorld (i.e. tiles, holes, and obstacles) cannot move themselves. However, it is possible to make the objects in the TileWorld dynamic, i.e. appear and disappear with time by simply adding another KS that uses a probability rate to change the number and position of tiles, holes, and obstacles in the TileWorld. A benefit of the blackboard architecture is that new KSs can easily be added to provide more agents or to change the way the TileWorld is being simulated.

Careful organisation of the data on the blackboard is required to make sure that the agents and all other KSs can interact with each other properly through the blackboard. The organisation of the data also needs to minimise the number of partitions with which a particular KS works. This is to reduce the number of times the KS needs to restart, since a KS restarts whenever a partition that it is working with has changed. As mentioned earlier, the format of the data on the blackboard also needs to be carefully constructed so that queries from the KS clients are as easy, efficient and legible as possible. With these criteria in mind, the blackboard was partitioned as shown in Figure 3. The most important data string format is the one on the TileWorld Environment partition, for example:

[Location 1 , 4 contains Agent 1 , NO HOLE , NO OBSTACLE , NO TILE]
 [Location 1 , 5 contains NO AGENT , Hole 1 , NO OBSTACLE , NO TILE]

[Location 1 , 6 contains NO AGENT , NO HOLE , Obstacle 1 , NO TILE]
 [Location 2 , 4 contains NO AGENT , NO HOLE , NO OBSTACLE , Tile 1]

The order of objects is fixed as Agent-Hole-Obstacle-Tile to facilitate queries from the KSs. If the location contains an agent then “Agent *x*” where *x* is the number of the agent would replace “NO AGENT”, and similarly for holes, obstacles and tiles.

As can be seen in Figure 3, all Agent KSs can access the TileWorld Environment partition and, in theory, can see the whole TileWorld. This could be argued to be an incorrect implementation of the TileWorld test-bed, but it is assumed that all agents are benevolent and will only access those areas of the TileWorld Environment partition that they are intended to.

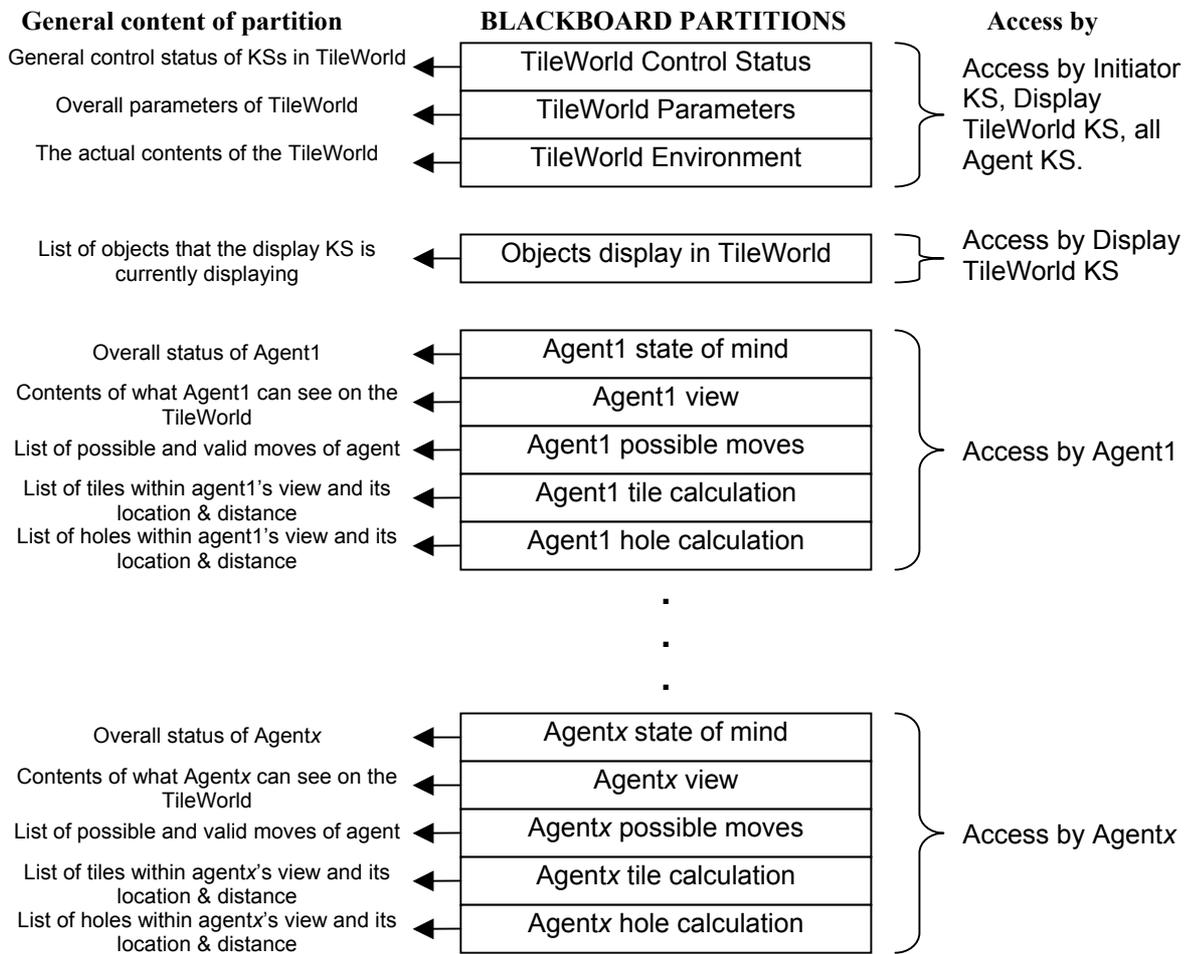


Figure 3: Partitions on the Blackboard

3. IMPLEMENTING THE TILEWORLD

Three types of KSs were implemented in DARBS for this TileWorld test-bed: *Initiator KS*, *Display TileWorld KS*, and *Agent KS*. A screen capture of the TileWorld running on DARBS is shown in Figure 4. Each is implemented as a rule-based KS, described below.

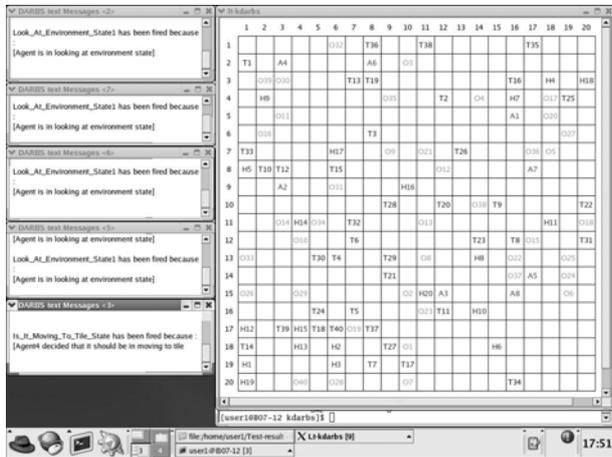


Figure 4: Screen Capture of TileWorld on DARBS

3.1 Initiator KS

The main purpose of the Initiator KS is to setup the parameters of the TileWorld and to generate the world from the parameters. The two main rules of this KS and their functions are:

- Init_TileWorld
 - Set the size of the TileWorld
 - Set the number of agents in the TileWorld
 - Set the number of holes in the TileWorld
 - Set the number of obstacles in the TileWorld
 - Set the number of tiles in the TileWorld
- Create_TileWorld
 - Generate a random position within the TileWorld for each of the agents, holes, obstacles, and tiles.
 - Store this information on the blackboard.

The FIRABILITY_FLAG of this KS is set to true, which means that this KS will run only once unless a change occurs in the partition in which it is interested.

3.2 Display TileWorld KS

The main purpose of the *Display TileWorld KS* is to display the content of the TileWorld in a graphical form. This is done with the help of the Qt library from Trolltech (Dalheimer 2002). The Qt library provides the graphical function calls for drawing and updating the graphical TileWorld with its agents, holes, obstacles and tiles. The rules in this KS and their functions are as follows:

- Display_Initial_Screen
 - Draw the TileWorld grid and label it

- Update_Agent_Display
 - Find the location of all the agents in the TileWorld from the blackboard and display them accordingly.
- Update_Hole_Display
 - Find the location of all the holes in the TileWorld from the blackboard and display them accordingly.
- Update_Obstacle_Display
 - Find the location of all the obstacles in the TileWorld from the blackboard and display them accordingly.
- Update_Tile_Display
 - Find the location of all the tiles in the TileWorld from the blackboard and display them accordingly.
- Update_Total_Objects_Display
 - Find the objects that are currently being display in the TileWorld.
- Update_Deleted_Tile
 - Delete the objects that are no longer on the blackboard from the displayed TileWorld.

The FIRABILITY_FLAG of this KS is also set to true.

3.3 Agent KS

The Agent KSs control the agents in the TileWorld. There is a set of 31 rules for each agent in the TileWorld. The behaviour and intelligence of the agent are coded in the following rules (all rules are listed here, and the function of selected rules is explained):

- Initialise_Agent
- Update_Internal_Status
- Generate_SearchSpace_State
- Look_At_Environment_State1
- Look_At_Environment_State2
- Is_It_Exploring_State
 - Change the agent state to Exploring state if the agent is in Thinking state and the agent is currently carrying no tile and there is no tile within the viewing range OR
 - Change the agent state to Exploring state if the agent is in Thinking state and the agent is currently carrying a tile and there is no hole within the viewing range.
- Is_It_Moving_To_Tile_State
 - Change the agent state to Moving To Tile state if the agent is in Thinking state and the agent is not carrying a tile and there is a tile within the viewing range and the tile is not on the same grid as the agent.
- Is_It_Hole_Filling_State
- Is_It_Moving_To_Hole_State
 - Change the agent state to Moving To Hole state if the agent is in Thinking state and the agent is carrying a tile and there is a hole within the viewing range and the hole is not on the same grid as the agent.
- Is_It_Picking_Up_Tile_State

- Generate_Possible_Moves
- Is_North_Move_Valid
- Is_North_Move_NotValid
- Is_East_Move_Valid
- Is_East_Move_NotValid
- Is_South_Move_Valid
- Is_South_Move_NotValid
- Is_West_Move_Valid
- Is_West_Move_NotValid
- Exploring_State
 - Generate a random step based on the possible moves and the last made move. Store the random generated step on the agent's state of mind partition. Change agent's Exploring state to Making A Move state.
- Moving_To_Tile_State
- Moving_To_Hole_State
- Get_Tile_Distance_State
- Get_Hole_Distance_State
- Find_Closest_Tile_State
- Find_Closest_Hole_State
- Generate_Step_Closer_To_Tile_State
- Generate_Step_Closer_To_Hole_State
- Pick_Up_Tile_State
- Hole_Filling_State
 - Drop the tile it is carrying into the hole it is standing on and recalculate the agent's score. Change agent's state back to Generate Searchspace state.
- Making_Move_State

The FIRABILITY_FLAG of this KS is set to true_always, which means that this KS runs continuously and restarts if a change occurs in the partition in which it is interested.

4. TESTS AND RESULTS

A preliminary test has been carried out on a single Intel Pentium 4 2.8GHz processor with 1GB of DDR RAM running the Red Hat 9 Linux operating system. All the agents in the TileWorld have a viewing radius of two cells. An 18×18 TileWorld was run with 30 obstacles, 15 tiles, and 25 holes. The position of the obstacles, tiles, and holes, and the initial position of the agents were all randomly generated using the C++ standard random number generator function, rand() with a seed of 10. The TileWorld test-bed was run with one, two, three, four, and five agents. For all the runs, the average time taken for an agent to make a move was calculated over 10 moves. The time per move was calculated by subtracting the time of an agent's move from the time of its subsequent move. An agent is considered to make a move when it has changed the TileWorld environment (i.e. moved to another cell, picked up a tile, or dropped a tile into a hole). Restarts due to the TileWorld being changed by other agents are not considered as moves. Figure 5 shows the average

time per move normalised to a single-agent TileWorld's average time per move.

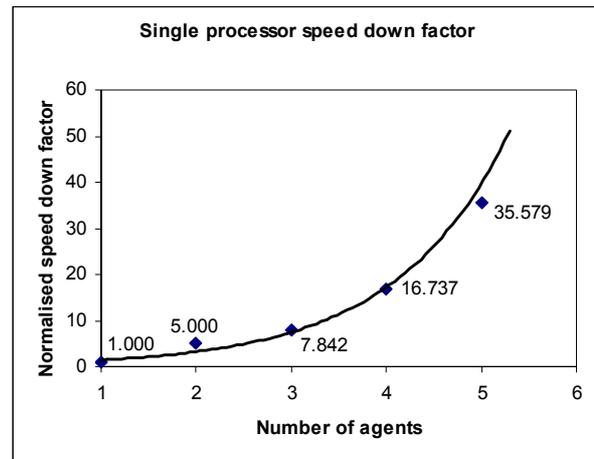


Figure 5: Normalised Single Processor Slow-down Factor

In Figure 5, an exponential function is fitted to the results of the agent simulation on the TileWorld. This shows that the slow down is approximately exponential as the number of agents being simulated increases. This is as expected, as the single processor has to time-slice more processes as the number of agents increases. Another reason for the slow down is that as the number of agents increases, the number of restarts each agent has to make increases, thus increasing the time required for each agent to make a move. This can be improved by having more intelligent restarts, for example by checking whether changes to the partition affect the current thinking of the agent, and if so to what extent, before deciding to restart again.

A multi-processor test is currently in preparation. This is expected to show a less marked slow down in simulation as the number of agents increases. It is inevitable that there will be some slow down as the number of agents increases because there will be communication overheads and access contention between agents for the blackboard. One way of reducing this is to reduce the amount of information stored on the blackboard. However, this would conflict with the concept of openness of information on the blackboard, which provides upgradeability and helps users to understand what is happening in the simulation. Therefore a balance needs to be struck between the amount of information to be stored on the blackboard and the speed of the simulation.

5. CONCLUSIONS AND FUTURE WORK

The TileWorld test-bed has been successfully implemented on DARBS running on a single PC using the Linux operating system. It is expected that the slow performance will improve when this system is run in parallel on separate PCs connected together via TCP/IP as the load of the extra agents would be distributed over

the separate PCs. The speedup factor between n agents in a single processor and n agents in multi-processors can then be compared and evaluated. Further tests will be carried out to investigate the causes of slowness and, where possible, improvements to the system will be made. One cause of slowness is that the KS tries to fire all the rules even when certain rules are known beforehand to be dependent on other rules. A rule dependency table (Hopgood 1994) can be created before runtime to prevent the KS from trying to fire any rule until the rules upon which it depends have fired.

An improvement for this TileWorld test-bed would be to have another KS that relays what the agent sees from the TileWorld Environment partition onto the Agent View partition. In this way, the agent would have no direct access to the TileWorld Environment partition other than making a move on the TileWorld. This would make it easier to implement intelligent restarts as the Agent KS would only need to restart on changes to the Agent View partition and could ignore changes to the TileWorld Environment partition.

Another consideration is the way the KS checks whether a rule's condition is met. The current implementation checks every sub-condition before evaluating whether the composite condition is true. This can be improved on by evaluating the composite condition as it checks each sub-condition. Consider the following example:

```
IF
[
    condition1
    AND
    condition2
    AND
    condition3
]
THEN
[
    actions
]
```

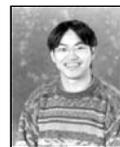
The KS can stop checking *condition2* and *condition3* if *condition1* is false because the composite condition will be false. This type of on-the-fly evaluating would reduce the number of messages sent to the blackboard as each condition-check requires the KS to send a message to the blackboard. The use of on-the-fly evaluation can reduce the communication traffic, thus allowing other KSs to send and receive messages faster.

REFERENCES

- Dalheimer, M.K. 2002. "Programming with Qt". Germany: O'Reilly.
- Engelmore, R.; Morgan, T.; ed. 1988. "Blackboard Systems". Great Britain: Addison Wesley, pp. 2-15.
- Ephrati, E.; Pollack, M. E.; Ur, S. 1995. "Deriving multi-agent coordination through filtering strategies" *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 679-685.

- Hopgood, A.A. 1994. "Rule-based control of a telecommunications network using the blackboard model", *Artificial Intelligence in Engineering*, 9, pp 29-38.
- Hopgood, A.A.; Phillips, H.J.; Picton, P.D.; Braithwaite, N.St.J. 1998. "Fuzzy logic in a blackboard system for controlling plasma deposition processes" *Artificial Intelligence in Engineering*, 12, pp. 253-260.
- Jennings, N. R.; Wooldridge, M. J.; ed. 1998. "Agent Technology: Foundations, Applications, and Markets". Germany: Springer, 1998. pp. 32-34.
- Kinny, D.; Georgeff, M.; Hendler, J. 1992. "Experiments in Optimal Sensing for Situated Agents", *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence*, PRICAI'92, pp. 1176-1182.
- Lees, M.; Logan, B.; Theodoropoulos, G. 2003. "Adaptive Optimistic Synchronisation For Multi-Agent Distributed Simulation", *Proceedings 17th European Simulation Multiconference*, pp. 77-82.
- Nolle, L.; Wong, K. C. P.; Hopgood, A. A. 2001. "DARBS: A Distributed Blackboard System", *Research and Development in Intelligent Systems XVIII*, Bramer, Coenen and Preece (eds.), Springer, pp 161-170.
- Pollack, M. E., and Ringuette, M. 1990. "Introducing the Tileworld: Experimentally Evaluating Agent Architectures", *Proceedings of the Eighth National Conference on Artificial Intelligence*, AAAI Press, pp. 183-189.
- Uhrmacher, A.M.; Schattenberg, B. 1998. "Agents in Discrete Event Simulation", In: Andre Bargiela and Eugene Kerckhoffs (eds.) *Proceedings of the 10TH European Simulation Symposium "Simulation in Industry - Simulation Technology: Science and Art" (ESS'98)*, SCS Publications, Ghent, pp. 129-136.

AUTHOR BIOGRAPHIES



KUM WAH CHOY is a PhD student at the Nottingham Trent University. He obtained his BEng (Hons) in Electronics & Computing in 2001. He has spent a placement year with Xerox Ltd as a Software/Test Engineer. His current research is in the field of distributed embedded systems, agent system, blackboard systems, and artificial intelligence.



ADRIAN HOPGOOD is professor of computing and head of the School of Computing and Technology at the Nottingham Trent University, UK. He is also a visiting professor at the Open University. His main research interests are in intelligent systems and their practical applications. He graduated with a BSc (Hons) in physics from the University of Bristol in 1981 and obtained a PhD from the University of Oxford in 1984. He is a member of the British Computer Society and a committee member for its specialist group on artificial intelligence.



LARS NOLLE graduated from the University of Applied Science and Arts in Hanover in 1995 with a degree in Computer Science and Electronics. After receiving his PhD in Applied Computational Intelligence from The Open University, he worked as a System Engineer for EDS. He returned to The Open University as a Research Fellow in 2000. He joined The Nottingham Trent University as a Senior Lecturer in Computing in February 2002. His research interests include: applied computational intelligence, distributed systems, expert systems, optimisation and control of technical processes.



BRIAN O'NEILL graduated in 1968 from Trinity College, University of Dublin and obtained his PhD from Liverpool University in 1972. Before taking up the lecturing post at Nottingham Trent in 1977, he held research posts at Glasgow and Southampton Universities. His research activities originated from work on electronic instrumentation for powder flow control. This work required the development of fast real-time control and processing, using parallel processing, for its implementation. From this start he has widened his area of research to the design of hardware packet routing switches for inter-processor communication.

CONTINUOUS REINFORCED SNAP-DRIFT LEARNING IN A NEURAL ARCHITECTURE FOR PROXYLET SELECTION IN ACTIVE COMPUTER NETWORKS

Sin Wee Lee
Dominic Palmer-Brown*
Computational Intelligence Research Group
School of Computing
Leeds Metropolitan University
Beckett Park Campus, Leeds LS6 3QS
*Email : [D. Palmer-Brown@leedsmet.ac.uk](mailto:D.Palmer-Brown@leedsmet.ac.uk)

Christopher Roadknight
BTexact Technologies
BT Adastral Park
Martlesham Heath, Ipswich IP5 3RE
and
Computational Intelligence Research Group
School of Computing
Leeds Metropolitan University

KEYWORDS

Computational Intelligence, Artificial Neural Networks, Category Learning, Reinforcement Learning.

ABSTRACT

A new continuous learning method is used to optimise the selection of services in response to user requests in an active computer network simulation environment. The learning is an enhanced version of the 'snap-drift' algorithm, which employs the complementary concepts of fast, minimalist (snap) learning and slower drift (towards the input patterns) learning, in a non-stationary environment where new patterns arrive continually. Snap is based on Adaptive Resonance Theory, and drift on Learning Vector Quantisation. The new algorithm swaps its learning style between these two self-organisational modes when declining performance is detected, but maintains the same learning mode during episodes of improved performance. Performance updates occur at the end of each epoch. Reinforcement is implemented by enabling learning on any given pattern with a probability that increases linearly with declining performance. This method, which is capable of rapid re-learning, is used in the design of a modular neural network system: Performance-guided Adaptive Resonance Theory (P-ART). Simulations demonstrate the learning is stable, and able to discover alternative solutions in rapid response to new performance requirements and significant changes in the stream of input patterns.

INTRODUCTION

The Adaptive Resonance Theory (ART) Network

Developments (Carpenter and Grossberg 1987a) of the original ART (Grossberg, 1976a; Grossberg 1976b) networks include ART1 that self-organises recognition

categories for arbitrary sequences of binary input sequences; and ART2 which does the same for either binary or analogue inputs (Carpenter and Grossberg 1987b). Subsequently, ART3 (Carpenter and Grossberg 1990) has been used to implement parallel searches of compressed or distributed recognition codes (output categories) in a neural network hierarchy. Following the successful implementation of the theory in real-time applications, further development has seen the creation of ART2-A (Carpenter et al. 1991a), which is 2 or 3 orders of magnitude faster than ART2. Fuzzy ART (Carpenter et al. 1991b) the fuzzy extension of ART, incorporated computations from fuzzy set theory. Extensions to ART networks to allow supervised learning were also introduced (Palmer-Brown 1992); and ARTMAP (Carpenter et al. 1991c) and Fuzzy ARTMAP (Carpenter et al. 1992) autonomously learn to classify based on predictive success. Furthermore, there are several other versions of ART network (Tan 1997; Carpenter et al. 1998; Bartfai and White 2000), including supervised multi-layer, self-growing systems (Palmer-Brown 1992).

Limitations

There are limitations of ART networks in non-stationary environments where self-organisation needs to take account of periodic or occasional performance feedback:

- The ART network tends to organize itself into a stable state during fast learning whereby the weights stop changing in the presence of new inputs.
- There is no external feedback to improve the performance of the network when it stabilises with poor performance.

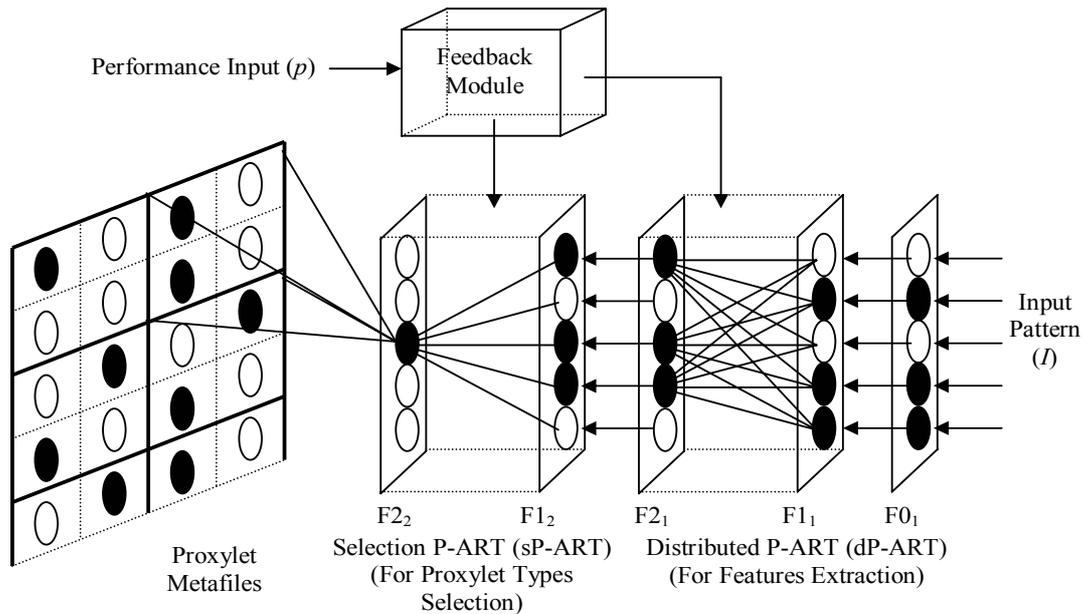


Figure 1: Architecture of the P-ART Network

PERFORMANCE-GUIDED ART (P-ART)

P-ART Architecture

The P-ART network proposed is a modular, multi-layered architecture as shown in Figure 1. It is composed of 3 modules, a Distributed P-ART (dP-ART) network, a Selection P-ART (sP-ART) network and a Kohonen Self-Organising Map. The $F1_1 \leftrightarrow F2_1$ connections of the dP-ART network and $F1_2 \leftrightarrow F2_2$ of the sP-ART are interconnected through weighted bottom-up and top-down connections that can be modified during the learning stage. For clarity, only the connections from the F1 layer to the active (winning) F2 node in each P-ART module are shown. The $F0_1 \rightarrow F1_1$ and two P-ART modules connected through $F2_1 \rightarrow F1_2$ are unidirectional, one to one and non-modifiable. Each of the $F2_2$ nodes is hard-wired onto a specific pre-trained region of the Kohonen Feature map where similar available proxylets (the target outputs) are spatially organised on the 2-D map according to their featural similarity.

Overview of the Operation of the System

On presentation of an input pattern at the input layer $F0_1$, the dP-ART will learn to group the input patterns according to their general features using the novel learning principles developed in this work from the snap-drift' algorithm recently developed (Lee et al. 2002; Lee et al. 2003; Lee et al. 2004). The latest version has several improvements over the previous one in terms of the normalization process, and the synchronization of learning between the s and d P-ARTs; but the two key differences are performance guided toggling of learning between snap and drift, and

the introduction of a probabilistic aspect to enhance reinforcement and stability. The standard matching and reset mechanism of ART (Carpenter and Grossberg 1987a) is retained: If no existing matching prototype is found, i.e. when the stored pattern prototypes are not a good match for the input, the winning $F2_1$ node is reset and another $F2_1$ node is selected. When no corresponding output category can be found, the network considers the input as novel, and generates a new output category node that learns the current input pattern.

The three winning $F2_1$ nodes, whose prototypes are best match to the current input pattern, are used as the input data to the P-ART module for selecting an appropriate output type (called a proxylet in the target application). For the purpose of selecting the required proxylet, the proxylet type information indicated by the P-ART references pre-trained locations on the Kohonen Self-Organising Map (SOM) (Kohonen 1982; Kohonen 1990a), which represent specific proxylets. If the proxylet is unavailable, one of its neighbours is selected (the most similar alternative available).

A non-specific performance measure is used because, as in many applications, there are no specific performance measures (or external feedback) in response to each individual output decision. This measure is used to encourage or discourage reselection of outputs (proxylet types) to occur in order to improve the performance of the neural system. The continuous learning method is the snap-drift algorithm. It involves toggling between snap and drift modes depending on performance changes. Snap and drift are alternative forms of adaptation, and they are described in the next section,

THE LEARNING. The following is a summary of the steps that occurs in P-ART:

- Step 1: Initialise parameters: ($\alpha = 1$, $\sigma = 0$)
- Step 2: For each epoch, t
 Measure or calculate performance in the range $\{0,1\}$ over the last epoch, $P(t)$.
 Performance improvement, $PI = P(t) - P(t-1)$
 Set probability of learning, $PL = 1 - P(t)$
- Step 3: For each new input pattern
 Find the D winning nodes with the largest input (or create new nodes for mismatches)
- Set learn (adapt) true with probability PL.
- If learn is true test learning strategy condition:
 IF ($PI \leq 0$) THEN
 Weights of d-PART adapted according to the alternate learning procedure: (α, σ) becomes Inverse (α and σ) in equation (8) below
 ELSE
 Weights of d-PART adapted according to the same procedure as in the last epoch: (α, σ) unchanged.
- Step 4: Process the output pattern of $F2_1$ as input pattern of $F1_2$
- Find winning node (just one) in $F2_2$.

Weights of s-PART adapted according to the same learning probability and strategy conditions as above, except that the in first half of the learning epoch, both dP-ART and sP-ART learn, whereas in the second half of the epoch, only sP-ART learns. This allows relearning of the mapping from features to selections without the moving target problem of those features changing simultaneously.

THE LEARNING

Snap-Drift

In an environment where new patterns are introduced over time, the learning utilises a novel snap-drift algorithm based on fast, convergent, minimalist learning (snap) and cautious learning (drift) when the performance is good. Snap is based on a modified form of ART; and drift is based on Learning Vector Quantization (LVQ) (Kohonen 1990b). The two forms are combined within a semi-supervised learning system that shifts its learning style whenever it receives a drop in the performance feedback. So, in general terms, the snap-drift algorithm can be stated as:

$$w = \alpha(\text{Fast_Learning_ART}) + \sigma(\text{LVQ}) \quad (1)$$

where α and σ are determined by performance feedback. In previous simulations, α and σ were real

values (Lee et al. 2002; Lee et al. 2003; Lee et al. 2004). In this paper, α and σ are set to (0, 1) or (1, 0) depending on changes in performance, and the learning is then enabled probabilistically.

Input Encoding

A form of coarse coding (Eurich 1997) is used to represent proportional differences between numeric data encoded within the input patterns, e.g. the representation of the value 15 must be closer in input space to the representation of value 20 than that of say 30. The input pattern is arranged in a 25 bit vector. Each property, such as bandwidth, time, file size, loss and completion guarantee, occupies 5 bits of the overall pattern. Table 1 shows the realistic range for each of the request properties. The coding of the user request is performed as illustrated in Table 2, across a different range for the 5 bits in the case of each property. The input patterns are generated by maintaining the coding of each field in turn and randomly generating the codes for rest of the fields for every 20 patterns, giving 1000 patterns in all.

Table 1: Value Ranges of User Request Properties

Properties	Ranges
Bandwidth	10Kb/s \rightarrow 2000Kb/s
Time	1ms \rightarrow 1000ms
Loss	20% \rightarrow 60%
Cost	0.1p \rightarrow 100p
Completion Guarantee	40% \rightarrow 100%

Table 2: Example Coding of Bandwidth in User Requests

Ranges (Kb/s)	User Request
200 \rightarrow 400	10000
800 \rightarrow 1000	01100
1800 \rightarrow 2000	00001

Weights Initialisation

The weights are calculated as floating point and are initialised at the beginning of the simulations. Top-down weights are set randomly to either 0 or 1.

$$w_{ji}(0) = [0,1] \quad (2)$$

Thus, a simple distributed affect will be generated at the output layer of the network, with different patterns tending to give rise to different activations across F_2 from the start. The bottom-up weights w_{ij} are assigned initial values corresponding to the initial values of the top-down weights w_{ji} . This is accomplished by equation (3):

$$w_{ij}(0) = \frac{w_{ji}(0)}{|w_{ji}(0)|} \quad (3)$$

The Distributed P-ART (dP-ART) Learning

On presentation of input pattern, the bottom-up activation is calculated using (4). Then the D number of F2₁ nodes with the highest bottom-up activation, using (5), are selected.

$$T_J = \sum |w_{ij} \cap I| \quad (4)$$

$$T_J = \max\{T_J | J = 1, 2, \dots, M\} \quad (5)$$

D is set to 3 in this application. If the distributed output categories are found with the required matching level, the three F2₁ nodes will enter into resonant state and learn using (6):

$$w_{ji}^{(new)} = \alpha(I \cap w_{ji}^{(old)}) + \sigma(w_{ji}^{(old)} + \beta(I - w_{ji}^{(old)})) \quad (6)$$

where w_{ji} = top-down weights vectors; I = binary input vectors, and β = the drift speed constant = 0.5.

When $\alpha = 1$, (6) can be simplified to:

$$w_{ji}^{(new)} = (I \cap w_{ji}^{(old)}) \quad (7)$$

This invokes fast minimalist learning, causing the top-down weights to reach their new asymptote on each input presentation:

$$w_J \rightarrow I \cap w_J^{(old)} \quad (8)$$

In contrast, when $\sigma = 1$, (6) simplifies to

$$w_{ji}^{(new)} = (w_{ji}^{(old)} + \beta(I - w_{ji}^{(old)})) \quad (9)$$

This causes a simple form of clustering or LVQ at a speed determined by β . As describe in the pseudo code presented in **Operation of the System** above, learning is a combination of the two forms of adaptation, because the mode is toggled between snap and drift whenever performance has deteriorated during the previous epoch. In addition, whether adaptation occurs or not on a given pattern is a probabilistic decision, whereby the probability of the snap or drift occurring is proportional to declining performance. The novel bottom-up learning of the P-ART is a normalised version of the top-down learning:

$$w_{ij}^{(new)} = \frac{w_{ji}^{(new)}}{|w_{ji}^{(new)}|} \quad (10)$$

where $w_{ji}^{(new)}$ = top-down weights of the network after learning. Poor performance can occur when the final

selection of proxylet type is wrong, *even if* the general feature extracted by dP-ART is valid. To cope with this, the dP-ART learning is toggled on-off every half-epoch so that sP-ART can readjust its learning of selections without modification of the general features in dP-ART, thus resolving a moving target problem.

The Selection P-ART (sP-ART) Learning

The outputs produced by the dP-ART act as input to the sP-ART. The behaviour of sP-ART is the same as that described in section **P-ART Architecture**, with one exception; only the F2 node with the highest activation is adapted. Each output node of the sP-ART points to a set of available application-specific groupings (in this case proxylet types). The proxylet type data, containing attributes of the types, is used as off-line training data for the SOM so that it forms a map with similar proxylets placed on adjacent nodes. This allows each output node of the sP-ART to be 'hardwired' onto regions of the SOM. The task of the sP-ART is therefore to learn to associate the correct group of input patterns with an output node that is hardwired to a region of the SOM. The effect of learning and relearning within the sP-ART module is that specific output nodes will relate different groups of input patterns to different regions of the SOM until the performance feedback indicates that it is indexing the SOM regions that select the most appropriate proxylets. In that event, the learning probability is low, so that even if the snap-drift has not yet converged, further adjustment is slow.

The Performance Feedback

The external performance feedback into the PART reflects the performance requirement in different circumstances. Various performance feedback profiles in the range {0, 1} are fed into the network to evaluate the dynamic stability and effectiveness of the learning. Initially, some very basic tests with performances of 1 or 0 were evaluated in a simplified system (Lee et al. 2002; Lee et al. 2003; Lee et al. 2004). Below, the simulations involve computing the performance based on a parameter associated with the winning output neuron. Ultimately, a realistic commercial external performance feedback criteria will be established, which will be obtained from BT, to evaluate the improvement in performance of the network learning under realistic external performance feedback. Factors which affect performance include latencies for request with differing time to live, dropping rate for requests with differing time to live, and different charging level with related Quality of Service.

BRITISH TELECOM (BT) APPLICATION

Application Layer Active Network (ALAN)

British Telecom (BT) is the main data network provider in the UK. At present, most applications are run on edge devices (which send and receive data, but do not route third party data), such as servers, PCs and WAP enabled devices. There are strong arguments for moving as many of these applications as possible into the network (Tennenhouse and Wetherall 1996), thereby ensuring optimal placement of applications with respect to performance, version synchronicity (so that more users have the same version), and increased security. 'Active Networking' (Tennenhouse and Wetherall 1996) aims to achieve this application migration into the network by running code within the network on specialised routers. It gives users the ability to load software components onto network devices dynamically without explicit references to any third party. The ALAN architecture (Fry and Ghosh 1999) enabled the user to supply JAVA based active-service codes known as proxylets that run on a network device. Each networked server runs the 'Execution Environments for Proxylets' (EEPs) that contains the user supplied software. The purpose of the architecture is to locate the software at optimal points of the end-to-end path between the server and the clients.

Automated Active Network Management using Distributed Genetic Algorithm (GA)

The original ALAN proposal, the management system supports conventional management agent interfaces (Marshall 1999; Marshall et al. 2000) that respond to instructions from the system operators. Each application is individually placed in the network. However, since ALAN with the potential for an enormous range of services, it is necessary to combine the active services with an automated and adaptive management solution. Recently, a novel adaptive approach, a Distributed Genetic Algorithm (GA) solution was introduced by BT Research Laboratories (Marshall and Roadknight 2001). It performs proxylet placement. Here, P-ART provides a means of finding a set of conditions that produce optimum proxylet selection in an EEP containing the frequently requested proxylets that have been placed. Continuous performance guided adaptation of the mapping of input patterns, which contain the main attribute values of user proxylet requests, performs intelligent proxylet type selection.

THE P-ART SIMULATION

P-ART is used for learning and mapping user requests onto appropriate proxylets. The test patterns consist of 1000 input vectors. Each test pattern characterizes the properties of a network request, such as bandwidth, time, file size, loss and completion guarantee. These test patterns are presented in random order with 10 patterns per epoch for 100 epochs where the

performance, p , is calculated according to the average bandwidth of selections. This on-line continuous random presentation of test patterns simulates the real world scenario in which pattern presentation order is random, so that a given network request might be repeatedly encountered while others are not used at all.

RESULTS & CONCLUSIONS

Results are presented in Figure 2, 3 and Table 3 and 4. They are representative (typical) of many simulations that have been run. Performance feedback is updated at the end of each epoch of 10 patterns. Much longer epochs are less effective. The best results are for the shortest epochs for which the performance estimate remains a reasonable estimate of overall performance, which of course it would not be for a very small number of patterns. In this application, although there are 1000 patterns, there are only 100 general types, and hence 10 is approximately the smallest reasonable sample for which updates in performance may be trusted to increase or decrease with true overall performance. This will clearly be different for each application. A key observation behind the results presented in the tables and graphs here is that learning is actually over by epoch 64, after which no new selections of proxylets occur until the criteria change to low bandwidth. After 64 (640 pattern presentations), all the performance variation between epochs (the jitter in the performance curve) is due to the epochs being short (in other words, samples of 10 give approximately 70% accurate performance values), and hence the performance over 1000 patterns is actually constant at about the average of the values of the table values from 70-100, which is just under 70%.

In Table 4 and Figure 3 the performance criterion is swapped from high to low bandwidth after 100 epochs, and we see relearning and re-stabilisation occur, with similar results to above once convergence has occurred.

In conclusion, learning stabilizes reliably and is able to map the inputs onto appropriate proxylets. The simulations have shown that the snap-drift algorithm is able to provide continuous probabilistic real-time learning in order to improve the performance, based on the external performance feedback.

Further work is applying the same methods in other areas, but we are also working on attributing performance to each output neuron separately, so that the approach can be applied to classification tasks.

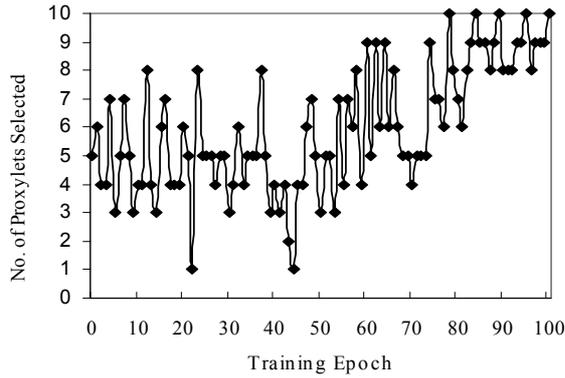


Figure 2: The Selection Frequency of the Proxylet Type. E.g. Bandwidth Bands: Low Bandwidth Proxylet: 0 → 1000 Kb/S and High Bandwidth Proxylet Type: 1001 → 2000 Kb/S

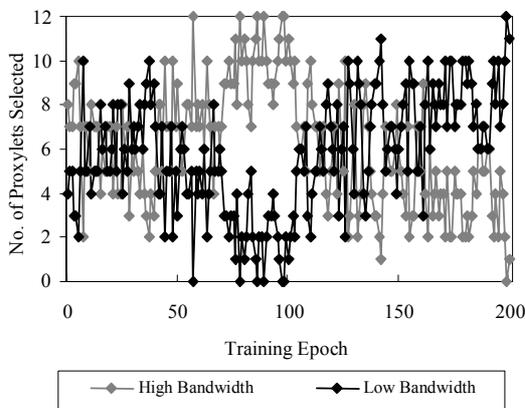


Figure 3: The Selection Frequency of Proxylet Type.

Table 3: Performance of P-ART

Epoch	Average No. of High Bandwidth Proxylet Selected (/10)	Performance (%)
0 - 9	4.9	49
30 - 39	4.8	48
40 - 49	4.0	40
50 - 59	5.2	52
70 - 79	6.6	66
80 - 89	8.5	85
90 - 99	8.7	87

Table 4: Performance of P-ART with Switching of Performance Criterion

Epoch	Average No. of High B/W Proxylet Selected (/12)	Average No. of Low B/W Proxylet Selected (/12)	High B/W Proxylet Selection (%)	Low B/W Proxylet Selection (%)
0 - 9	6.08	3.92	50.69	32.64
10 - 19	5.25	4.75	43.75	39.58
30 - 39	3.50	6.50	29.17	54.17
40 - 49	6.00	4.00	50.00	33.33
50 - 59	6.33	3.67	52.78	30.56
60 - 69	5.83	4.17	48.61	34.72
70 - 79	7.83	2.17	65.28	18.06
80 - 89	8.42	1.58	70.14	13.19
90 - 99	8.33	1.67	69.44	13.89
100 - 109	6.67	3.33	55.56	27.78
110 - 119	5.17	4.83	43.06	40.28
130 - 139	4.42	5.58	36.81	46.53
150 - 159	3.75	6.25	31.25	52.08
160 - 169	2.83	7.17	23.61	59.72
170 - 179	3.42	6.58	28.47	54.86
180 - 189	2.83	7.17	23.61	59.72
190 - 199	0.08	9.92	0.69	82.64

REFERENCES

- Bartfai, G. and R. White. 2000. "Incremental Learning and Optimization of Hierarchical Clustering with ART-based Modular Networks." In *Innovations in ART Neural Networks*, L. C. Jain, B. Lazzerini and U. Halici (Eds.). Physica-Verlag, 87 - 132.
- Carpenter, G. A. and S. Grossberg. 1987a. "A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition Machine," *Computer Vision, Graphics and Image Processing*, Vol. 37, 54-115.
- Carpenter, G. A. and S. Grossberg. 1987b. "ART2: Self-Organization of Stable Category Recognition Codes for Analogue Pattern." *Applied Optics*, Vol. 26, 4919 - 4930.
- Carpenter, G. A. and S. Grossberg. 1990. "ART 3: Hierarchical Search Using Chemical Transmitter in Self-Organizing Pattern Recognition Architectures." *Neural Network*, Vol. 3, No. 4, 129 - 152.
- Carpenter, G. A.; S. Grossberg and D.B. Rosen. 1991a. "ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition." *Neural Networks*, Vol. 4, 493 - 504.
- Carpenter, G. A.; S. Grossberg and D.B. Rosen. 1991b. "Fuzzy ART: Fast Stable Learning and Categorization of Analogue Pattern by an Adaptive Resonance System." *Neural Networks*, Vol. 4, 759 - 771.
- Carpenter, G. A.; S. Grossberg and J. H. Reynold. 1991c. "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Networks." *Neural Networks*, Vol. 4, 565 - 588.
- Carpenter, G. A.; S. Grossberg; A. Markuzon; J. H. Reynold and D. B. Rosen. 1992. "Fuzzy ARTMAP: A Neural Network Architecture for Incremental

- Supervised Learning of Analogue Multidimensional Maps." *IEEE Transactions in Neural Networks*, Vol. 3, No. 5, 698 – 713.
- Carpenter, G. A.; B. Milenova and B. Noeske. 1998. "dARTMAP: A Neural Network for Fast Distributed Supervised Learning." *Neural Networks*, Vol. 11, 793 – 813.
- Eurich, C. W.; H. Schwegler and R. Woesler. 1997. "Coarse Coding: Applications to the Visual System of Salamanders." *Biological Cybernetics*, Vol. 77, 41 – 47.
- Fry, M. and A. Ghosh. 1999. "Application Layer Active Network." *Computer Network*, Vol. 31, No. 7, 655 – 667.
- Grossberg, S. 1976a. "Adaptive Pattern Classification and Universal Recoding. I. Parallel Development and Coding of Neural Feature Detectors." *Biological Cybernetics*, Vol. 23, 121 – 134.
- Grossberg, S. 1976b. "Adaptive Pattern Classification and Universal Recoding. II. Feedback, Expectation, Olfaction, and Illusions." *Biological Cybernetics*, Vol. 23, 187 – 202.
- Kohonen, T. 1982. "Self-Organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics*, Vol. 43, 53 – 69.
- Kohonen, T. 1990a. "The Self-Organizing Maps." In *Proceeding of IEEE*, Vol. 78, No. 9, 1464 – 1480.
- Kohonen, T. 1990b. "Improved Versions of Learning Vector Quantization." In *Proceeding of the International Joint Conference on Neural Networks*, Vol. 1, 545 – 550.
- Lee, S. W.; D. Palmer-Brown; J. Tepper and C. M. Roadknight. 2002. "Performance-guided Neural Network for Rapidly Self-Organising Active Network Management." In *Soft Computing Systems: Design, Management and Applications*, A. Abraham J. Ruiz-del-Solar and M. Köppen (Eds.). IOS Press, Amsterdam, 21 – 31.
- Lee, S. W.; D. Palmer-Brown; J. Tepper and C. M. Roadknight. 2003. "Snap-Drift: Real-Time Performance-guided Learning." In *Proceeding of International Joint Conference on Neural Networks*, Vol. 2, 1412 – 1416.
- Lee, S. W.; D. Palmer-Brown and C. M. Roadknight. 2004. "Performance-guided Neural Network for Rapidly Self-Organising Active Network Management." *Accepted for Neurocomputing*.
- Marshall, I. W. 1999. "Application Layer Programmable Internetwork Environment." *British Telecom Technology Journal*, Vol. 17. No. 2, 82 – 94.
- Marshall, I. W.; J. Hardwicke; H. Gharid; M. Fisher and P. Mckee. 2000. "Active Management of Multi-Service Networks." In *Proceeding of the IEEE Network Operations and Management Symposium (Hawaii)*. 981 – 983.
- Marshall, I. W. and C. M. Roadknight. 2001. "Provision of Quality of Service for Active Services." *Computer Networks*, Vol. 36, No. 1, 75 – 85.

- Palmer-Brown, D. 1992. "High Speed Learning in a Supervised, Self Growing Net." In *Proceeding of the International Conference of Artificial Neural Network*, Vol.2, 1159 – 1162.
- Tan, A-H. 1997. "Cascade ARTMAP: Integrating Neural Computation and Symbolic Knowledge Processing." *IEEE Transactions in Neural Networks*, Vol. 8, No. 2, 237 – 250.
- Tennenhouse, D. and D. Wetherall. 1996. "Towards An Active Network Architecture." *Computer Communication Reviews*, Vol. 26, No. 2, 5 – 18.

AUTHOR BIOGRAPHIES



SIN WEE LEE was born in Melaka, Malaysia, in 1976. He graduated with first class honours in electronics and computing engineering from the Nottingham Trent University, United Kingdom, in 1999. His PhD's thesis focuses on the development of

performance-guided neural network for active network management. From 2000 to 2001, he was a systems engineer at Malaysia Multimedia University in Malaysia. In December 2001, he joined the School of Computing, Leeds Metropolitan University, Leeds, United Kingdom, with a research scholarship from EPSRC/BT Research Laboratories in Neural Networks. As a research assistant, he works with Dominic Palmer-Brown, on the improvement and development of connectionist language processing, improvements to the snap-drift algorithm and its applications.



DOMINIC PALMER-BROWN is Professor of Neurocomputing and the leader of the Computational Intelligence Research Group, in the School of Computing at Leeds Metropolitan University, UK. The Group have active research links with several organisations,

including British Telecom Research Labs, The Centre for Ecology and Hydrology, and with other universities. A key focus of our research is Neurocomputing and related methods of adaptation and learning in cognitive science, intelligent data analysis, and pattern recognition. Dominic has published over 40 international conference and journal papers and supervised 10 PhDs since 1993, having completed his own PhD on an adaptive resonance classifier in 1991. His interests have principally concerned supervised and performance-guided ART, enhanced MLPs for intelligent data analysis, and architectures incorporating MLPs and SRNs for thematic knowledge extraction and natural language processing. He was editor of the review journal *Trends in Cognitive Sciences* during 2000-2 before rejoining Leeds Met.

BOOLEAN SYMMETRY FUNCTION SYNTHESIS BY MEANS OF ARBITRARY EVOLUTIONARY ALGORITHMS - COMPARATIVE STUDY

Ivan Zelinka, Zuzana Oplatkova
Institute of Control Processing and
Information Technologies
Faculty of Technology
Tomas Bata University in Zlin
Mostni 5139 Zlin, Czech Republic
{zelinka,oplatkova}@ft.utb.cz

Lars Nolle
School of Computing and Mathematics
The Nottingham Trent University
Burton Street
Nottingham, NG1 4BU, UK
Lars.nolle@ntu.ac.uk

KEYWORDS

symbolic regression, genetic programming, grammar evolution, analytic programming, optimisation, SOMA

ABSTRACT

This contribution introduces analytical programming, a novel method that allows solving various problems from the symbolic regression domain. Symbolic regression was firstly proposed by J. R. Koza in his genetic programming and by C. Ryan for grammatical evolution. This contribution explains the main principles of analytic programming, and demonstrates its ability to synthesise suitable solutions, called programs. It is then compared with genetic programming and grammatical evolution. This comparative study is concerned with three Boolean k-symmetry problems from Koza's genetic programming domain, which are solved by means of analytical programming. Here, two evolutionary algorithms are used with analytical programming: differential evolution and self-organizing migrating algorithm.

INTRODUCTION

The term *symbolic regression* (SR) represents a process in which measured data is fitted by a suitable mathematical formula like $x^2 + C$, $\sin(x)+1/e^x$, etc. Amongst mathematicians, this process is quite well known and can be used when data of an unknown process can be obtained. For long time, SR was only the domain of humans but for the few last decades it has also become the domain of computers. Today there are two methods, which can be used for SR by means of computers. The first one is called genetic programming (GP) (Koza 1998), (Koza et al 1999) and the second one is grammatical evolution (O'Neill and Ryan 2002), (Ryan et al.1998).

The idea of how to solve various problems using SR by the means of evolutionary algorithms (EA) was introduced by from John Koza who used genetic algorithms (GA) for GP. Genetic programming is basically a symbolic regression, which is done by using evolutionary algorithms instead of a human being. The ability to solve really hard problems was proved many

times, and hence, GP today performs so well that it can be applied, e.g. to synthesise highly sophisticated electronic circuits (Koza et al. 2003).

In the last decade of the 20th century, a novel method for SR was developed by C.Ryan, which is called grammatical evolution (GE). Grammatical evolution can be regarded as an unfolding of GP, because of some common principles, which are the same for both algorithms. One important characteristics of GE is, for example, the fact that GE can be implemented in any arbitrary computer language compared with GP, which is usually done in LISP. In contrast to other evolutionary algorithms, GE was used only with a few search strategies, with a binary representation of the populations (O'Sullivan and Ryan, 2002). Another interesting investigation using symbolic regression was carried out by (Johnson) working on Artificial Immune Systems.

In this paper, a novel method is presented which was developed, called analytical programming (AP) (Zelinka 2002 a), (Zelinka 2002 b), (Zelinka and Oplatkova, 2003) and (Zelinka and Oplatkova, 2004). AP is also a tool for symbolic regression, based on different principles compared to GP and GE. The important principles of AP, together with tests and a comparison with GP, are documented in this contribution.

ANALYTIC PROGRAMMING

Term *analytic programming* was coined by the authors of this article as a matter of simplicity: Because it is possible to use almost any evolutionary algorithm for AP, each EA used for the new approach would add its name to the emerging algorithm, e.g. SOMA programming, DE programming, SA programming etc. This clearly would be confusing and complicated. Analytic programming indicates the use of an EA for analytic solutions synthesis (i.e. symbolic regression). That is the main reason for choosing the term 'analytic programming'.

Analytic programming was inspired by the methods of variations in Hilbert spaces and by GP. The

principles of AP are somewhere between these two philosophies: From GP stems the idea of the evolutionary creation of symbolic solutions, whereas the ideas of functional spaces and the building of resulting function by means of search process (usually done by numerical methods like the Ritz or Galerkin method) is adopted from Hilbert spaces. AP is based, as well as GP, on a set of functions, operators and so-called terminals, which are usually constants or independent variables, for example:

- functions: Sin, Tan, Tanh, And, Or
- operators: +, -, *, /, dt, ...
- terminals: 2.73, 3.14, t, ...

All these ‘mathematical’ objects create a set from which AP tries to synthesise an appropriate solution. The main principle of AP is based on discrete set handling, proposed in (Zelinka 2004) (see Figure 1 and Figure 2). Discrete set handling itself can be seen as a universal interface between EA and the problem to be solved symbolically. That is why AP can be carried out using almost any evolutionary algorithm. Analytical programming, together with a few basic examples, is for example discussed in more detail in (Zelinka and Oplatkova, 2003).

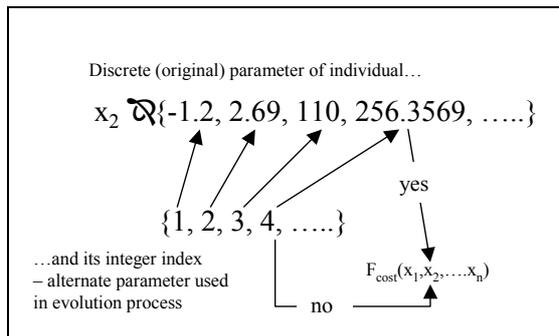


Figure 1: Discrete set handling.

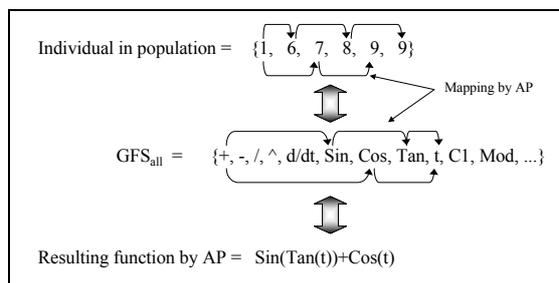


Figure 2: Main principle of AP.

Briefly said, in AP, individual consist of non-numerical expressions (operators, functions,...) as described above, which are in the evolutionary process represented by their integer indexes (Figure 1 and 2). This index then serves like a pointer into the set of expressions and AP uses it to synthesise the resulting

function-program for cost function evaluation (Zelinka 2004). Analytic programming exists so far in three versions. All three versions use for program synthesis the same sets of functions, terminals, etc., as Koza uses in GP (Koza 1998), (Koza et al 1999). The second version (AP_{meta}, lets call first version AP_{basic}) is modified in the sense of constant estimation. For example, Koza uses for the so-called sextic problem (Koza 1998) randomly generated constants, whereas AP here use only one, called K which is inserted into a formula at various places by the evolutionary process. When a program is synthesised, then all K 's are indexed so that K_1, K_2, \dots, K_n , are in the formula obtained, and then all K_n are estimated using a second evolutionary algorithm. Because EA (slave) “works under” EA (master, i.e. EA_{master} ► program ► K indexing ► EA_{slave} ► estimation of K_n) then this version is called AP with metaevolution - AP_{meta}. Because this version is quite time consuming, AP_{meta} was modified to the third version, which differ from the second one in the estimation of K . This is done by using a suitable method for non-linear fitting (AP_{nr}). This method has shown the most promising performance when unknown constants are present, results of some comparative simulations can be found in (Zelinka and Oplatkova, 2003). For the simulations described here, AP_{basic} was used.

PROBLEM DESIGN

Problem selection

The class of booleans k-symmetry problems was chosen for this comparative study, based on case studies reported in (Zelinka and Oplatkova 2003) and (Zelinka and Oplatkova 2004), namely 3-symmetry, 4-symmetry and 5-symmetry problems. In general, boolean symmetry problems mean that if input values to a system are symmetric, then the output is True. If the input is not symmetrically then the output is False. The number of all possible inputs (combinations) is 8 (3-symmetry), 16 (4-symmetry) and 32 (5-symmetry). An example of the truth table for 3-symmetry problems is given in Table 1. For truth tables for other problems solved by AP see www.ft.utb.cz/people/zelinka/ap.

Table 1: Truth table for Boolean 3-symmetry problem according to (Koza 1998)

Input 1	Input 2	Input 3	Output
True	True	True	True
True	True	False	False
True	False	True	True
False	True	True	False
True	False	False	False
False	True	False	True
False	False	True	False
False	False	False	True

All simulations were based on a set of logic functions And, Nand, Or, Nor, and the needed number of inputs A, B, C, ...

The Fitness Function

The fitness (cost function) has been calculated using the Hamming distance between truth table output and synthesised candidate program (1). The theoretical maximum value (the worst solution of all) of this cost function is 8 for a 3-symmetry problem, 16 for a 4-symmetry problem, and 32 in case of 5-symmetry problems. The minimal value (the best solution) is 0 for all k-symmetry problems. The aim of all the simulations was to find the best solution, i.e. a solution that returns the cost value 0. For numerical calculations, False and True were replaced by 0 and 1.

$$f_{\text{cost}} = \sum_{i=1}^{2^n} |TT_i - P_i|$$

(1)

TT_i - i^{th} output of truth table

P_i - i^{th} output of synthesised program

Optimisation Algorithm Used

For the experiments described here, stochastic optimisation algorithms, such as Differential Evolution (DE) (Price 1999) and SelfOrganizing Migrating Algorithm (SOMA) (Zelinka 2004), had been used. Alternative algorithms, like Genetic Algorithms (GA) and Simulated Annealing (SA), are now in process, and results are hoped to be presented soon. For an exact description of the algorithms use see (Price 1999) for DE and (Zelinka 2004) for SOMA.

EXPERIMENTAL RESULTS

Both algorithms (SOMA, DE) have been applied 50 times in order to find the optimum of all boolean k-symmetry problems. The primary aim of this comparative study is not to show which algorithm is better and worst, but to show that AP can be really used for different problems of symbolic regression by different EAs (based also on previous comparative studies and case studies (Zelinka 2002 a), (Zelinka 2002 b), (Zelinka and Oplatkova, 2003) and (Zelinka and Oplatkova, 2004)).

The control parameter settings have been found empirically and are given in Table 2 (SOMA) and Table 3 (DE). The main criterion for this setting was to keep the same setting of parameters as much as possible and of course the same number of cost function evaluations as well as population size (parameter PopSize for SOMA, NP for DE).

Outputs of all simulations are depicted in Figures and numerically reported in Tables 4 - 9. Figures 3, 4, 6, 7, 9 and 10 show results of all 50 simulations for each k-symmetry problem. Figures 5, 8 and 11 show a mutual comparison of algorithm performance in the point of

view of the number of cost function evaluations. Length of synthesised programs is in tables marked as 'leaf count' (LC) – leafs are the elements of a program, i.e. And, Nor, Or, Input_A, etc.

Examples of typical solutions synthesised by both algorithms is represented by formulas (2) for a 3-symmetry problem, (3) for a 4-symmetry problem, and (4) for a 5-symmetry problem.

$$((A\bar{v}B)\bar{\wedge}(C\wedge B))\bar{\wedge}((C\bar{v}A)\bar{v}C\wedge C\wedge A)\bar{v}(((C\wedge A)\bar{\wedge}(C\wedge A))\bar{\wedge}(C\wedge A\wedge A))\bar{\wedge}((C\wedge A\wedge C\wedge A)\bar{\wedge}A)$$

(2)

$$(D\bar{\wedge}D\bar{v}(B\bar{\wedge}(D\wedge C\wedge B\wedge A\wedge v-C\wedge-B\wedge D\wedge A\wedge v-D\wedge-A\wedge C\wedge B)\bar{v}D))\wedge C\bar{v}(((D\wedge C\wedge B\wedge A\wedge v-C\wedge-B\wedge D\wedge A\wedge v-D\wedge-A\wedge C\wedge B)\wedge B)\bar{\wedge}(D\wedge C\wedge B\wedge A\wedge v-C\wedge-B\wedge D\wedge A\wedge v-D\wedge-A\wedge C\wedge B))\bar{\wedge}(B\bar{v}C))\wedge((A\bar{v}D)\bar{v}(D\wedge C\wedge B\wedge A\wedge v-C\wedge-B\wedge D\wedge A\wedge v-D\wedge-A\wedge C\wedge B)\wedge(D\wedge C\wedge B\wedge A\wedge v-C\wedge-B\wedge D\wedge A\wedge v-D\wedge-A\wedge C\wedge B))$$

(3)

$$((B\bar{v}E)\bar{\wedge}D\bar{v}((A\bar{v}A)\bar{v}(E\wedge D\wedge B\wedge A\wedge v\bar{v}E\wedge D\wedge C\wedge A\wedge v-B\wedge E\wedge C\wedge A\wedge v-D\wedge-B\wedge E\wedge A\wedge v-E\wedge-A\wedge D\wedge B\wedge v-E\wedge-D\wedge-B\wedge-A)\bar{\wedge}B))\bar{v}((D\bar{\wedge}(E\wedge D\wedge B\wedge A\wedge v\bar{v}E\wedge D\wedge C\wedge A\wedge v-B\wedge E\wedge C\wedge A\wedge v-D\wedge-B\wedge E\wedge A\wedge v-E\wedge-A\wedge D\wedge B\wedge v-E\wedge-D\wedge-B\wedge-A))\wedge(B\bar{v}B)\bar{v}(E\bar{\wedge}B)\wedge((E\wedge D\wedge B\wedge A\wedge v\bar{v}E\wedge D\wedge C\wedge A\wedge v-B\wedge E\wedge C\wedge A\wedge v-D\wedge-B\wedge E\wedge A\wedge v-E\wedge-A\wedge D\wedge B\wedge v-E\wedge-D\wedge-B\wedge-A)\bar{v}B))$$

(4)

Table 2: SOMA setting for Boolean k-symmetry problems, k=3,4,5

	3	4	5
PathLength	3	3	3
Step	0.11	0.11	0.3
PRT	0.1	0.1	0.1
PopSize	300	300	300
Migrations	30	30	85
MinDiv	-0.1	-0.1	-0.1
Individual Length	30	30	30

Table 3: DE setting for Boolean k-symmetry problems, k=3,4,5

	3	4	5
NP	300	300	300
F	0.8	0.8	0.8
CR	0.2	0.2	0.2
Generations	800	800	2000
Individual Length	30	30	30

Table 4: Boolean 3-symmetry problem by SOMA

	Cost Function Evaluations	Leaf Count
Minimum	202	31
Average	4769	81
Maximum	14991	165

Table 5: Boolean 3-symmetry problem by DE

	Cost Function Evaluations	Leaf Count
Minimum	79	28
Average	2439	83
Maximum	8130	142

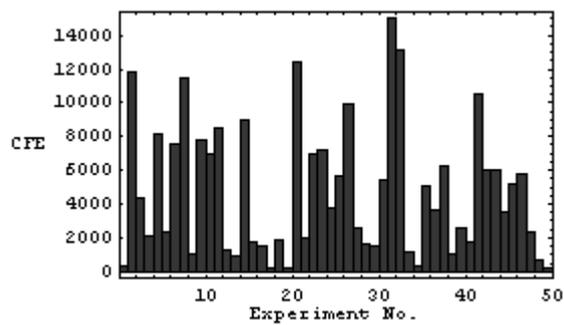


Figure 3: 3-symmetry by SOMA for 50 successful hits out of 50.

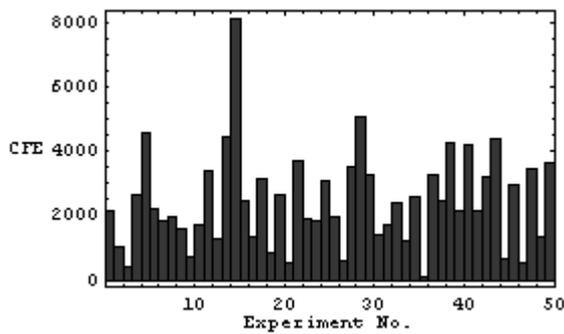


Figure 4: 3-symmetry by DE for 50 successful hits out of 50.

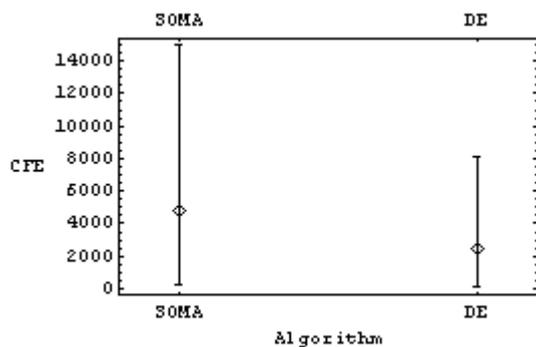


Figure 5: Algorithm performance of 3-symmetry by SOMA and DE.

Table 6: Boolean 4-symmetry problem by SOMA

	Cost Function Evaluations	Leaf Count
Minimum	23427	81
Average	84872	130
Maximum	185732	230

Table 7: Boolean 4-symmetry problem by DE

	Cost Function Evaluations	Leaf Count
Minimum	13696	50
Average	47790	133
Maximum	114709	206

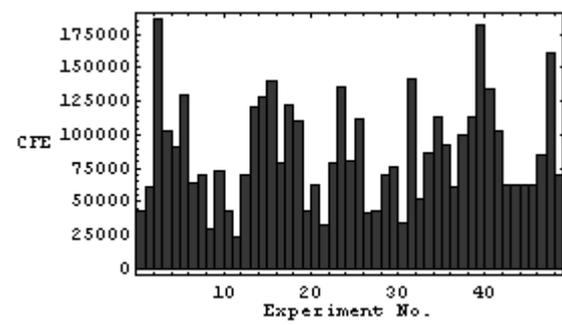


Figure 6: 4-symmetry by SOMA for 49 successful hits out of 50.

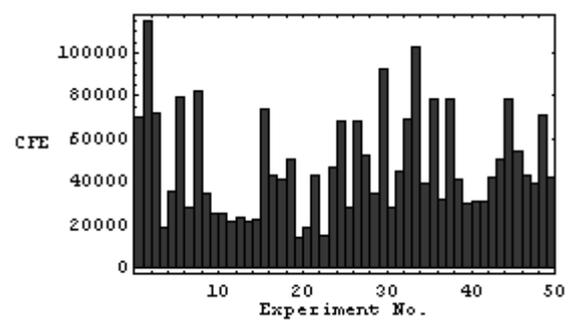


Figure 7: 4-symmetry by DE for 50 successful hits out of 50.

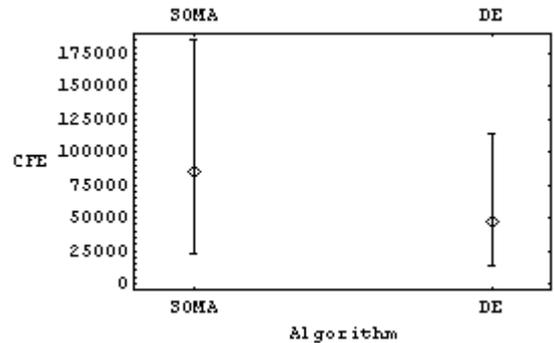


Figure 8: Algorithm performance of 4-symmetry by SOMA and DE

Table 8: Boolean 5-symmetry problem by SOMA

	Cost Function Evaluations	Leaf Count
Minimum	40029	48
Average	119903	170
Maximum	225972	284

Table 9: Boolean 5-symmetry problem by DE

	Cost Function Evaluations	Leaf Count
Minimum	58273	70
Average	172084	176
Maximum	363432	369

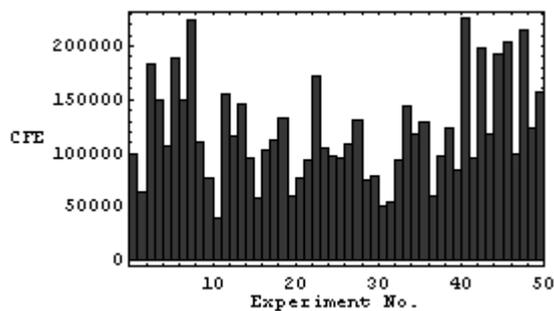


Figure 9: 5-symmetry by SOMA for 50 successful hits out of 50.

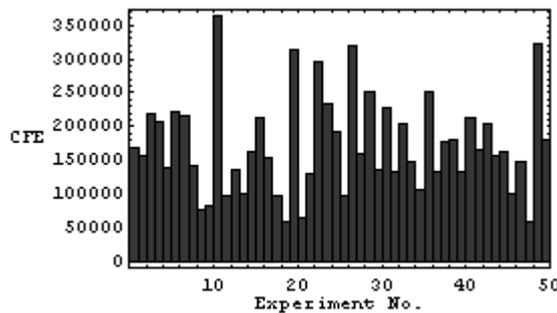


Figure 10: 5-symmetry by DE for 50 successful hits out of 50.

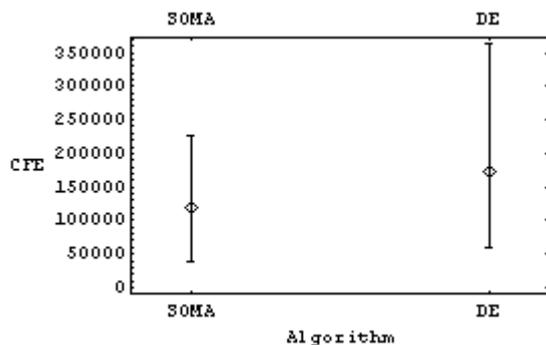


Figure 11: Algorithm performance in 5-symmetry by SOMA and DE

CONCLUSIONS

The method of analytic programming described here is relatively simple, easy to implement and easy to use. Based on its principles and its universality (it was tested with 4 evolutionary algorithms – SA, GA, SOMA and DE) it can be stated that AP is a meta algorithm rather than an algorithm itself.

The main aim of this paper was to show how various boolean k-symmetry problems were solved in the past (using GP), and how they can be solved by means of evolutionary algorithms applied in AP. Analytic programming was used here in three basic comparative simulations. Each comparative simulation was 50 times repeated and all 450 results (50 simulations for each algorithm and for each problem) were used to create graphs and tables for AP performance evaluation.

For the comparative study two algorithms were used - DE (Price 1999) and SOMA (Zelinka 2004). Using a wide variety of optimisation algorithm, i.e. with different structure and their different ability to locate global extreme, were chosen to prove that AP can be regarded as an equivalent to GP, and that it can be implemented using arbitrary evolutionary algorithms. As a conclusion the following statements are presented:

1. **Reduction of cost function evaluation.** During the simulations described here, a significant low number of cost function evaluations, needed to reach the optimal solution, were observed. While for GP usually 600000 cost function evaluations were needed, as reported in (Koza 1998) for 5-symmetry, AP usually needed 40029 – 363432 evaluations (see Table 8 and 9).
2. **Reduction of population size.** In all simulations only 300 individuals were used. When comparing the population size (4000 and 16000), used in GP as mentioned in (Koza 1998), then AP uses population with 133 - 533 times less individuals. This is probably another reason for the low number of cost function evaluations (see previous point).
3. **Reached results.** Based on results reported in Tables 2 – 9 and Figures 3 - 11 it can be stated that all simulations give satisfactory results and thus AP is capable of solving this class of problems.
4. **Mutual comparison.** When comparing both algorithms, then it is visible that both algorithms give good results. Parameter setting for both algorithms were based on heuristically approach and thus there is a possibility that better settings can be found there.
5. **Universality.** AP was used to solve differential equations (Zelinka 2002 b), trigonometrically data fitting (Zelinka 2002 a), four polynomial problems from (Koza 1998) (Sextic, Quintic,

Sinus Three, Sinus Four) by four EAs in (Zelinka and Oplatkova, 2003) and Boolean even-k-parity functions synthesis (Zelinka and Oplatkova, 2004). Together with the results for Boolean k-symmetry functions reported here it can be stated that AP is a universal method for symbolic regression by means of arbitrary EAs.

Future research is one of the key activities in the frame of AP. According to all results obtained during time it is planned that the main activities would be focused expanding of this comparative study for genetic algorithms and simulated annealing.

ACKNOWLEDGEMENT

This work was supported by grant No. MSM 26500014 of the Ministry of Education of the Czech Republic and by grants of the Grant Agency of the Czech Republic GACR 102/03/0070 and GACR 102/02/0204.

REFERENCES

- Johnson Colin G., Artificial immune systems programming for symbolic regression, In C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, editors, Genetic Programming: 6th European Conference, LNCS 2610, p. 345-353
- Koza J. R., M. A. Keane, M. J. Streeter, 2003, Evolving Inventions, Scientific American, February 2003, p. 40-47, ISSN 0036-8733
- Koza J.R. 1998, Genetic Programming II, MIT Press, ISBN 0-262-11189-6, 1998
- Koza J.R., Bennet F.H., Andre D., Keane M., 1999, Genetic Programming III, Morgan Kaufmann pub., ISBN 1-55860-543-6, 1999
- O'Neill M. and Ryan C. 2002, Grammatical Evolution. Evolutionary Automatic Programming in an Arbitrary Language. Kluwer Academic Publishers, ISBN 1402074441
- O'Sullivan John, Conor Ryan, 2002, An Investigation into the Use of Different Search Strategies with Grammatical Evolution Proceedings of the 5th European Conference on Genetic Programming, p.268 - 277, 2002, Springer-Verlag London, UK, ISBN:3-540-43378-3
- Price K. 1999, An Introduction to Differential Evolution, in New Ideas in Optimization, D. Corne, M. Dorigo and F. Glover, Eds., s. 79-108, McGraw-Hill, London, UK, 1999. ISBN 007-709506-5
- Ryan C., Collins J.J., O'Neill 1998, M. Grammatical Evolution: Evolving Programs for an Arbitrary Language. Lecture Notes in Computer Science 1391. First European Workshop on Genetic Programming 1998
- Zelinka I., 2002 a, Analytic programming by Means of Soma Algorithm. Mendel '02, In: Proc. 8th International

Conference on Soft Computing Mendel'02, Brno, Czech Republic, 2002, 93-101., ISBN 80-214-2135-5

Zelinka I., 2002 b, Analytic programming by Means of Soma Algorithm. ICICIS'02, First International Conference on Intelligent Computing and Information Systems, Egypt, Cairo, 2002, ISBN 977-237-172-3

Zelinka I., Oplatkova Z., 2003, Analytic programming – Comparative Study. CIRAS'03, The second International Conference on Computational Intelligence, Robotics, and Autonomous Systems, Singapore, 2003, ISSN 0219-6131

Zelinka I., Oplatkova Z., 2004, Boolean Parity Function Synthesis by Means of Arbitrary Evolutionary Algorithms - Comparative Study", In: 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004), Orlando, USA, in July 18-21, 2004, in print

Zelinka Ivan, 2004, SOMA – Self Organizing Migrating Algorithm“, Chapter 7, 33 p. in: B.V. Babu, G. Onwubolu (eds), New Optimization Techniques in Engineering, Springer-Verlag, ISBN 3-540-20167X

AUTHOR BIOGRAPHIES



IVAN ZELINKA was born in Czech Republic, and went to the Technical University of Brno, where he studied technical cybernetics and obtained his degree in 1995. He obtained Ph.D. degree in technical cybernetics in 2001 at Tomas Bata University in Zlin. Now he is senior lecturer (artificial intelligence, theory of information) and head of department. His e-mail address is: zelinka@ft.utb.cz and his Web-page can be found at <http://www.ft.utb.cz/people/zelinka>.



ZUZANA OPLATKOVA was born in Czech Republic, and went to the Tomas Bata University in Zlin, where she studied technical cybernetics and obtained her degree in 2003. She is now Ph.D. student. Her e-mail address is: oplatkova@ft.utb.cz and her Web-page can be found at www.zuzkaoplatkova.webz.cz



LARS NOLLE graduated from the University of Applied Science and Arts in Hanover in 1995 with a degree in Computer Science and Electronics. After receiving his PhD in Applied Computational Intelligence from The Open University, he worked as a System Engineer for EDS. He returned to The Open University as a Research Fellow in 2000. He joined The Nottingham Trent University as a Senior Lecturer in Computing in February 2002. His research interests include: applied computational intelligence, distributed systems, expert systems, optimisation and control of technical processes.

ON THE EFFECT OF STEP WIDTH SELECTION SCHEMES ON THE PERFORMANCE OF STOCHASTIC LOCAL SEARCH STRATEGIES

Lars Nolle
 School of Computing and Mathematics
 The Nottingham Trent University
 Burton Street, Nottingham, NG1 4BU, UK
 lars.nolle@ntu.ac.uk

KEYWORDS

Optimisation, Hill Climbing, Step Width Selection.

ABSTRACT

This paper examines and discusses the effects of different step width selection schemes on the effectiveness of stochastic local search strategies. It is shown that the success of these search strategies depends, among other things, heavily on the chosen neighbourhood definition for a particular application. Simulations have shown that the use of randomly selected steps with a defined upper limit increases the probability of finding the global optimum, in contrast of using a constant step width. It has also been demonstrated that the efficiency of stochastic local search strategies can be improved by reducing the maximum step width over time.

INTRODUCTION

Stochastic local search algorithms, like Hill-Climbing (Hopgood 2001) or Simulated Annealing (Metropolis et. al. 1953) (Kirkpatrick et. al. 1983), are a class of search strategies that start from a random point in the search space and that test new potential candidate solutions that are randomly selected from the 'neighbourhood' of the current solution. If the candidate solution has a higher fitness than the current solution it replaces the current solution.

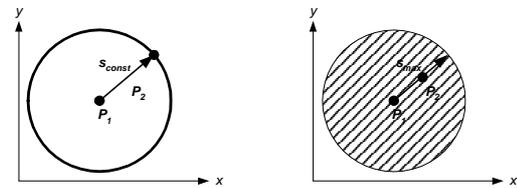
For continuous parameter optimisation, it is practically impossible to choose direct neighbours, because of the vast number of points in the search space. In this case, it is necessary to choose new candidate solutions from a wider neighbourhood, i.e. from some distance of the current solution, in order to travel in an acceptable time through the search space.

In principle, there are two ways to traverse through the search space, firstly using equidistant steps and secondly, using steps of random length.

Equidistant vs. Random Length Steps

The distance between the current solution P_1 and a candidate solution P_2 could either be a fixed step width s_{const} or it could have an upper limit s_{max} . In the first case, the neighborhood would be defined as the surface of a hypersphere around the current solution P_1

(Figure 1a), in the second case the neighborhood would be defined as the volume of the hypersphere and hence a candidate solution would be drawn randomly from within these hypersphere (Figure 1b).

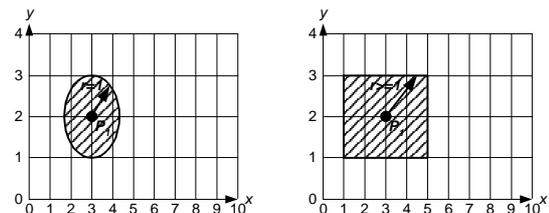


a) Constant Step Width b) Maximum Step Width

Figure 1: Possible Neighbourhood Definitions

Vector vs. Component Selection

In case of a constant step width, the displacement vector between P_1 and P_2 needs to be calculated for each step. One problem associated with this calculation is, that for dimensions of the input space with unequal ranges, the neighbourhood becomes distorted (Figure 2a). As a consequence to that, the input space will be sampled with a higher resolution in one dimension compared with the other one. To overcome this problem, one could determine the displacement for each input dimension separately using a scaling factor to compensate for the different resolution (Figure 2b).



a) Distortion b) Compensated

Figure 2: Effect of Distortion Due To Unequal Resolutions of Input Space Dimensions and the Compensation by Component Selection

Aim of Research

The aim of this work was not to compare different search algorithms in order to find the most efficient one to solve a set of test functions. Instead, the aim was to examine the effects of different step width selection schemes on the effectiveness of stochastic local search strategies. Therefore, the basic hill-climbing algorithm was used to find the minimum of two different test

functions, a high dimensional one and a multi-modal one.

TEST FUNCTIONS USED

For the simulations, two well established test functions were used, DeJong's first test function (DeJong 1975) and Schwefel's function (Schwefel 1981). Figure 3 shows a graphical representation of a two dimensional version of DeJong's 1st function, whereas Figure 4 depicts a two dimensional version of Schwefel's function.

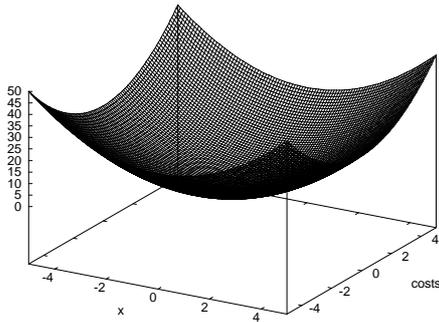


Figure 3 – DeJong's 1st Function (2D)

DeJong's 1st function, also known as sphere model, has a minimum cost value of zero within the defined range [-5,+5] for each input dimension. For the experiments a 10 dimensional version was used.

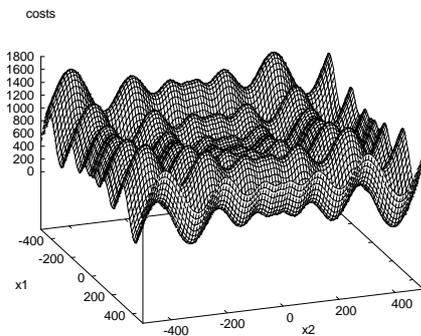


Figure 4 –Schwefel's Function (2D)

Schwefel's function has also a minimum cost value of zero within the defined range [-500,+500] for each input dimension. In this research a two dimensional version was used.

EQUIDISTANT VS. RANDOM LENGTH STEPS

In this set of experiments, the effect of using a random step width instead of a fixed step length has been examined. For both test functions, the step width, i.e. either constant or random, was varied and 1000 simulations per step width were carried out.

DeJong's 1st Function

For the 10 dimensional version of DeJong's 1st function, 2000 iterations were allowed per run. Figure 5 shows the results of the simulations. The dots represent the average costs achieved during the experiments and the error bars represent the standard deviation achieved.

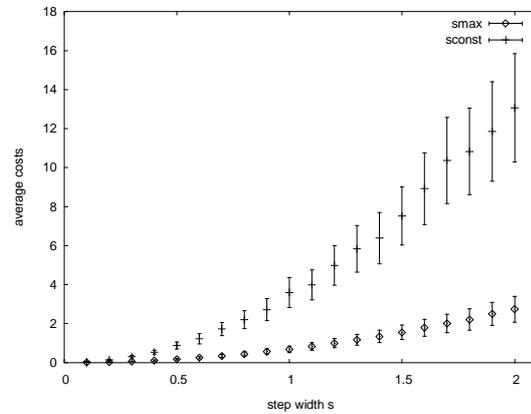


Figure 5: Average Costs vs. Step Width for DeJong's 1st Test Function

It can be seen that the average costs achieved using the s_{max} selection scheme were significantly lower compared to the s_{const} selection scheme. On the other hand, Figure 6 shows that it took the s_{max} selection scheme longer to converge.

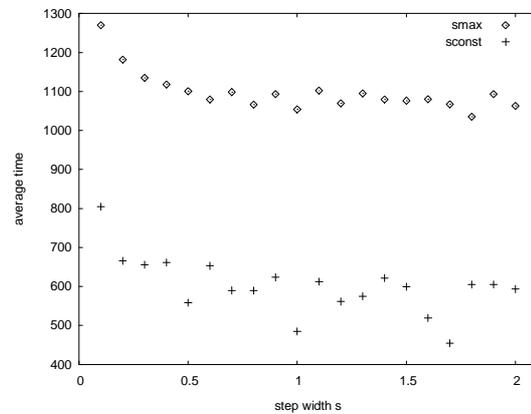


Figure 6: Average Time in Iterations for Finding the Best Solution During a Search for DeJong's Function

However, this is not a disadvantage, this only indicates that the s_{const} selection scheme simply failed to further improve the solutions found whereas the s_{max} selection scheme achieved a constant improvement even towards the end of the search runs.

Schwefel's Function

For the two dimensional version of Schwefel's function, 2000 iterations were allowed per run. Figure 7 shows the results of the simulations. The dots represent the average costs achieved during the experiments and the error bars represent the standard deviation achieved.

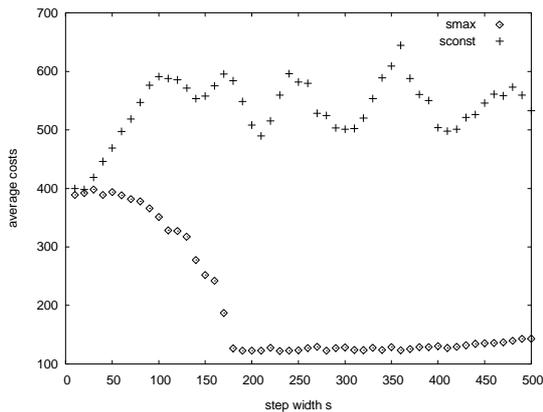


Figure 7: Average Costs vs. Step Width for Schwfel's Test Function

It can be seen from Figure 7 that small values for s_{max} result in relatively high costs, similar to the costs achieved using the s_{const} selection scheme. But these costs dropped dramatically when s_{max} reached 180 in this experiments, whereas the average costs achieved using s_{const} increased with increasing values for s_{const} .

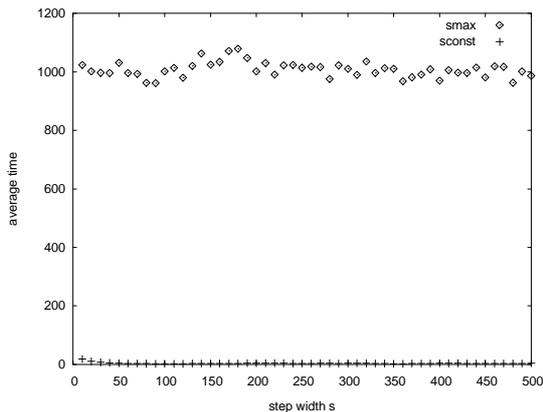


Figure 8: Average Time in Iterations for Finding the Best Solution During a Search for Schwfel's Function

Figure 8 shows the average time (in iterations) that was needed to reach the optimum for both selection schemes. Again, it can be seen that the s_{max} scheme found its best solutions after approximately 1000 iterations, in contrast to s_{const} , which converged significantly faster.

Discussion

Figure 9-Figure 13 show steps in typical search runs using the different selection schemes to minimise Schwfel's function. Each dot represents an accepted solution, i.e. the rejected solutions are not shown. The starting point for all experiments is on top of the local hill at about -200. Figure 9 and Figure 10 show the results for using a constant step width, Figure 11, Figure 12 and Figure 13 show the results for experiments using a maximum random step width.

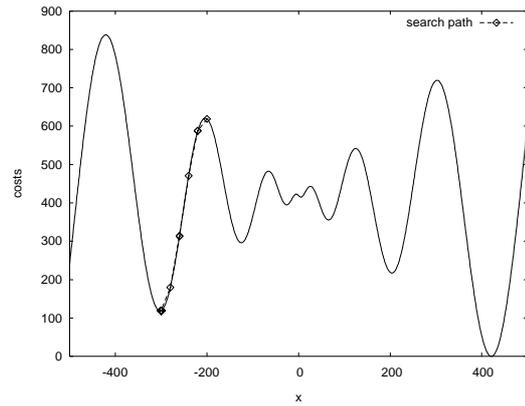


Figure 9: Constant Step Width $s_c = 20$

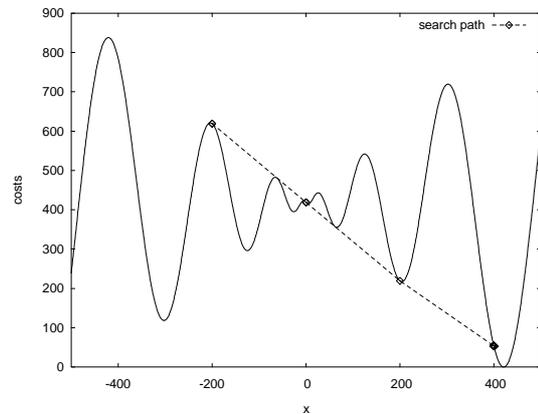


Figure 10: Constant Step Width $s_c = 200$

As it can be seen from Figure 9, if s_{const} is chosen to be too small, the algorithm gets trapped in an adjacent local optimum. If s_{const} is large enough, it is capable of jumping over local optima (Figure 10), but because the steps are of constant length, the number of points, which can be reached from a certain position, is limited. For example, in Figure 10, it is not possible to jump from the last point (near +400) to a point with lower associated costs, because all these points are within the constant step width.

When using a random step length s_{max} , the effect for choosing small values is similar to that for using a constant step width (Figure 11), but with increasing s_{max} , the ability to reach points with a better cost value increases. In Figure 12 for example, the algorithm was able to jump out of the valley on the right hand side of the start point into the valley of the left hand side, and it was also able to decent into the valley on the left hand side. However, because the maximum step width was not large enough to jump into the right hand side of the search space, which contains the global minimum, the algorithm only exploited the local optimum.

In Figure 13, the maximum step length is large enough to jump into the right hand side of the search space, but this time, because of the large steps that are possible, it jumps back into a local optimum.

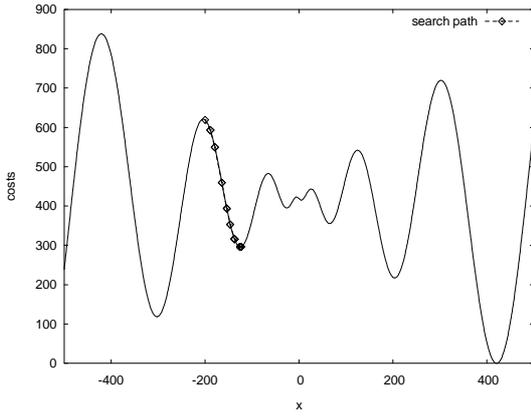


Figure 11: Maximum Step Width $s_{max} = 20$

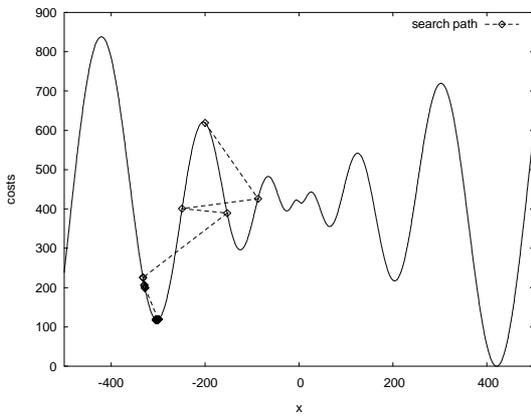


Figure 12: Maximum Step Width $s_{max} = 200$

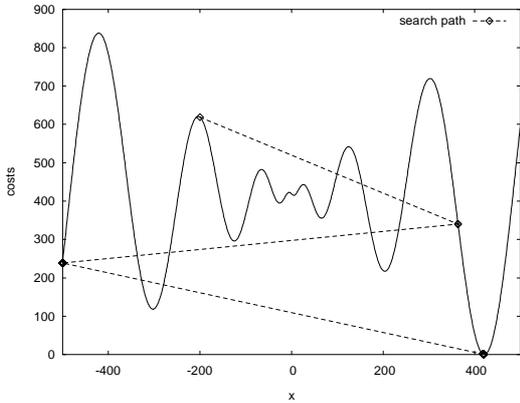


Figure 13: Maximum Step Width $s_{max} = 1000$

Conclusion

The results given above show, that, if the step width is chosen to be too small, both selection schemes will eventually get trapped in the nearest local optimum, whereas using a larger step width increases the ability to overcome local optima. However, for a constant step width it is more difficult to reach the global optimum, because of the limited number of possible search points that can be reached from a certain position.

This is overcome by using a maximum random step length, but this needs to be carefully selected: if it is chosen to be too large, the algorithm will lose its ability to exploit a potential region, which can also be seen in Figure 5, which documents the results for DeJong's 1st function, which only has one optimum.

But if the step width is chosen to be too small, it is not guaranteed that the algorithm can jump over local optima in order to reach the region that contains the global optimum. Therefore, the careful selection of appropriate values for s_{max} seems to be crucial for the success of the algorithm for a particular application.

REDUCING S_{MAX} OVER TIME

The findings above indicated that a large value for s_{max} supports exploration of the search space, whereas small numbers for s_{max} increase the exploitation capabilities of the search algorithm. It would be advantageous if an algorithm would explore the search space at the beginning of a search and exploit the most promising region in the later stages of the search.

Therefore, in the next set of simulations, three different scaling functions have been used to subsequently reduce s_{max} during the run, starting with a relatively large start value. Equation (1) achieves a linear reduction, equations (2) and (3) reduce s_{max} exponentially. Figure 14 shows a graphical representation of the three functions.

$$s_{max}(i) = LIN(i) = \frac{-s_0}{i_{max}} i + s_0 \quad (1)$$

$$s_{max}(i) = EXP1(i) = s_0 \cdot e^{\frac{-\alpha}{i_{max}} i} \quad (2)$$

$$s_{max}(i) = EXP2(i) = s_0 + s_0 \frac{1 - e^{\alpha i / i_{max}}}{1 - e^{-\alpha}} \quad (3)$$

Where:

- i : iteration
- i_{max} : maximum number of iterations
- s_0 : initial maximum step width
- $s_{max}(i)$: maximum step width at iteration I
- α : constant, determines the degree of non-linearity

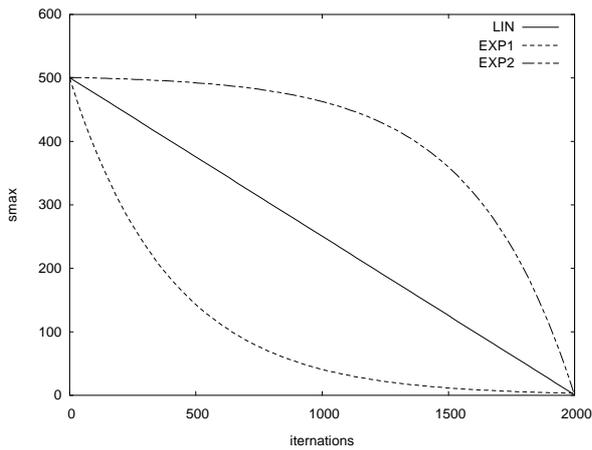


Figure 14: Scaling Functions Used

Each scaling function was used 1000 times and the results are compared with the best results obtained using a non-scaled maximum step width (see Figure 7).

Comparison of Results

Table 1 shows the result of the experiments for the non-scaling algorithm and the three algorithms using scaling functions, in Table 2 the performance of the algorithms are compared with the standard non-scaling one.

Table 1 – Results

	Constant	Linear	Exp1	Exp2
Average costs	132.90	124.29	121.99	125.22
Standard deviation	83.38	85.17	87.98	86.13
Average time	1105.9	1984.3	1800.3	1981.9
Standard deviation	578.0	69.9	211.8	117.1

Table 2 – Improvements by Algorithms

	Linear	Exp1	Exp2
Average costs	6.48%	8.21%	5.78%
Standard deviation	-2.15%	-5.52%	-3.30%
Average time	-79.43%	-62.79%	-79.21%
Standard deviation	87.91%	63.36%	79.74%

It can be seen that by using the scaling function EXP1 the average costs could be further reduced by up to 8.21%. The standard deviation of the average costs, on the other hand, increased by 5.52%. This, and the fact that the average time until the algorithm found the best solution increased by 62.79%, is probably an

indication that the algorithm converges slower because of the – now reduced – speed with that the algorithm is traversing over the fitness landscape.

CONCLUSIONS

In this work it was shown that the definition of the neighbourhood is crucial to the success of the algorithm. Two different selection schemes are compared in this work: equidistant steps and random steps with an upper limit. It has been demonstrated that random steps with an upper limit outperform neighbourhood selection schemes with a constant step length.

It was also shown that using a scaling function to reduce the maximum step with over time could again increase the performance of the algorithm.

AUTHOR BIOGRAPHY



Lars Nolle graduated from the University of Applied Science and Arts in Hanover in 1995 with a degree in Computer Science and Electronics. After receiving his PhD in Applied Computational Intelligence from The Open University, he worked as a System Engineer for EDS. He returned

to The Open University as a Research Fellow in 2000. He joined The Nottingham Trent University as a Senior Lecturer in Computing in February 2002. His research interests include: applied computational intelligence, distributed systems, expert systems, optimisation and control of technical processes

REFERENCES

- Hopgood, A.A. 2001. "Intelligent Systems for Engineers and Scientists". 2nd ed., CRC Press
- Kirkpatrick, S., Gelatt Jr, C. D., Vecchi, M. P. 1983 "Optimization by Simulated Annealing". *Science*, Vol. 220, No. 4598, 671-680
- Metropolis, A., Rosenbluth, W., Rosenbluth, M. N., Teller, H., Teller, E. 1953. "Equation of State Calculations by Fast Computing Machines". *Journal of Chemical Physics*, Vol. 21, No. 6, 1087-1092
- DeJong, K.A. 1975. "An analysis of the behavior of a class of genetic adaptive systems". PhD Thesis, University of Michigan
- Schwefel, H.-P. 1981. "Numerical optimization of computer models". Wiley and Sons

Performance of Hybrid Genetic Algorithms Incorporating Local Search

T. Elmihoub, A. A. Hopgood, L. Nolle and A. Battersby
*The Nottingham Trent University, School of Computing and Technology,
Burton Street, Nottingham NG1 4BU, United Kingdom
{tarek.elmihoub, adrian.hopgood, lars.nolle, alan.battersby}@ntu.ac.uk*

ABSTRACT

This paper investigates the effects of learning strategy and probability of local search on the performance of hybrid genetic algorithms. It compares the performance of two genetic–local hybrids using different learning strategies and different probabilities of local search. Two test functions are used for the comparisons. The results show that the solution quality of hybrids is not only affected by the Lamarckian or Baldwinian learning strategy, but also by the probability of local search. This probability, together with the learning strategy, has a great impact on population size requirements. These requirements are also affected by the local search method, and the fitness landscape. Reducing the population size can lead to an increase in the algorithm convergence speed.

INTRODUCTION

The ability of genetic algorithms to capture a global view of the search space, when combined carefully with the fast convergence of local search methods (Turney 1996), can often produce an algorithm that outperforms either one alone (Bobo and Goldberg 1997). Hybridizing a local search method provides the global genetic search algorithm with some local knowledge that can guide and may accelerate the search to the global optimum (Hart 1994).

The usual motivation for hybridization in optimization practice is the achievement of increased efficiency (Goldberg and Vossler 1999). The efficiency of any hybrid depends on many factors, e.g. how the hybrid decides between global and local knowledge (Bobo and Goldberg 1997), how it strikes a balance between the cost and value of local knowledge (Hart 1994), and how successfully local knowledge are utilized by the global genetic algorithm (Whitley et al. 1994). The efficiency of any hybrid can be measured by comparing its performance with that of the global genetic algorithm alone. Espinoza et al. (2001) have proposed an adaptive hybrid algorithm that can increase convergence speed to the global optimum. The same authors also show the effect of a local search method on reducing the population size of the algorithm compared with the population size of the standard genetic algorithm (Espinoza et al. 2003).

In this paper, a further step is taken in this direction by investigating the effect of the learning strategy and probability of local search on the performance on both the adaptive hybrid algorithm and the standard staged hybrid. The effect of both these factors on the population size requirements, convergence speed and solution quality has been studied.

LAMARCKIAN EVOLUTION AND BALDWIN EFFECT

One of the important issues of hybrid genetic algorithms is how the information gained during local search is used by the global algorithm. Either the Lamarckian or the Baldwinian approach can be used. In the Lamarckian approach the traits acquired during the learning process are passed from parents to their offspring. This means that both the genetic structure of an individual and its associated fitness value are modified to reflect the changes in phenotype structure as a result of performing local search (Whitley et al. 1994). The Baldwin Effect is somewhat Lamarckian in its results but using different mechanisms (Turney 1996). In the Baldwinian approach the learning process can help the individual to adapt to its environment and as a result to survive and gain more chance to pass on its traits to the next generation. In this case, only the improved fitness value is modified to reflect the effect of performing local search, thereby allowing individuals with the ability to learn to proliferate in the population.

Although Lamarckian evolution has been universally rejected as a viable theory of genetic evolution in nature, using it as learning strategy in genetic algorithms can improve their convergence speed (Whitley et al. 1994). The Lamarckian strategy can disrupt schema processing of genetic algorithms in staged hybrid algorithms and in some cases this may lead to the premature convergence problem (Whitley et al. 1994). In many real-world applications, it is not possible to use the Lamarckian approach because the inverse mapping from phenotype to genotype is computationally intractable (Turney 1996). The Baldwinian approach, in spite of being characterized by slow convergence speed compared with that of Lamarckian (Whitley et al. 1994), has a smoothing effect on the search landscape and does not disrupt the global genetic search (Gruau and Whitley 1993).

Utilizing either form of learning is more effective than the standard genetic algorithm approach without a local

improvement procedure (Whitley et al. 1994). The effectiveness of pure Lamarckian, pure Baldwinian or any mixture of them is affected by the fitness landscape, the representations, and local search method used (Whitley et al. 1994; Houck et al. 1997; Michalewicz and Nazhiyath 1995).

BALANCE BETWEEN COST AND VALUE OF LOCAL KNOWLEDGE

In any hybrid algorithm, a local search can be applied either to every individual in the population or only few individuals. Applying a local search to every individual in the population can waste resources without providing any more useful information than applying it to only a small fraction of the population. The use of a large fraction of the population can limit exploring the search space by allowing the genetic algorithm to evolve for a small number of generations. A more selective use of local search can improve the efficiency of hybrids (Hart 1994). Deciding on the optimal fraction of the population that should perform local search, and the basis on which these individuals are chosen, has been investigated by Hart (1994). The cost of local knowledge is measured by the number of function evaluations performed by a local search method to gain that knowledge and its value is measured by its effect on increasing the convergence speed and/or solution quality of the algorithm. The probability of local search can affect the minimum population size of the hybrid which, in turn, can affect the convergence speed of the algorithm. This effect should not be ignored when deciding between different local search probabilities.

POPULATION SIZE REQUIREMENTS

Efficient Population sizing is critical in genetic algorithms for getting the most out of a fixed budget of function evaluations. In (Harik et al. 1997) two factors that influence convergence quality are considered to estimate the population size of genetic algorithms. These factors are the initial supply of building blocks and the selection of the best building blocks over their competitors. The gambler ruin model is used to derive the following relation for population size of genetic algorithms

$$N = \frac{-2^{k-1} \ln(\alpha) \sigma_{bb} \sqrt{\pi(m-1)}}{d}$$

where k is the building-block order, which represents the minimum number of binary digits that have physical significance to the solution of the problem. α is the probability of failure, σ_{bb} is the standard deviation of the building blocks, d is the signal difference between the best and second-best building blocks, and m is the maximum number of building-blocks within a single string. The term

$\sqrt{\pi(m-1)}$ represents the noise interference between competing building-blocks. This term can be approximated using the fitness function standard deviation, $\sigma_{fitness}$ (Reed et al 2000).

The computational complexity of a genetic algorithm is measured as the number of function evaluations that are required to attain an optimal solution. The number of function evaluations can be calculated by multiplying the population size (N) by the number of generations required for convergence (t). The number of generation required is strongly affected by the relative rates at which genes within the population converge. The lower and upper bounds for the convergence rates for genetic algorithms applications are functions of $O(\sqrt{l})$ and $O(l)$ for tournament selections, where l is the string length (Thierens et al. 1998). The building blocks of most engineering problems converge at variable rates within the population (Reed 2000). This phenomenon is known as domino convergence. The expected number of generations (t) required under domino convergence for all locations to be converged is given by the following equation.

$$t_{domino} \approx 2l$$

Another phenomenon that is closely related to domino convergence is “genetic drift” (Thierens et al. 1998). This phenomenon occurs in a population when crossover and mutation cause genes to fluctuate and converge to non-optimal values in the absence of selection pressure. Although the genes with reduced relevance to the solution experience reduced selection pressure, they may converge to non-optimal values under the crossover and mutation operations. The expected number of generations for genes to converge in the absence of selection within a randomly generated initial population is given by the following equation (Thierens et al. 1998):

$$t_{drift} \approx 1.4N$$

Domino convergence to optimal solution should occur before genetic drift can occur. The following inequality needs to be satisfied:

$$t_{domino} < t_{drift}$$

or, in terms of population size and string length,

$$N > 1.43l$$

Since carefully designed hybrid genetic algorithms often converge faster than standard genetic algorithms, their convergence to the global optimum can occur even if population size is not greater than $1.43l$. The population size for hybrid algorithm should satisfy the following relation

$$2l \frac{t_{do\ min\ o}}{t_{drift}} < 1.43N$$

where t_{hybrid} is the number of generations required to for a hybrid to converge.

The local search method affects the signal difference between the best individual and the second best, and this can either increase or decrease the population size. It can also decrease the standard deviation of the population and this leads to a decrease in the population size.

ALGORITHMS AND TEST FUNCTIONS

Two hybrids with different mechanisms for deciding between global and local search were used to gain some insight into the effect of learning strategy and probability of local search on the performance of hybrids. The standard staged hybrid genetic algorithm (SSH) (Mathias and Whitley 1994) and the adaptive staged hybrid genetic algorithm (ASH) (Espinoza et al. 2001) have been tested using two multimodal test functions.

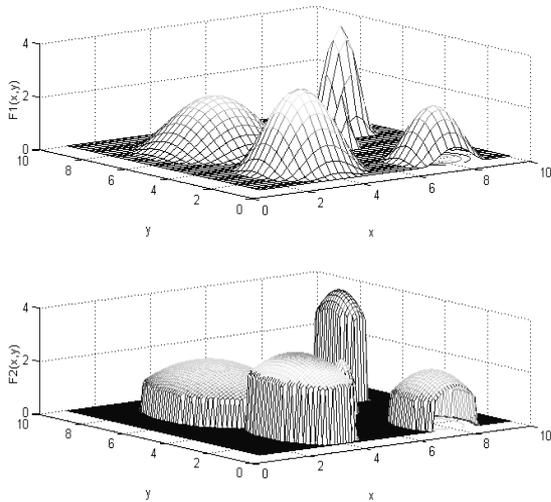


Figure 1: Fitness Landscapes for the Test Functions

In the standard staged hybrid genetic algorithm (SSH), the local search step is defined by three basic parameters: frequency of local search, probability of local search and number of local iterations. The local search frequency measures how frequently local search is performed; the probability of the local search represents the fraction of individuals in the population that undergo local search at each local search iteration; and the number of local search iterations represents the number of local search iterations performed at each local search process.

The adaptive staged hybrid genetic algorithm (ASH), uses feedback from the current state of the search process

to direct the algorithm to decide between global and local methods (Espinoza et al. 2001). The algorithm works with the same operators as SSH. It performs local search only if new regions of search space are being discovered, and local knowledge can help to guide the search. The probability of the local search is controlled by a deterministic rule that keeps this probability less than a specific value. When local search no longer improves the average fitness more than the most recent global search iteration, the search goes back to the global search.

Two multimodal test functions, with multiple basins of attraction, have been used in the current work. The first function, F1, has conical basins of attraction. Its global maximum is 4 and is located at (7.0,8.5) (Goldberg and Vossler 1999; Espinoza et al. 2001). The second function, F2, has elliptical basins of attraction. This function has a global optimum of 4 located at (7.0, 8.5) (Espinoza et al. 2001). Figure 1 shows the fitness landscapes of F1 and F2.

The steepest descent method (Press et al.1993) was used as a local searcher. The steepest descent algorithm uses the derivatives of the function to estimate the best step size to climb to the local optimum from the current position in the basin of attraction.

SIMULATIONS AND DISCUSSION

In order to evaluate the effect of learning strategy and local search probability on the hybrids' performance, a set of experiments was performed. Both hybrids use the simple elitist genetic algorithm with binary tournament selection, single-point crossover, and simple mutation. For all experiments, the probability of local search was 0.4 and the probability of mutation was $1/N$ where N is the population size (Reed et al. 2000). For SSH, the frequency of local search was 3 and the number of local iterations was 3. For ASH, the maximum number of local iterations was 3, e was 0.2, and the local threshold value was 0.6. Each variable was represented by 30-bit string with a total of 60 bits for each chromosome. The stopping criterion for all experiments was that 80% of the population had converged to the solution.

Effects on Convergence Speed

In the experiment to evaluate the effect of learning strategy on convergence speed of hybrid algorithms, both the adaptive and standard staged algorithms used a probability of local search of 0.1, and population sizes of 800 and 1200 for F1 and F2 (Espinoza 2001). The stopping criterion was that 80% of the population converged within 0.000001 boundaries of the best ever found solution.

The results show, as expected, that increasing the fraction of the population that evolves according to the Lamarckian approach leads to an increase in the convergence speed. This increase is not linear. For

example, when applying ASH on F2, the speed of convergence increases sharply as the learning approach changes from pure Baldwinian (100% Baldwinian) to a mixture of 80% Baldwinian and 20% Lamarckian. In this interval the number of function evaluations decreases from 85,000 to about 37,000, while it decreases to 25,000 evaluations for the pure Lamarckian approach. Figure 2 shows the effect of learning strategy on the convergence speed of the adaptive staged hybrid. The effect of learning strategy on the convergence speed of standard staged hybrid and the adaptive staged hybrid are similar for both test functions.

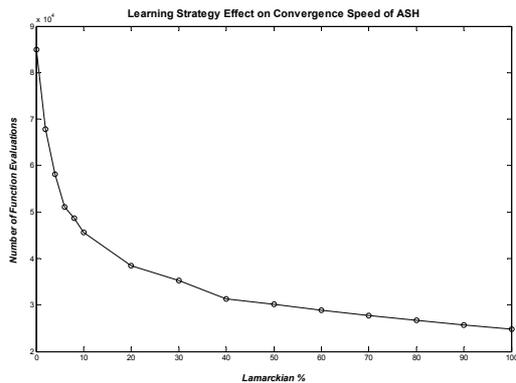


Figure 2: Effect of Learning Strategy on Convergence Speed

Effects on Solution Quality

The results of previous experiments show no clear relation between learning strategy and solution quality. This led us to consider how the local search probability interacts with the learning strategy and how this interaction affects the quality of solutions. An experiment was carried out to consider the effect of local probability on the solution quality for different population sizes (100, 400, 800, and 1200). The results of these experiments show that as probability of local search increases, the effect of learning strategy becomes apparent (figure 3). The graphs in figure 4 show that, when the probability of the local search is kept small, the quality of the solution is insignificantly affected by the learning strategy. As this probability increases, the quality of the solutions degrades with an increasing Lamarckian percentage in the learning process. This means using small local search probabilities for both algorithms, even with pure Lamarckian, can produce high quality solutions because the disruption to schema processing caused by these small probabilities is neglected and has no effect on global search process.

The results in Figure 4 show that a mixture of 20% Lamarckian and 80% Baldwinian produces the most stable solution quality for F2, regardless of the probability of the local search. A mixture of 75% Baldwinian and

25% Lamarckian produces the most stable solution quality for F1 (Figure 5). The results from both hybrid algorithms show that a pure Baldwinian approach does not always produce the optimal solution quality and that the optimal learning strategy depends on the probability of local search.

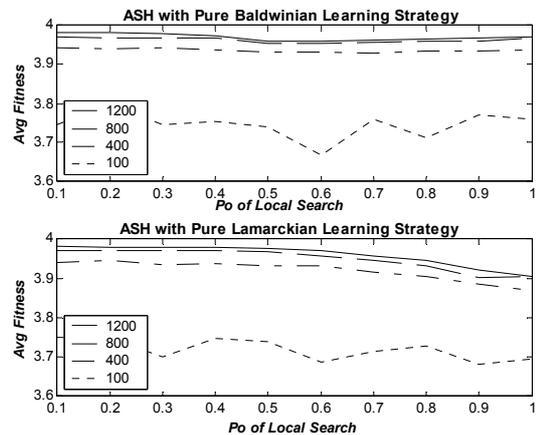


Figure 3: Effect of Learning Strategy and Search Probability on Solution Quality

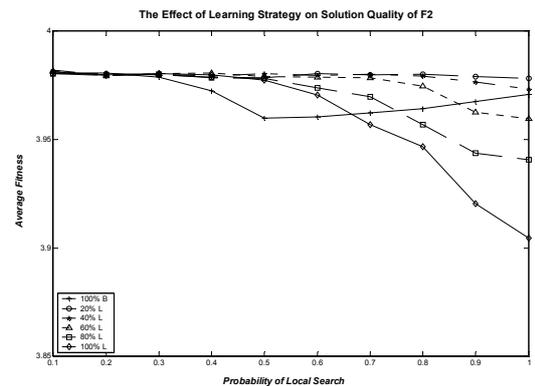


Figure 4: Solution Qualities for F2

Effect on Population Size

The aim of this experiment was to show how the probability of local search and learning strategy affect the minimum size requirements for both hybrids. The results were obtained by using bisection method. Starting with a population size of 10, the population size is doubled until the population converges to the desired solution quality. After the solution quality is attained, the population size is set midway between the current size and the last unsuccessful population size. This process is repeated until the difference between population sizes is less than or equal to 10. The stopping criterion was that 80% of the population converged within 0.000001 boundaries of the

global optimum. The settings of other parameters were as in the previous experiments.

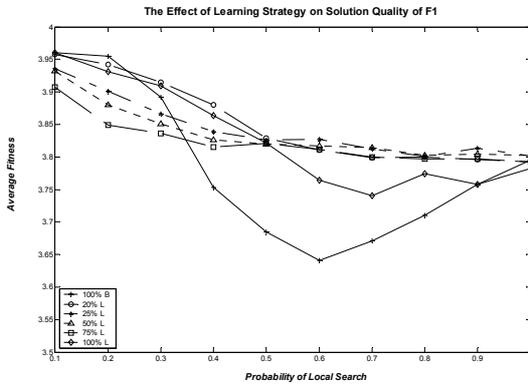


Figure 5: Solution Qualities for F1

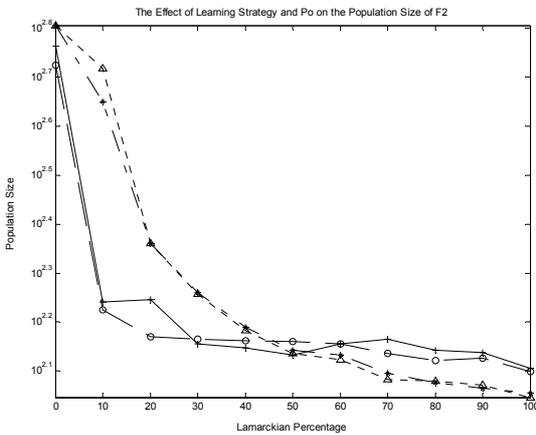


Figure 6: Effect of Learning Strategy and Search Probability on Population Size

The results of SSH and ASH on the second test function are similar. Figure 6 shows that, as the probability of local search increases, the population size decreases for a pure Lamarckian approach. On the other hand, with a pure Baldwinian strategy, the population size increases as the probability of local search increases. For a pure Baldwinian strategy with local search probability of more than 0.4, the population size exceeds that of a pure genetic algorithm (minimum population size=640). The results also show that the relationship between the local search probability and the change in the population size depends on the learning strategy used. For example, using a partial Lamarckian of 50% or more, an increase in the local search probability results in a decrease in population size. With a partial Lamarckian of less than 50%, an increase in the local search probability leads to an increase in the population size. For both hybrids, a decrease in population size leads to an increase in the convergence

speed. In general, increasing the Lamarckian percentage decreases the population size and increases the convergence speed. The experiments also show that the solution quality of the pure Baldwinian approach is the optimal and the solution quality is degraded as both the Lamarckian percentage and the probability of local search increase. The solution quality for impure Baldwinian strategies, as shown in Figure 7, seems to be more dependent on the probability of local search than on the learning strategy.

The local search can decrease both the standard deviation of the population and the signal difference between the best and second-best solutions, since the population size depends directly on the standard deviation of the population and the signal difference. A decrease in the former decreases the population size and a decrease in the latter increases the population size.

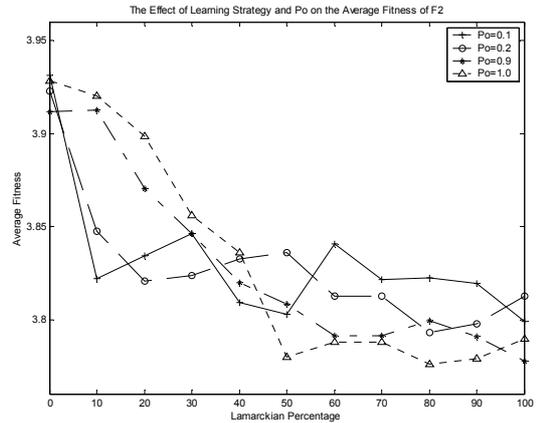


Figure 7: Effect of Lamarckian Proportion and Search Probability on Solution Quality

The increase in the population size requirements for the pure Baldwinian approach can be explained as follows. In a pure Baldwinian, the local search needs some help from evolution process to keep decreasing the ratio of standard deviation to signal difference. Pure Baldwinian can reduce this ratio at the end of the local search. However, in the next global iteration, if the value of local knowledge is insufficient to keep the global genetic algorithm reducing this ratio, the algorithm will lose some of its resources (i.e. local function evaluations) without reducing that ratio. In this case, a high probability of local search cannot lead to any reduction in the population size since it increases the probability of losing the algorithm's resources. However, a low local search probability reduces the probability of lost resources while increasing the probability of maintaining the reduction in

the above-mentioned ratio by the global genetic algorithm. In addition to the probability of local search, the effectiveness of pure Baldwinian in reducing the population size depends on the value of local knowledge and this depends on the method of local search and fitness landscape.

On the other hand, the opportunity to keep the gained reduction in this ratio is improved by using a partial Lamarckian strategy. As the percentage of Lamarckian increases, the probability of keeping this reduction increases. An increase in the probability of local search increases the probability of reducing the ratio and reducing the population size.

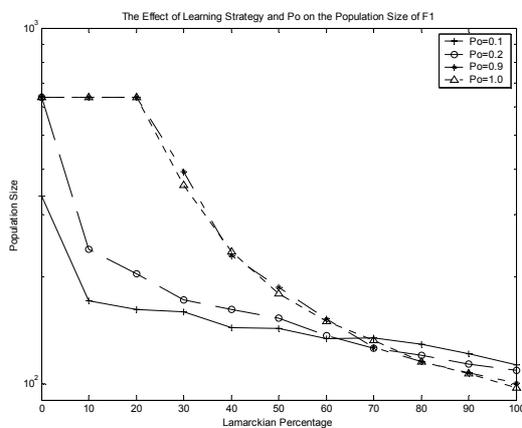


Figure 8: Effect of Learning Strategy and Search Probability on Population Size of F1

Figure 8 shows the results of running the same experiment on the first test function. For a Lamarckian percentage of 65% or more, an increase in the probability of local search results in a reduction in the population size. For other percentages, an increase in this probability leads to an increase in the population size requirements. The convergence speed depends on the population size; as the population size decreases the convergence speed increases. Comparison of Figures 6 and 8 shows that the switch point on the Lamarckian axis between increasing and reducing the population size is shifted from about 50% for F2 to about 65% for F1. This is due to the differences in the fitness landscape of both functions. While the local search can provide more significant local knowledge in F1 than in F2, an impure Lamarckian approach requires a more partial Lamarckian to accelerate the genetic assimilation process.

Additionally, the effect of the local search method on F1 is to enable any solution in a basin of attraction to climb to the exact local optimum in a single step. Consequently, increasing the probability of local search does not necessitate decreasing the signal difference

between the best and second-best solutions. It also makes the selection process more difficult as the search process progresses when using a pure Baldwinian approach. In contrast, in F2 the local search method sends any point in the basin of attraction to a point near local optima and not to the local optimum itself.

The local search method can provide more significant local knowledge from the landscape of F1 than F2. This is why the reduction in the population size requirements of F1, using a pure Lamarckian approach, is greater than that of F2. This also makes the genetic assimilation process more difficult for F1 using a pure Baldwin effect compared with F2. The use of a partial Lamarckian can accelerate the genetic assimilation process. The exact value of the switch point depends on the value of the local knowledge.

CONCLUSIONS AND FUTURE WORK

The simulations show that using a low probability of local search and using a pure Lamarckian learning strategy can improve the convergence speed of the algorithm without disrupting the schema processing of the global genetic algorithms. They also show that, depending on the learning strategy used, increasing the probability of local search can decrease or increase the population size. As a result, the convergence speed is affected by the probability of local search. The results show that there is a relation between the probability of local search and the population size.

These experiments have attempted to provide an insight into how the probability of local search and learning strategy affect the population size requirements. We now plan to study how the population size can affect the optimal local search probability by developing a self-adaptive hybrid algorithm that encodes the number of local iterations within the chromosomes themselves and to study how their values propagate during the evolution process.

REFERENCES

- Bobo, F. and Goldberg, D. 1997. "Decision Making in a Hybrid Genetic Algorithm", In T. Back (Ed.) Proc. the 1997 IEEE International Conference on Evolutionary Computation, pp. 121-125. IEEE Press
- Espinoza, F., B; Minsker, B. and Goldberg, D. 2001. "A Self Adaptive Hybrid Genetic Algorithm". L. Spector, E. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors. Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'2001. San Francisco, Morgan Kaufmann Publishers.
- Espinoza, F.; Minsker, B. and Goldberg, D. 2003. "Performance Evaluation and Population Size reduction for Self Adaptive Hybrid Genetic Algorithm (SAHGA) ", Proc. the Genetic and

- Evolutionary Computation Conference, 2723, 922-933, Morgan Kaufmann
- Goldberg, D. and Vossen, S. 1999. "Optimizing Global-Local Search Hybrids" Proc. the Genetic and Evolutionary Computation Conference, pp 220-228, Morgan Kaufmann.
- Gruau, F. and Whitley, D. 1993. "Adding Learning to the Cellular Development of Neural network: Evolution and Baldwin Effect", Evolutionary Computation, 1(3), 213
- Harik, G.; Cantu-Paz, E; Glodberg, D., and Miller, B. 1997. "The Gambler's Ruin Problem, Genetic algorithms, and the Sizing of Populations", In T. Back (Ed.) Proc. the 1997 IEEE International Conference on Evolutionary Computation, pp. 121-125.
- Houck, C.; Joines, J.; Kay, M; and Wilson, J. 1997. "Empirical investigation of the benefits of partial Lamarckianism", Evolutionary Computation, v.5, n.1, 31- 60.
- Hart, W. E. 1994. "Adaptive Global Optimization with Local Search". Doctoral Dissertation, University of California, San Diego, CA
- Mathias, K.; Whitley, D. ; Stork, C. and Kusuma, T. 1994. "Staged Hybrid Genetic Search for Seismic Data Imaging", 1994, International Conference on Evolutionary Computation
- Michalewicz, Z. and Nazhiyath, G., 1995. "Genocop III: A Co-evolutionary Algorithm for Numerical Optimization Problems with Nonlinear Constraints", Proc. of the 2nd IEEE International Conference on Evolutionary Computation, Vol.2, 647-651
- Press, W.; Teukolsky, S.; Vetterling, W. and Flannery, B. 1993. "Numerical Recipes in C", Cambridge University Press
- Reed, P.; Minsker, B. and Glodberg D. 2000. "Designing a Competent Simple Genetic algorithm for Search and Optimization". Water Resource Research, 36(12), 3757-3761
- Thierens, D., Goldberg, D. and Guimaraes P. 1998. "A. Domino Convergence, Drift, and the Temporal-Saliency Structure of Problems. In The 1998 IEEE International Conference on Evolutionary Computation Proceedings, pp. 535-540, IEEE Press, New York, NY
- Turney, P. 1996 "Myths and Legends of the Baldwin Effect", Proc. The ICML (13th International Conference on Machine Learning), 135-142.
- Whitley, D., Gordon, V. and Mathias, K. 1994. "Lamarckian Evolution, the Baldwin Effect and Function Optimization", Parallel Problem Solving from Nature-PPSN III, 6-15.

BIOGRAPHIES



TAREK EL MIHOUB is a PhD student at the Nottingham Trent University. He obtained his MSc in Engineering Multimedia in 2002 from Nottingham Trent University. He worked as a teaching assistant in Al-Fatah University in Libya and as a manager of IT department in the Libyan Environment general authority. His current research is in the field of optimization, genetic algorithms, and artificial intelligence.



ADRIAN HOPGOOD is professor of computing and head of the School of

Computing and Technology at the Nottingham Trent University, UK. He is also a visiting professor at the Open University. His main research interests are in intelligent systems and their practical applications. He graduated with a BSc (Hons) in physics from the University of Bristol in 1981 and obtained a PhD from the University of Oxford in 1984. He is a member of the British Computer Society and a committee member for its specialist group on artificial intelligence.



LARS NOLLE graduated from the University of Applied Science and Arts in Hanover in 1995 with a degree in Computer Science and Electronics. After receiving his PhD in Applied Computational Intelligence from The Open University, he worked as a System Engineer for EDS. He returned to The Open University as a Research Fellow in 2000. He joined The Nottingham Trent University as a Senior Lecturer in Computing in February 2002. His research interests include: applied computational intelligence, distributed systems, expert systems, optimisation and control of technical processes.



ALAN BATTERSBY obtained an MSc in Computer Science from Hatfield Polytechnic in 1977. Prior to joining the Computing department at Nottingham Trent University, Alan was a Computing Development Officer for Bedfordshire Education Authority. His research interests include: Fuzzy Logic applied to Robotics, wavelets, compression and the Internet.

COMPLEX SYSTEMS

KNOWLEDGE CAPTURE TO SUPPORT INFORMATION FLOW MANAGEMENT IN COMPLEX SYSTEMS

Edward Szczerbicki
School of Engineering
The University of Newcastle
Newcastle NSW 2308, Australia
E-mail: Edward.Szczerbicki@newcastle.edu.au

KEYWORDS

Information management, Knowledge capture, Information flow modelling.

ABSTRACT

Systems become increasingly complex. Their decomposition into smaller units is the usual way to overcome the problem of complexity. This has historically led to the development of atomized structures consisting of a limited number of autonomous subsystems that decide about their own information input and output requirements, i.e. can be characterized by what is called an information closure. Autonomous subsystems can still be interrelated and embedded in larger systems, as autonomy and independence are not equivalent concepts. These ideas are gaining very strong interest in both academia and industry, and the atomized approach to information flow modelling and evaluation is an idea whose time has certainly come. This presentation discusses some modelling and evaluation issues, and challenges existing in the exciting area of knowledge capture for information flow management support for autonomous subsystems.

INTRODUCTION

Managing complex systems that function in changing and uncertain information-rich environments requires greater understanding and knowledge about the role of information in systems operation. To gain this understanding, an approach is needed that could be used to model and evaluate information flow in different situations. Such an approach is presented in this paper.

In fact, this paper goes well beyond the above in proposing an approach considering important practical issues of information flow, i.e. delays, incompleteness, imprecision and loss in value. The current practice of dealing with such issues are mostly when problems are detected and reactively. This situation may not be desirable and definitely be a major drawback for systems that more and more rely on the timeliness and quality of information for their operation. The proposed approach, in this respect, would greatly enhance the understanding of

the various factors that influence the quality of information to the benefit of better decisions in adequate time, which in fact is the core of the philosophy behind any information system development.

System decomposition into smaller units is the usual way to overcome the problem of complexity. This has historically led to the development of atomised structures consisting of a limited number of *autonomous subsystems/agents* that decide about their own information input and output requirements, i.e. can be characterised by what is called an *information closure*. Autonomous subsystems/agents can still be interrelated and embedded in larger systems, as autonomy and independence are not equivalent concepts. These ideas are recently gaining very strong interest in both academia and industry, and the atomised approach to systems modelling, design and development is an idea whose time has certainly come (Morimoto 2001, Tharumarajah 1999, Prakken 2000, O'Grady 19999). The issues discussed in this paper will focus on information flow for autonomous subsystems/agents.

In a real-world industrial context, autonomous subsystems/agents consist of groups of people and/or machines tied by the flow of information both within a given subsystem and between this subsystem and its external environment (Szczerbicki 1996a). We will briefly present a modelling approach that could be used to evaluate such an information flow. The suggested approach allows for the evaluation of an information flow to be performed for different types of external and internal environments of a given system. The approach also accommodates the question of uncertain and imprecise information flow modelling.

ANALYTICAL MODELLING APPROACH

An autonomous subsystem/agent is usually functioning in the external environment which determines the decision-making process. Its knowledge could be described by the following:

- (i) characteristic of the external environment (relationship between variables describing the environment and its dynamics),
- (ii) characteristic of the internal environment, i.e. the relationship between the actions of the members of an agent,
- (iii) the range of information about variables describing external environment.

The formal representation of the above knowledge is presented in this Section. For the knowledge extraction purposes, a general approach is needed that captures the whole of the behaviour of an agent. Such an approach, based on correlation between information and energy, is very briefly outlined next. Certain features implemented in previous research presented in (Szczerbicki 2000, 2002a, 2002b) are included for the sake of completeness.

Let A represent the set of possible actions which can be undertaken by the members of an agent, Z the set of corresponding consequences, and X random variables describing the actual state of the external environment. It can be assumed that:

$$z=f(a,x) \quad (1)$$

as the particular consequence (z) depends usually on an action (a) undertaken in the particular state of the environment (x). On the other hand, the decision about particular action depends on information that is available about the state of the environment. If β stands for the decision function, we have

$$a=\beta(d) \quad (2)$$

where d represents information.

For general description of the function $f(a,x)$ let us consider certain correlation between information, action, and energy. Its theory is relatively young, but it has already been pointed out that in certain situations energy can be replaced by information and vice versa (matsumoto 1999, Bogdan 2000). This replacement is of statistical character and according to it for certain amount of information, say C_1 , certain task can be performed using E_1 energy (Szczerbicki 1996a). Then, for a given C_1 there exists the best way (action A_{opt}) to fulfill the job, i.e. the action which uses E_1 energy. Actions different than A_{opt} result in more energy consumption. The above concept was presented in detail in (Szczerbicki 1996a, 2003) and used to arrive at the best decision functions β_i . It can be shown that for n-person agent we have (Szczerbicki 1996a, 2003):

$$\beta_i(d_i) + \sum_{j \neq i} \alpha_j \beta_j(d_j) \mid d_i = E(b_i \mid d_i) \quad (3)$$

where $i, j = 1, 2, \dots, n$.

Formalization of agent decision making process expressed by (3) is a tool necessary for modelling and evaluation of information flow in an autonomous system. Information flow connects agent members with the external environment described by random variables X. The connection is represented by information structure. This structure is modelled by matrix C in which $c_{ij}=1$ if the i th member has obtained (either by observation or communication) information about the j th variable X realization (if $c_{ij}=0$ he/she has not got it). The i th variable X realization can be observed only by the i th member of the agent. He/she can be informed about other realizations only when communication (information exchange) inside the agent is organized. The value of information structure defined above is given by the following (Szczerbicki 2000):

$$VC = \min E[f(a, X) \mid C_0] - \min E[f(a, X) \mid C], \quad (4)$$

where $\min E[f(a, X) \mid C_0]$ represents the utility of information structure C_0 in which $c_{ij}=0$ for each i and j . Using (3) the VC can be represented by:

$$VC = E[b^T \beta]. \quad (5)$$

With the modelling tools given by (3) and (5) one can easily extract knowledge about autonomous systems functioning in various decision situations. In Table 1 some samples of such a knowledge are specified for static environments [11]. This knowledge is easily codified and can be used in control, command, and management of autonomous systems. Similar rules can be easily captured for decision situations involving dynamic environments (please see [11] for details).

Table 1: Production rules describing agents functioning in static environment

RULE 11	
IF	an external environment of an autonomous agent is static,
AND	it is described by random variables,
THEN	the value of an information structure that represents the flow of information between the agent and its environment depends on interaction between agent members, correlation between random variables, and their variance.
RULE 12	
IF	an external environment of an autonomous agent is static,
AND	it is described by a random variable,

	THEN	the value of information about this variable realization is proportional to the value of its variance.		THEN	negative correlation in the external environment is preferred.
RULE 13	IF	an external environment of an autonomous agent is static,		RULE 19	IF
	AND	it is described by random variables,			AND
	THEN	full information has the value that is always greater or equal to the value of any other information structure.			AND
					THEN
RULE 14	IF	an external environment of an autonomous agent is static,		RULE 20	IF
	AND	there is no interaction in the internal environment,			AND
	THEN	it is enough to restrict the information flow only to observation; organizing an information exchange does not improve the value of a resulting information structure.			AND
					THEN
RULE 15	IF	an external environment of an autonomous agent is static,		RULE 21	IF
	AND	there is an interaction in the internal environment,			AND
	AND	the relationship between variables describing the external environment is of statistical character,			AND
	THEN	information structure should include observation and communication.			AND
					AND
RULE 16	IF	an external environment of an autonomous agent is static,			THEN
	AND	the relationship between variables describing the external environment is given by function dependence,			
	THEN	communication between agent members does not affect the value of information structure; information flow should be restricted to observation.		RULE 22	IF
					AND
RULE 17	IF	an external environment of an autonomous agent is static,			AND
	AND	interaction in the internal environment is of substitute character,			AND
	THEN	positive correlation in the external environment is preferred.			THEN
RULE 18	IF	an external environment of an autonomous agent is static,		RULE 23	IF
	AND	interaction in the internal environment is of complementary character,			AND
					THEN

correlation in the external environment.

SOFT MODELLING APPROACH

A formal quantitative model as presented in the previous Section can be helpful in creation of knowledge connected with an information flow evaluation in autonomous systems. Because of its complexity the model cannot be used for analysis and evaluation of an information flow in all possible decision situations. Qualitative modeling and reasoning, on the other hand, are areas of Artificial Intelligence (AI) that focus on reasoning about the behaviour of real life complex systems without relying on numbers. In the development of an information structure for a given system, Qualitative Reasoning (QR) tools can play a role similar to that of traditional analysis based on the mathematical model.

Next, some non-quantitative tools are discussed for addressing the problem of knowledge acquisition for an autonomous subsystem/agent in various decision situations.

Connectionist systems

Problem solving tasks, such as information structure development, may be considered pattern classification tasks. The system analyst learns mappings between input patterns, consisting of characteristics of system's external and internal environment, and output patterns, consisting of information structures to apply to these characteristics. Thus, neural networks (neural-based expert systems) offer a promising solution for automating the learning process of the analyst.

As we already know, systems analyst, while developing an information structure for a given system, transforms certain characteristics of a system into recommendations concerning the flow of information. These characteristics represent the input for the system and their full description (for both static and dynamic environments) includes 5 parameters: correlation in the external environment (r), dynamics (t), interaction in the internal environment (q), delay (d), and type of the process describing the external environment (w). Output consists of the following decisions (recommendations): (i) observation (or sensing) should be present, and (ii) exchange of information should be present. An input portion together with an output portion of the data represents a training pair. The training pairs were used to train a 5-10-2 neural network (Szczerbicki 1996b).

The target values for each output node were normalised in such a way that the maximum target

for each node received a value of 0.75 and the minimum target for each node received a value of 0.25. The training values for each input node were identically normalised. The learning rate and momentum term of 0.9 were used in the network. The network was trained using error back propagation procedure with a training tolerance of 5%. The network was considered trained if, for all training pairs and output nodes, $|(desired\ output - actual\ output)/(desired\ output)| < tolerance$.

After training, additional characteristics of a system were generated for use by the network. Five sets of characteristics were submitted to the network. In response, the network suggested five information flow recommendations. In all cases the recommendations agree with the IF ... AND ... THEN rules presented in Table 1.

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decisionmaking knowledge from the supplied data (Safavian and Landgrebe 1991). Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C_1, C_2, \dots, C_k is as follows (Quinlan 1990):

- Step 1. If all the objects in S belong to the same class, for example C_i , the decision tree for S consists of a leaf labelled with this class.
- Step 2. Otherwise, let T be some test with possible outcomes O_1, O_2, \dots, O_n . Each object in S has one outcome for T so the test partitions S into subsets S_1, S_2, \dots, S_n where each object in S_i has outcome O_i for T . T becomes the root of the decision tree and for each outcome O_i we build a subsidiary decision tree by invoking the same procedure recursively on the set S_i .

The above procedure is applied to training sets. The training sets are delivered from the analysis based on the quantitative model presented earlier.

Suppose, for illustration purposes, that we are interested in decision making situations involving static environment only. For such cases the following rules can be delivered using decision tree classifiers (Szczerbicki 1996b):

- Rule 1
- IF an external environment of a system is static
- AND it is described by random variables

AND there is no interaction in the internal environment
 THEN communication (exchange of information) between system elements is not necessary

Rule 2

IF an external environment of a system is static
 AND it is described by random variables
 AND there is interaction in the internal environment
 AND the relationship between variables describing the external environment is of statistical character
 THEN exchange of information between system elements should be organised

Rule 3

IF an external environment of a system is static
 AND it is described by random variables
 AND there is interaction in the internal environment
 AND the relationship between variables describing the external environment is given by function dependence
 THEN exchange of information between system elements is not necessary

The use of decision trees is simple and as effective as the analysis based on a rigorous mathematical model (the production rules formulated above are the same as the rules based on quantitative modelling given in Table 1).

Signed directed graphs

A directed graph, or digraph, is a graph in which all edges are directed (Chartrand and Oellermann 1993). A signed digraph is a digraph with either + or - associated with each edge. SDG nodes are chosen as variables relevant to or representative of the problem that is studied. There is an edge from variable A to variable B if a change in A has a significant direct effect on B. The sign of the edge is + if an increase in A leads to an increase in B, and a decrease in A leads to a decrease in B. The sign is - if the effect is opposite; an increase in A leads to a decrease in B, and a decrease in A leads to an increase in B.

According to the mathematical model, information flow depends on the following state parameters: delay of information (d), amount of information (a), dynamics in the external environment (w), variance in the external environment (s), and interaction in the internal environment (q). The above parameters influence the loss in the value of information caused by delay (L1), the loss in the value of information caused by incompleteness (L2), and total loss (LV). Based on relationships and dependencies described

by mathematical model, the SDG can be developed and then simplified. Two principles are used for the simplification process. The first one is the principle of removal of intermediate nodes and the other one is the simplification of positive feedback loop.

The following logic rules can be written for the model after simplification (Szczerbicki 2002b):

SDG Rule 1:

*IF [d=+] .and. p[dLV]
 THEN it is a possible solution pattern for a positive change in d*

SDG Rule 2:

*IF [a=+] .and. n[aLV]
 THEN it is a possible solution pattern for a positive change in a*

SDG Rule 3:

*IF [w=+] .and. p[wLV]
 .and. p[wd]
 .and. p[dLV]
 THEN it is a possible solution pattern for a positive change in w*

Using the above logic rules, the qualitative behaviour of the SDG model can be found. It is easy to notice that the corresponding qualitative states (consistent patterns) for the parameters of our interest are given as follows:

(i) solution pattern for a positive change in d

d	a	w	LV
+	0	0	+

(ii) solution pattern for a positive change in w

d	a	w	LV
0	+	0	-

(iii) solution pattern for a positive change in a

d	a	w	LV
+	0	+	+

The above results of qualitative simulation are again the same as quantitative information flow modelling and evaluation. For example, they depict the adverse character of two contrary information attributes, i.e. delay and incompleteness. They also show clearly the effects of increasing dynamics in the external environment. More generally, the results show that as far as the analysis of overall directions of a system behaviour is concerned the simple qualitative model can be sufficient at a minimum level of complexity.

CONCLUSION

This paper tries to signal some of the emerging challenges and opportunities in the area of information flow modelling and simulation based on formal mathematical modelling platform. It also includes the preliminary results of some non-quantitative procedures applied in the process of knowledge acquisition for information management. The procedures show the potential for use in reasoning and retrieval of knowledge describing the flow of information between a system and its external environment as well as within a system. It was shown that the techniques applied are able to provide general knowledge about system functioning in static and dynamic external environments. The techniques presented illustrate the ease and appropriateness of such methods for dealing with implicit knowledge and also provide a model for extension into other expert domains.

REFERENCES

- Bogdan R.J. 2000. *Minding Minds*. MIT Press, Boston.
- Chartrand, G. and O.R. Oellermann. 1993. *Applied and algorithmic graph theory*. McGraw-Hill, New York.
- Matsumot, G. 1999. "Brain Computing", *Artificial Life and Robotics*, 3, 24-27.
- Morimoto T. 2001. "Application for communication agent". In *Knowledge-based Intelligent Information Engineering Systems and Allied Technologies*, N. Baba et al (Eds). IOS Press, Amsterdam, 1619-1621.
- O'Grady P. 1999. *The Age of Modularity*. Adams and Steele, Iowa City.
- Prakken B. 2000. *Information, Organisation and Information Systems Design*. Kluwer, Boston.
- Quinlan, J.R. 1990. "Decision trees and decision making". *IEEE Transactions on Systems, Man, and Cybernetics*. 20, 339-346.
- Safavian, S. and Landgrebe D. 1991. "A survey of decision tree classifier methodology". *IEEE Transactions on Systems, Man, and Cybernetics*. 21, 660-668.
- Szczerbicki E. 1996a. "External environment of an autonomous manufacturing agent: dynamics and representation". *International Journal of Systems Science*, 27, 1211-1218.
- Szczerbicki E. 2000. "Simulation modelling for complex production systems". *Cybernetics and Systems: An International Journal*. 31, 333-351.
- Szczerbicki E. 2002a. "Intelligent integration for autonomous manufacturing systems". *International Journal of Knowledge-Based Intelligent Engineering Systems*. 6, 214-219.
- Szczerbicki E. (2002b). Soft modelling support for information management. *Cybernetics and Systems: An International Journal*. 33, 413-426.
- Szczerbicki E. 2003. "Hard and Soft Modelling Based Knowledge Capture for Information Flow Management". In *Knowledge Based Information Retrieval and Filterin.*, Abramowicz, W. (Ed), Kluwer Academic Publishers, Boston, MA, 75-100.
- Szczerbicki E. 1996b. "Decision trees and neural networks for reasoning and knowledge acquisition for autonomous agents". *International Journal of Systems Science*. 27, 233-239.
- Tharumarajah A. 1999. "A self-organising model for scheduling distributed autonomous manufacturing agents". *Cybernetics and Systems: An International Journal*, 29, 461-480.

AUTHOR BIOGRAPHY

EDWARD SZCZERBICKI (MSc, PhD, DSc) has had very extensive experience in the area of intelligent systems development, information theory, and knowledge and information management over an uninterrupted 25 year period which he spent in the top systems research centres and universities in the USA, UK, Germany, Poland, and Australia. In this area he contributed to the understanding of information phenomena and knowledge management in complex systems operating in changing environments characterised by informational uncertainties. He has published a total of 200 refereed papers, over 110 of which appeared in refereed journals and the rest in international conference proceedings. With his papers published in the beginning of the nineties in *IEEE Transactions on Systems, Man, and Cybernetics* and *International Journal of Systems Science* E Szczerbicki become one of the first scientists to develop autonomous systems based on information flow for the purposes of intelligent modelling and simulation decision support. This was his unique contribution to the emerging cross-disciplinary research field of *intelligent systems* for which information has a *value*, is treated as the main *resource*, and is the base for *intelligent* decision making. E. Szczerbicki is Associated Editor, Guest Editor or a Member of the Editorial Board for seven international journals in the general area of systems science and knowledge based systems.

SIMULATION IN SUPPORT OF ROBOTICALLY AUTOMATED VEHICLE SEAT TESTING

Edward J. Williams

Mark Hebert

Ryan Raftery

Industrial and Manufacturing Systems Engineering

2330 Engineering Complex

University of Michigan - Dearborn

4901 Evergreen Road

Dearborn, Michigan 48128 U.S.A.

KEYWORDS

Process simulation, automotive industry, manufacturing simulation

ABSTRACT

An international company with significant automotive-industry operations in the Detroit, Michigan, U.S.A. area wished to assess its current capacity to absorb a new influx of business from the Asian market. Discrete-event process simulation, a powerful tool for addressing such questions, proved its value in the hands of a team comprising students undertaking a senior-level class project in simulation and their instructor. This collaboration with industry is consistent with the university's traditions of furthering students' educational objectives via practical experience while increasing the competitiveness of local industry via infusion of analytical methods and techniques.

INTRODUCTION

A large international company with extensive operations in the Detroit, Michigan, U.S.A. area is a first-tier supplier of vehicle seats to multiple automotive manufacturers. Within the last year, this company has received an influx of new business from the Asian market. The company operates large testing facilities to evaluate the durability of seat backs and seat cushions. Corporate management requested a simulation study to determine what additional facilities (employees, equipment, etc.), if any, would have to be added to the testing laboratory to accommodate the increased seat-testing work. Since discrete-event process simulation can assess the performance of any work-flow system under varying hypothesized loads, it emerged as a natural choice of technology for corporate engineers (e.g., the student enrolled in the simulation course) and managers to enjoy the benefits of generous lead time to address any problems a simulation model predicted the test facility would have under increased load. Indeed, simulation has a long pedigree of successful results when used to address such questions, and some of its earliest and most unequivocal successes have been in manufacturing industries (Miller and Pegden 2000). At

the University of Michigan – Dearborn, student teams are routinely required to undertake course projects involving immediate practical application of the analytical techniques being learned, since such projects improve students' understanding and retention of the techniques by demonstrating their contributions to a company's or organization's efficiency and competitiveness (Williams 2000).

MODEL CONSTRUCTION, VERIFICATION, AND VALIDATION

Since the simulation software used in the course (System Simulation 458, a senior-level class for industrial engineering majors and having calculus and statistics prerequisites) is Arena™ (Kelton, Sadowski, and Sturrock 2004), it was natural and convenient to use Arena™ for this project, because its standard constructs (such as *Create*, *Dispose*, *Process*, and *Assign* modules among which entities move, and *Resources*, scheduled by *Shifts*, which entities use to undertake various processes) readily sufficed to model the processes under study. The Arena™ concept of a *Station* as a generic area where an entity undergoes conveniently categorized operations facilitated the construction of a model whose overall organization was readily understandable to client managers with little or no experience as clients of a simulation study. Furthermore, Arena™ provides an environment wherein development of the model and its two-dimensional animation routinely proceed concurrently (Bapat and Sturrock 2003).

The base model was responsible for accurately representing current operations within robotic cells in the testing laboratory. These robots are used to perform “ingress/egress” tests (the industry-specific name for tests concerning a motorist's or passenger's entrance into or exit from a vehicle) to determine the life cycle of seat cushions and/or seat backs. As shown in Figures 1 and 2 four pages hence, the robots “torture test” seats by moving a “butt form” or a “butt and back form” (these terms are the industry-specific names for molded plastic forms shaped like a typical human posterior) in and out of the seat repeatedly, representing seat wear caused by a person entering and leaving the seat numerous times.

If a particular type or style of seat has never been tested before (2% of all entering seats), the program of testing motions must first be programmed into the robot assigned to the test. On a staggered schedule, the robots are taken out of service one day per month for preventive maintenance and calibration. The automotive manufacturer (customer for the seats within the supply chain) specifies whether the “butt form” or the “butt and back form” will be used. The three significant current customers are divided between usage of the two forms; the prospective new business will specify use of the “butt and back form.” At the conclusion of these tests, engineers measure the wear to the seat trim and note any structural failures occurring within the cushion frame or pan. A seat found to have excessive wear and/or a suspected structural failure (25% of all seats) must undergo a jury review conducted by trim engineers from the customer involved.

Actual construction of the base model was relatively straightforward, using fundamental modules of Arena™ such as those mentioned above. For example, the *Create* modules on the left-hand side of Figure 3 (last page) represent work samples arriving from various current customers whose test requests involve the same equipment to be used for the expected new tests, plus expected samples from the new Asian customers. After assignment (*Assignment* modules) of graphic icons and priorities, the samples are batched into groups of five, corresponding to standard scheduling policy of the robots. Next, after assignment of cycle times from the client’s database, the work samples undergo preparation, including the work of obtaining required signatures. Times needed for sample preparation were obtained from a previous 6-sigma (Harry and Schroeder 2000) project; indeed, some client managers viewed this study as a natural extension of that project. Next, the samples undergo testing; two percent of these samples (representing previously untested types of seats) required preparatory programming of the robot. One-quarter of the samples suffer extensive material wear and/or structural damage during the test; those samples must then receive a jury review performed by trim engineers. Arena’s™ diamond-shaped *Decide* blocks randomly direct 2% of the samples to robot reprogramming or jury review. All samples, after collection of performance metrics such as time-in-system, then exit the model via the shipping dock, represented by a *Dispose* block.

After this model construction, the simulation analysts then decided to run the Arena™ model on the presumption that five seats – the maximum possible – would be run through each of the two robotic cells during each test cycle, even though the current average production is less. This decision accommodated the principal goal of the simulation: to assess the ability of the current system to absorb the new business without expensive investment in new personnel and/or equipment. For example, adding a robot would cost

approximately \$97,000; adding a programmer or test engineer, \$60,000 annually or slightly more.

The most significant problems in model construction were those of data collection. Most data, such as percentage of new seat types, arrival rates of seats from various customers, and length of robotic test sequences, were routinely obtainable from managers and engineers. However, under current operational policy, line supervisors forbade direct observational data collection of the times spent by technicians moving seat samples from one test station to another. To circumvent this problem, one of the analysts unearthed data used in a Six Sigma project the previous calendar year whose original objective was to explore ways of decreasing downtime among the robots. Meanwhile, model construction continued expeditiously with data collection, a concurrency strongly advocated by (Johansson and Grünberg 2001) to reduce calendar project time elapsed – important here due to both the inevitable academic-calendar constraints and the eagerness of client management to receive guidance relative to potentially needed capital investments.

Model verification and validation (Balci 1998) were achieved by various traditional methods, including traces and step-by-step observation of the animation, structured walkthroughs (Weinberg 1971) of the model and its data, extreme-value testing (typically blending with sensitivity analysis and involving the most uncertain input data values), and direct comparison of the base model output with observations in the test facility. These last were accurate within 4% after verification and validation, incorporating correction of various problems, was deemed sufficient. As a typical example of a problem needing attention, the original travel time to a holding area was specified as NORM(1.0, 0.25) (the Arena™ notation for a random variate normally distributed with mean 1.0 and standard deviation 0.25). However, the standard deviation is dangerously large compared to the mean – in a large group of long replications, this sampled value may be expected to be negative fully 37 times in 10^6 .

RESULTS OF THE STUDY

After verification and validation were completed relative to the current system, the simulation team and the client investigated the following scenarios:

1. Add the anticipated new business to the system load without adding any resources.
2. Add the anticipated new business and add an engineer to the second shift.
3. Add the anticipated new business and add an eighth robot.
4. Add the anticipated new business and add a programmer.

Comparison of the last three scenarios with the first proved disappointing. These comparisons were undertaken using a warm-up time of one month, a run

time of one year, and ten replications to build confidence intervals, using the Student-*t* distribution, at the 95% level for performance metrics. These confidence intervals bracketed either a performance metric for one scenario, or the difference between performance metric values for two scenarios. As a precaution against non-normality severe enough to undermine use of the Student-*t* distribution, replication results were checked for normality using the Kolmogorov-Smirnov test (Law and Kelton 2000), which failed to reject the null hypothesis of normality at $\alpha = 0.10$. Additionally, since some significance levels for rejection of normality did approach 0.10, those confidence levels were recalculated using the distribution-free method of Tukey based on Wilcoxon's signed rank test (Hollander and Wolfe 1999). These confidence intervals were in excellent agreement with those calculated from the Student-*t* distribution. Results of scenario #1 demonstrated that the system could handle the increased load in terms of throughput, but with longer time-in-system than hoped. Relative to scenario #1, the remaining scenarios provided negligible increases in throughput. Scenarios #2 and #4 provided slight reductions (about 5%) in time-in-system *for current customers* relative to scenario #1. Scenario #3 provided a moderate reduction (about 10%) in time-in-system *for current customers* relative to scenario #1. In each scenario, current customers' work was assumed of higher priority than the new work. However, management deemed none of these improvements sufficient cost justification for the (expensive) investment inherent in acquisition of another engineer, another robot, or another programmer, especially since predicted times-in-service for the new business were unduly long. Additionally, sensitivity analysis demonstrated that even these modest improvements were strongly dependent on the presumed amount of new business to be provided by the new customer – and both the amount of new business and the rate of increase from current to expected load on the system (i.e., rate of introduction of the new business) were naturally uncertain in an economic sense. Therefore, client management's response to the results of the study was to reject the three proposed investments (and any possible additive combination thereof) in favor of developing contingency plans for outsourcing of work during periods of sufficiently heavy and/or unexpected customer demand. A summary of the quantitative results is presented in Table 1 below, in which the new work is prioritized

Scenario	Throughput	New Work Time in System
Current	218	Not applicable
New work	326	684 hours
+Engineer	329	685 hours
+8 th robot	332	812 hours
+Programmer	327	714 hours

Table 1. Summary of Scenario Comparisons

Results in the third column of this table clearly warned management that adding a resource (an engineer, a robot, or a programmer) without careful attention to revamping service priority policies would indeed improve overall service and particularly service to current customers – but provide such slow service to the new business that it “might well not linger long.” This warning was deemed one of the most valuable results of the study.

SUMMARY AND CONCLUSIONS

The most significant aftermaths of this project were the client's desire to use this model (to be revised) further, treating it as a “living document,” and indeed to use simulation in a variety of new situations. Significantly, this simulation study was the first ever undertaken by this client relative to the Ingress/Egress testing division of the business, notwithstanding that this work contributes about \$700,000 annually to the corporation's annual income. Therefore, the project team wrote extensive documentation, both internal and external to the model itself, to facilitate its ongoing use and modification, and make simulation, as an analytical technique, more accessible to client engineers and managers, as advocated by (Seila, Ceric, and Tadikamalla 2003). For example, the client plans to use the model to assess possible improvements achievable by changing the test batch size from its current value of five; the batch size might plausibly be dynamically determined by current system conditions instead of being fixed. Client management has now embraced the philosophy “If at first you don't succeed, you probably should have simulated it” (harrell, Ghosh, and Bowden 2004).

ACKNOWLEDGMENTS

The assistance of Professor Roger Klungle, a valued colleague and statistician, is gratefully acknowledged. Colleague Rohit Trividi provided assistance in the formatting and layout of the figures within this paper. Additionally, two anonymous referees have provided helpful criticisms and suggestions after a careful reading of the paper.

REFERENCES

- Balci, Osman. 1998. Verification, Validation, and Accreditation. In *Proceedings of the 1998 Winter Simulation Conference*, Volume 1, eds. D. J. Medeiros, Edward F. Watson, John S. Carson, and Mani S. Manivannan, 41-48.
- Bapat, Vivek, and David T. Sturrock. 2003. The Arena Product Family: Enterprise Modeling Solutions. In *Proceedings of the 2003 Winter Simulation Conference*, Volume 1, eds. Stephen E. Chick, Paul J. Sánchez, David Ferrin, and Douglas J. Morrice, 210-217.
- Harrell, Charles, Biman K. Ghosh, and Royce O. Bowden. 2004. *Simulation Using ProModel*, 2nd edition. New York, New York: The McGraw-Hill Companies, Incorporated.

- Harry, Mikel, and Richard Schroeder. 2000. *Six Sigma*. New York, New York: Random House, Incorporated.
- Hollander, Myles, and Douglas A. Wolfe. 1999. *Nonparametric Statistical Methods*, 2nd edition. New York, New York: John Wiley & Sons, Incorporated.
- Johansson, Björn, and Thomas Grünberg. 2001. An Enhanced Methodology for Reducing Time Consumption in Discrete Event Simulation Projects. In *Proceedings of the 13th European Simulation Symposium*, eds. Norbert Giambiasi and Claudia Frydman, 61-64.
- Kelton, W. David, Randall P. Sadowski, and David T. Sturrock. 2004. *Simulation with Arena*, 3rd edition. New York, New York: The McGraw-Hill Companies, Incorporated.
- Law, Averill M. and W. David Kelton. *Simulation Modeling and Analysis*, 3rd edition. Boston, Massachusetts: The McGraw-Hill Companies, Incorporated.
- Miller, Scott, and Dennis Pegden. 2000. Introduction to Manufacturing Simulation. In *Proceedings of the 2000 Winter Simulation Conference*, Volume 1, eds. Jeffrey A. Joines, Russell R. Barton, Keebom Kang, and Paul A. Fishwick, 63-66.
- Seila, Andrew F., Vlatko Cerić, and Pandu Tadikamalla. 2003. *Applied Simulation Modeling*. Belmont, California: Thomson Learning, Incorporated.
- Weinberg, Gerald M. 1971. *The Psychology of Computer Programming*. New York, New York: Van Nostrand Reinhold Company.
- Williams, Edward J. 2000. Enhancing Collaboration Among Operations Research and Simulation Educators and Practitioners. In *Proceedings of the 2nd International Conference on Simulation, Gaming, Training and Business Process Reengineering in Operations*, eds. Yuri Merkurjev, Birger Rapp, and Galina Merkurjeva, 22-26.

AUTHOR BIOGRAPHIES



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/HTM, SLAM IITM, SIMANTM, 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. His e-mail address is:

williams@umdsun2.umd.umich.edu and his university Web-page can be found at <http://www-personal.umd.umich.edu/~williams>.



RYAN P. RAFTERY is an undergraduate student at the University of Michigan – Dearborn pursuing his bachelor's in Industrial and Manufacturing Systems Engineering with a concentration in manufacturing. Throughout his undergraduate tenure at the university, he has participated in cooperative education programs with companies such as Johnson Controls, Incorporated and Lear Corporation. During these work-study programs, he has worked within a range of engineering departments, such as Advanced Manufacturing and Product Design. As a graduate of the university's Honor's College, his coursework has ranged from Bible study and World War II to manufacturing processes and facilities design. Prior to his enrollment as an undergraduate student at the University of Michigan – Dearborn, Ryan attended Dearborn High School, where he was graduated in June of 2000 with honors while serving as student council vice-president and as a two-year starter on the varsity football team. He is a current member of the International MODAPTS Association [IMA], where he is a certified practitioner of the time-study methodology, and a member of the Society of Manufacturing Engineers [SME], where he serves as the Student Chair of the University of Michigan – Dearborn chapter S126. Ryan is a senior and his future intentions include being graduated in December of 2004 and attending graduate school to earn his master's degree in business. His email address is rftery@umd.umich.edu.



MARK HEBERT is a senior at the University of Michigan – Dearborn majoring in Industrial and Manufacturing Systems Engineering with an individualized concentration, and is a member of Alpha Pi Mu (Industrial Engineering Honor Society). He will be graduated in May of 2004 and begin work on his Master's degrees in Business Administration and in Industrial Engineering. Mark is currently employed at a major automotive seating supplier in the Detroit Michigan area. His position is currently in product validation while focusing on the aspects of Six Sigma. He has earned his Design For Six Sigma Certification and is finishing his Six Sigma Green Belt training. Upon graduation, Mark will be participating in an operations rotation for his current employer, focusing on all aspects of the industry. His email address is mhebert88@yahoo.com.



Figure 1. Robot, “Butt and Back Ford,” and Seat Under Test



Figure 2. “Butt Form,” Close-Up

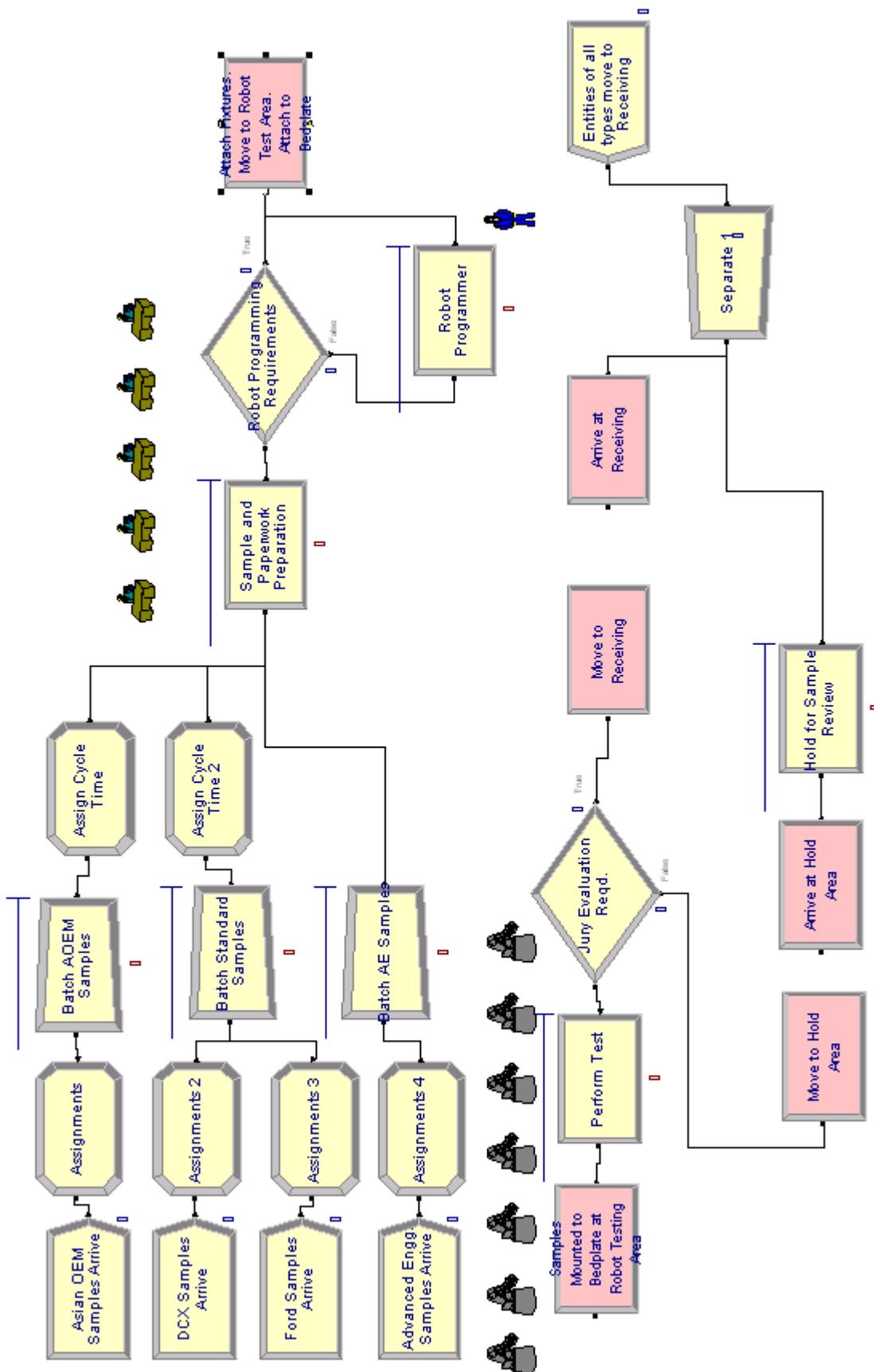


Figure 3. Screen Shot of Arena® Model

MODELING A SERVICE DISCOVERY BRIDGE USING RAPIDE ADL

Ahmed Sameh
Dept. of Computer Engineering
The George Washington University
Washington, DC 20052
Email: sameh@gwu.edu

Rehab El-Kharboutly
Dept. of Computer Science,
The American University in Cairo
P.O.Box 2511, Cairo, Egypt
Email: sameh@aucegypt.edu

KEYWORDS

SDP, ADL, Jini, UPnP, Rapide

ABSTRACT

The exploding deployment of network enabled mobile devices, along with the expansion of networked services have created the need for users to easily manage these devices and services and also to coordinate with one another. Service Discovery Protocol (SDP) enables networked devices, applications, and services to seek out and find other complementary networked devices, applications, and services needed to properly complete specified tasks. A variety of Service Discovery Protocols have been proposed by the market and academia, including Jini, UPnP, SLP, Salutation and Bluetooth. For these protocols to co-exist, they should exhibit interoperability features. A number of bridging techniques have been proposed and implemented. Efforts have been on going to analyze these bridges from an architectural point of view. A most suitable means for such purpose is Architecture Descriptive Languages (ADLs). ADLs, like Rapide, enable the simulation of distributed systems such as Service Discovery Protocols. In this paper we propose a one directional bridging system (Jini-UPnP Bridge). To validate the proposed system, we model and simulate the bridge using Rapide ADL simulation and analysis toolset. We perform a number of simulation tests and use the Rapide Poset viewer to analyze the simulator's output Poset tree of events. The bridge overhead, compared to a non-bridged native Jini service was found to be about 93.5%. The bridge performance was measured under both light and heavy network loads. Under light loads the bridge achieved 0.071% improvement, while its performance has degraded 0.034% under heavy load. The bridge performance was also measured when bridging multiple services. The results fall in reasonable ranges from 1.00079s to 1.00143s for the overall bridging time. To further validate our model, we performed a set of experiments to test communication failures.

INTRODUCTION

The number of networked services is expected to increase enormously in the incoming era. Other than traditional services (e.g. printing, scanning and faxing), new networked-services for business purposes, such as network based computational systems, or light weight

services, such as restaurant directories and translators, are becoming available and highly important. For an effective use of these services, users should have means for direct and easy access to these services. Service Discovery Protocol (SDP) presents an attractive solution for services discovery and coordination (Bettstetter and Renner 2000).

One of the main factors of judging the efficiency of a given SDP is its ability to interoperate with other SDPs. Interoperability is a vital issue since it would enable services and clients with different service discovery protocols to communicate and interact with one another. Some of the SDPs use a proxy or bridge as a solution to enable services that don't support their SDP to nevertheless have role in their federations.

In this paper, we present a new approach for bridging between Jini and UPnP. We use architectural modeling to develop a Jini-UPnP Bridge. We validate our work by carrying out a series of simulation tests and experiments on the executable architectural model. Initially, we set a hypothetical topology of Jini and UPnP clients and services in addition to our proposed Jini-UPnP bridge. This setup is used to verify that the Jini-UPnP Bridge is capable of registering a UPnP Service that offers a JiniFactory, with the Jini Lookup service. The basic functionalities of The Jini-UPnP Bridge are tested and verified. We assess the performance of the Jini-UPnP Bridge through a number experiments including: 1- measuring the overhead of bridging a UPnP service versus direct registration of a Jini native service, 2- measuring the performance of the bridge under both light and heavy network loads, 3- deducing the performance of the bridge on bridging multiple UPnP JiniFactory services. Moreover, we performed the set of experiments conducted by Dabrowski and Mills in (Dabrowski and Mills 2001) to test the behavior of our hybrid-bridging environment in cases of communication failures. We compared their results to ours to validate the correctness of our model (El-Karboutly 2002).

This remainder of this paper is organized as follows: In Section 2, we describe the proposed Jini-UPnP bridging technique. First we give a high level design view and then we present some implementation details. Our tests and experimental work is discussed in Section 3. We conclude in Section 4.

THE PROPOSED Jini-UPnP BRIDGE

One of the main factors of evaluating and judging any of the available SDP protocols is the extent to which it allows for interoperability. A bridge between UPnP and Jini has not been investigated before; though it has been mentioned as possibility in a number of references (IBM 1999) (Richard 2000) (ADL 1997).

Both Jini and UPnP introduce the concept of bridging a foreign network device as part of their specifications. Jini refers to it as a network proxy (Luckham 2001). While UPnP refers to it explicitly as a UPnP Bridge (Wang 2003). In both SDPs, the bridging concept is based on introducing a foreign device to the SDP environment through the use of a representing entity that speaks on its behalf (a bridge).

The choice of bridging Jini and UPnP is based upon the fact that both protocols, though similar at the core functionality level, have dissimilar points of strength. Both Jini and UPnP support the same set of basic SDP operation, including service advertisement and service discovery. They both support the concept of leasing for registered services and support eventing and notification mechanisms for updating service information. Jini a centric protocol, based on the presence of a central cache manager, is an example of three-party protocols, which cannot function without a Lookup Service. On the other hand, UPnP is decentralized and is more of a peer-to-peer communication model. Compared to Jini, UPnP is a lightweight protocol. This is due to the fact that Jini requires the presence of a JVM for all its entities. Bridging between Jini and UPnP will enable thin services that don't have a JVM to announce their services to Jini clients. Jini's most attractive feature is the ability of downloading services driver's or proxy, which enables easy and direct usage of the service.

Our work is built on the concept of a Jini network proxy described in Jini Device Architecture and is based on the efforts of Eric Guttman in (Guttman and Kempf 1999). A Jini-UPnP Bridge is an entity that enables services that support UPnP protocol to be reachable by Jini clients. For Jini clients, Jini-UPnP is a transparent layer that they are unaware of. The UPnP services that are advertised via the bridge are treated as native Jini services.

The proposed Jini-UPnP Bridge is modeled as a special network node that can communicate with other network nodes in both Jini and UPnP protocols. It mainly acts as a *Service User* (i.e. Control Point) in UPnP environment and a *Service Manager* (Service) in Jini environment. It waits for announcements made by UPnP devices and services that are willing to advertise their presence to the Jini clients and acts as a representative, almost a mirror for them in the Jini environment.

The first order of business of the proposed bridge is to prepare an appropriate entry for UPnP services, in the Jini

Lookup Service. This involves primarily setting the appropriate attributes required and creating a service object as part of Jini service's registration.

UPnP services that are willing to advertise their presence to Jini clients are not required to have a JVM installed. They are mainly required to have a **Jini driver Factory** (Guttman and Kempf 1999). A Jini driver factory is a (*.jar) file that bares a manifest for the advertised service. A Java Archive File (*.jar) file is used to bundle multiple files into a single archive file. Typically a JAR file contains the class files and auxiliary resources associated with applications.

The proposed bridging process is done through the following steps:

The Jini-UPnP bridge searches the UPnP reachable entities to find devices and services that have Jini driver Factory or waits till it receives announcements made by Jini driver Factory services.

Once a Jini driver Factory service is found, the Jini-UPnP bridge obtains a complete description of the service including attributes, GUI URL and control URL.

The URL of the Jini driver factory is composed by extending the control URL with a unique identifier. The Jini driver factory is downloaded using GET method over HTTP.

The Jini-UPnP bridge performs attributes transformation from UPnP format to Jini format to prepare for service registration. Upon successfully translating the entire service attributes and obtaining the Jini driver factory, the Jini-UPnP bridge registers the discovered service with **Jini Lookup Service**. Using the Jini driver factory, the bridge creates a service object that is used for registration. Registration is done by sending a join request with all necessary attributes to **Jini Lookup Service** that adds the new service to its cache.

Whenever a Jini client needs our bridging service, it contacts **Jini Lookup Service** and downloads the instantiated object that is used to drive the service. Like any typical Jini service, the Jini-UPnP bridge should be equipped with JVM to be able to participate in the Jini SDP.

The first step in modeling our bridge is to set a hybrid Service Discovery environment, where different services and clients speak different service discovery protocols. This means that we would have n Jini services, m Jini clients, e Jini lookup services, p UPnP services and q UPnP clients, where n,m,e,x,p,q are natural numbers > 0 and by setting them we define our topology. This topology would be ADL modeled such that entities are able to perform normal service discovery operations with no conflicts.

Having the two NIST Rapide models for Jini and UPnP (Dabrowski and Mills 2001) , we merged the two models into one model with both Jini and UPnP interfaces and main modules in preparation to build our proposed bridge. The proposed Jini UPnP bridge is basically a network node that acts as a *UPnP SM* in UPnP environment and a *Jini SU* in Jini environment. It's basic sub modules are the basic components of UPnP SM and Jini SU models, in addition to sub modules that perform bridging.

The main sub modules of Jini-UPnP Bridge architecture are:

UPnP Service User (UPnP SM): is a modified implementation of the UPnP SM entity that also includes ***UPnP Local Cache Manager*** and the ***UPnP SU Filter***. The ***UPnP Local Cache Manager*** is modified such that it handles attribute translation from UPnP to Jini and also Jini driver factory download.

Jini Service Manager (Jini SM) : is a modified implementation of the Jini SU that communicates directly with the UPnP SM module of the bridge to receive bridged services.

In normal UPnP SU, the local cache Manager module is an interface for the internal cache of the SU. It handles UPnP discovered service records, notifications and events. In our bridged model it also handles the functionality of managing a cache for the Jini driver factory of the discovered Jini Driver Factory services. It implements the interface `MANAGED_RESOURCE_JAR` which exposes two methods: `SUGetJar` that requests downloading a jar file for a given Jini Factory service, and `SMJarResponse` which is the response to a `SUGetJar` request. `MANAGED_RESOURCE_JAR` is represented in Rapide ADL as follows:

```
TYPE MANAGED_RESOURCE_JAR IS INTERFACE
ACTION
OUT
    SUGetJar
        (SU_ID, SM_ID : IP_Address; -- Source SU, target
SM
    QueryIssueTime : TimeUnit; -- time query issued
    URLField : Integer); -- This should be a URL
or a Device ID for identification purposes
IN
    SMJarResponse
        (SM_ID, SU_ID : IP_Address; -- Sending SM,
Receiving SU
    UniqueID : Integer; -- Unique Identifier for
SD
    Jar : String; -- a dummy string representing
the downloaded file
    TimeStamp : TimeUnit);
END;
```

Upon discovering the presence of a UPnP Service that provide a Jini Driver Factory, the Jini UPnP Bridge; first retrieves its complete description and downloads its jar file and then advertises its presence to the Jini Lookup Service. To perform the last functionality, Jini UPnP Bridge uses the interface `ADVERTISE_SERVICE`. `ADVERTISE_SERVICE` is responsible for propagating discovery of new service, change of a currently discovered service and deletion of a service to the JINI SM sub modules of the bridge. It is called by the Bridge Local Cache Manager sub module and implemented by the Jini Service Repository sub module.

`ADVERTISE_SERVICE` interface is presented as follows in Rapide ADL:

```
TYPE ADVERTISE_SERVICE IS INTERFACE
ACTION
OUT AddNewService(?Service_ID : Integer; -- ID of the
service
ServiceType, -- service type /name
ServiceAttributes, -- service attributes
ServiceAPI, -- service Proxy and APIs
ServiceGUI : String; -- service GUI
NLeaseTime,
NDuration : TimeUnit
- lease duration),
    ChangeServiceEv (?Service_ID : Integer; -- ID of the
service
ServiceAttributes:String -- new service
Attributes ),
    DeleteServiceEv (?Service_ID : Ind_Service_ID; --
Service ID
ExpireOption : String --Expire Option);
END; --ADVERTISE_SERVICE
```

A UPnP service that wishes to be used by Jini clients through our Jini UPnP bridge, should provide a Jini driver factory. The Jini UPnP Bridge issues an HTTP Get command to download the Jini driver factory file. A change was necessary to the UPnP SM Rapide Model for providing this functionality. The `MANAGED_RESOURCE_JAR` interface, introduced in the last section, is added to the UPnP Service Manager Model to be implemented by the UPnP SM_Repository sub module.

The overall Rapide model for a hybrid SDP environment with Jini UPnP Bridge consists basically of six different types of network entities: Jini SM, Jini SU, Jini SCM, UPnP SU, UPnP SM and Jini UPnP Bridge. Each of these modules implements the basic functionality of UPnP and Jini SDP Protocols. The Jini UPnP Bridge modules implements protocols of Jini SM and UPnP SU in addition to bridging functionality.

On the **network level**, the Jini-UPnP bridging environment consists of network nodes that are connected through communication links. Communication links are

mainly TCP/IP and UDP connections that are used for multicasting and unicasting messages. These communication links are modeled in our Rapide ADL as separate entities representing different multicasting and unicasting functionality.

The six network nodes: Jini SM, Jini SU, Jini SCM, UPnP SU, UPnP SM and Jini UPnP Bridge consist of **major functional components**. These are shown on the **Entity Major Functions** layer or the third layer from top. For Example the **Jini Service Manager** entity consists of **a Service Repository** and **SCM discovery** modules.

The lower level in the architecture shows the **main functional subcomponents**. These are the main components that carry out the main functionalities in the system. Some of these subcomponents are modeled as a Rapide interface and are implemented by different higher level models, while the rest are implemented as independent low level functionality modules. The main functional subcomponents of the **SCM Discovery** module, which is a basic module required in all Jini entities, is divided into three groups: Direct Discovery Protocol subcomponents, Aggressive Discovery subcomponents and Lazy Discovery subcomponents. Subcomponents that implement **Lazy Discovery Protocol** are: the **Announcement Responder**, which listens passively for announcements from entities that the SCM may wish to discover, the **Announcer** subcomponent, whose role is to send announcements to entities that may wish to discover the SCM to which it belongs, the **SCM API Server**, which provides service interfaces (APIs) to discovering entities after the initial response by the discovering entity to the SCM announcement and the **Executive** subcomponent whose main task is to control switching between aggressive, lazy and directed discovery.

Jini-UPnP BRIDGE TESTING and PERFORMANCE MEASURES

The next step after modeling the bridging between Jini and UPnP is to verify that the basic functionality of the bridge is correct through simulation tests. The Rapide toolset provides a set of compilation and runtime execution tools whose output is a simulation of the Rapide architectural model. The output of the simulation could be analyzed in various ways, including constraint checking, analysis for surprises and depiction of behavior. We chose to analyze the output of our simulation using the **Partial Order Set (Poset)** browser. Poset browser enables us to view how a given architectural design behaves. It represents casual event simulations in a DAG form, nodes representing events and directed arcs representing causality.

In each of our tests, we first establish initial conditions by constructing a topology of Jini and UPnP basic entities in addition to the Jini-UPnP Bridge. The following tests have been conducted and proven successful: 1- testing to validate that initial discovery and advertisement activities

in our hybrid environment of both Jini and UPnP entities, function correctly, 2- testing a complete scenario of bridging a Jini Service to examine the correctness of the bridging process, 3- testing that the proposed UPnP Jini Bridge successfully propagates changes that occur in the JiniFactory service to the SU Jini clients that have previously discover it, 4- testing to confirm that the JiniFactory service shutdown is propagated successfully to Jini SCM through Jini-UPnP Bridge.

We have conducted five experiments to measure the performance of the proposed Jini-UPnP bridge. In the following we discuss and report only four of them, naming: 1- measuring the overhead of bridging a UPnP service verses direct registration of Jini native service, 2- measuring the performance of the bridge under both light and heavy network loads, 3- deducing the performance of the bridge on bridging multiple UPnP JiniFactory services. Moreover, we performed the experiments conducted by Dabrowski and Mills in (Dabrowski and Mills 2002) to test the behavior of our hybrid-bridging environment in cases of communication failures. We compared their results to ours to validate the correctness of our model (El-Karboutly 2002).

The usage of a bridge in a hybrid system implies the presence of an overhead in time and resources. We are interested in measuring the overhead of bridging a UPnP service compared to having that same service as a native Jini service. The overhead is measured in terms of time and the number of messages exchange.

The following table shows the most relevant parameters and values for our experiment.

Table 1 Jini-UPnP Rapide Model Input Parameters

	Parameter	Value
General Parameters	Simulation overall time	3600s
	Node Startup Delay	1-15 s uniform
Behavior in both Jini and UPnP architectures	Polling interval	180s
	Registration TTL	1800s
UPnP specific behavior	Announcement interval	1800s
	Msearch query interval	120s
	SU purges SD	At TTL expiration
Jini specific behavior	Probe interval	5s (7 times)
	Announce interval	120s
	SM or SU purges SD	After 540s with REX only
Jini UPnP Bridge specific behavior	Jar file size	11Kb

Transmission and processing delays	UDP transmission delay	10 μ s constant
	TCP transmission delay	10-100 μ s uniform
	Per item processing delay	10 μ s for cache items
		10 μ s for other items

First, we ran the Jini Rapide model with a topology of one Jini Service Cache Manager (SCM), two Jini Service Users (Jini SUs) and one Jini Service manager (Jini SM), where one of the Jini SUs requests a service of the same type as that offered by the Jini SM. We measure the time taken and the number of messages exchanged since the Jini SM starts up and until the Jini SU receives the service description. Next, we run our Jini-UPnP Bridged model with a topology of one Jini SCM, two Jini SU, one Jini SM, one Jini-UPnP Bridge, one UPnP SU and two UPnP SM. The time taken by a Jini SU to discover a requested UPnP service is measured. This time value is the sum of the time taken for Jini UPnP Bridge to discover the services; the time the bridge registers this service with the Jini SCM and the time the Jini SCM forwards the service description to the interested Jini SU.

Measurements for Jini are done on two stages; first we measure the time taken for Jini SM to register with SCM and the number of messages needed. We assume that SCM discovery has already taken place. The time taken for this operation, as shown in the results is **TIME TAKEN 1 = 0.064s**, and the number of messages exchanged is four messages (NUM MSGs 1 :4). The second stage is where the SCM starts matching the newly added service description to the available SU requests. Two messages are exchanged for this operation to complete and the total time needed is **TIME TAKEN 2 = 0.00081s**. Thus the total time for the whole operation starting with SM registration to SU discovery takes **TOTAL TIME = 0.06481s** on average.

Bridging a UPnP SM service to be reachable for Jini SUs is done in three stages. First the Service SM is discovered by the Jini-UPnP bridge, then the bridge registers the service with Jini SCM. The time taken for a Jini-UPnP bridge to discovery and obtain the complete description of Jini Factory service is **TIME TAKEN 1:1.00132s** where five messages are exchanged in this operation. Secondly, the bridge registers the newly discovered service with the SCM by exchanging two messages in **TIME TAKEN 2:.00022**. The last stage is where the SCM matches the added service to the notification for services that SUs have registered with the SCM earlier. This operation exhausts about **TIME TAKEN 3: 0.00061s**. The total time consumed in the process of bridging **TOTAL TIME = 1.00215s**

Comparing the results for a native Jini service to that of bridging the service through Jini-UPnP Bridge, it is clear

that the bridging process has an overhead of about **0.93734s** or a 93.5% overhead.

Network Bandwidth is a main factor in the behavior of any distributed system. The performance of different entities in a SDP is very much affected by network delays as a main parameter. In our model for Jini-UPnP Bridge, we simulate network bandwidth by having network delay as one of the main model input parameters. Parameters are defined for unicast and multicast delays between any pair of nodes and also for the network as a whole. The following tests record the effect of varying network delays on the performance of UPnP-Jini Bridge.

In the pervious experiment we were interested in measuring the overhead of bridging a service in terms of time and number of messages. We fixed the TCP/IP network delay to a typical network delay value of 10-100 μ s uniform. To measure the performance of the Jini-UPnP Bridge in a light loaded network, we repeat the experiment done in the previous section with the same input parameters, yet changing the TCP/IP network delay to **10-30 μ s uniform**. The results would be compared to those obtain in the pervious section. We repeated the experiment ten times to compute the average overall time taken by the bridge.

Compared to the results obtained in the previous experiment, the bridge performance increases about 0.071 % with a less loaded network (i.e. higher bandwidth) of 10-30 μ s uniform delay. The results show an improved value for the time of registration with the bridge from 1.00132 s in normal network to 1.000617 in a less loaded network. We are more interested in the last time value (**Overall Time**) since the time taken to download the Jini driver factory is a factor of it. The results are up to our expectations since an overall improvement in time delay is noticed.

To measure the performance of Jini-UPnP Bridge in a congested network, we apply the same experiment with a higher network load with the same input parameters, yet changing the TCP/IP network delay to **80-100 μ s uniform**. The results would be compared to those obtain in case of typical network delays. We repeated the experiment ten times to compute the average overall time taken by the bridge.

Compared to the results in normal network condition that are obtained in the previous experiment, the bridge performance degraded about 0.034 % with a congested network (i.e. low bandwidth) of 80-100 μ s uniform delay. The result is as expected since the effect of having a low bandwidth is of direct effect on the time taken to transfer messages and to download Jini driver factory. The overhead in time is more obvious in the time taken for registration with the bridge, as downloading the Jini driver factory file is a factor in it.

A UPnP client (UPnP SU), in a pure UPnP environment, is capable of discovering and communicating with multiple UPnP Services at the same time. Also, a Jini Service Manager (Jini SM) could advertise and register the availability of more than one service. Our UPnP-Jini Bridge is primarily composed of both a UPnP SU and Jini SM. Thus a UPnP-Jini Bridge is capable of bridging more than one UPnP service and registering it with the Jini SCM at the same time. We are interested in testing this capability of our modeled Jini-UPnP Bridge to bridge successfully multiple services at the same time and also to depict the effect of multi-service bridging on the Bridge performance.

In the previous experiment, we've chosen a topology with one UPnP Service (UPnP SM) that offered a Jini Factory and is bridged using the UPnP Jini Bridge. In this experiment, we conduct a topology of five UPnP SMs to be bridged, one UPnP Jini Bridge, one UPnP Service User, one Jini SCM, two Jini SUs and one Jini SM. We assume the same input delays and parameters presented above. We record the time taken for a Jini SU to discover a requested UPnP service. This time value is the sum of the time taken for Jini UPnP Bridge to discover the services; the time the bridge registers this service with Jini SCM and the time the Jini SCM forwards the service description to the interested Jini SU.

The results obtained are not uniform, yet they fall in a certain time range, for example the overall time taken by UPnP-Jini bridge to bridge a given service ranges between 1.00079 s and 1.00143 s. These results are expected since the behavior of the bridge is a function of the number of events it receives at the same time and the way it schedules the incoming events. The results fall in reasonable ranges and are close to the results obtained in case of bridging one service. These results are also dependent on the time each node starts announcing its service. Nodes that announce their services consecutively with a small time variant (e.g Nodes 2, 3), cause high frequency of events on the bridge, which results in degradation in the bridge performance and higher delay values.

CONCLUSION

The problem we addressed in this research is enabling thin servers and lightweight devices to offer their services to Jini clients through passive and indirect registration using our proposed Jini-UPnP Bridge. This problem has been addressed before by using SLP instead of Jini (Guttman and Kempf 1999), yet the bridging between Jini and UPnP has not been investigated before in SDP research literature.

We modeled and simulated our solution using Rapide ADL toolkit. Modeling is an approach for designing quickly, efficiently and correctly. It allowed us to control the quality and performance. We've chosen Rapide ADL to benefit from the set of modeling and simulation tools it

offers. We used architectural models of Jini and UPnP as a basis to create hybrid discovery environment including both Jini and UPnP and to design and model our proposed bridge. For testing and simulating the bridge, we created a hypothetical topology of Jini and UPnP clients and services in addition to our proposed Jini-UPnP bridge. We simulated the topology to verify that the Jini-UPnP Bridge is capable of registering a UPnP Service that offers a JiniFactory, with the Jini Lookup service. The Jini-UPnP Bridge is tested for cases where the bridged service is updated or deleted. A number of performance experiments have been done on the bridge.

REFERENCES

- ADL 1997, "Using Architecture Description Languages (ADLs) to Improve Software Quality and Correctness in Dynamic Distributed Systems" http://www.itl.nist.gov/div897/ctg/adl/sdp_projectpage.html
- Bettstetter, C. and C. Renner, 2000, "A Comparison of Service Discovery Protocols and implementation of the Service Location Protocol", *In Proceedings of EUNICE 2000, Sixth EUNICE Open European Summer School, Twente, Netherlands.*
- Dabrowski, C. and K. Mills, 2001, "Analyzing Properties and Behavior of Service Discovery Protocols using an Architecture-based Approach", *Proceedings of Working Conference on Complex and Dynamic Systems Architecture.*
- Dabrowski, C. and K. Mills, 2002 "Understanding Self-healing in Service-Discovery Systems," *ACM Workshop on Self-Healing Systems*, Charleston.
- El-Kharboutly, R. 2002, "Modeling Jini-UpnP Bridge Using Rapide ADL", M.Sc. thesis in Computer Science, The American University in Cairo.
- Guttman, E. and J. Kempf, 1999, "Automatic Discovery of Thin Servers: SLP, Jini and the SLP-Jini Bridge," *Proc. 25th Ann. Conf. IEEE Industrial Electronics Soc. (IECON 99)*, IEEE, Press, Piscataway, N.J.
- IBM 1999, white paper, "Discovering Devices and Services In Home Networks".
- Luckham, D. 2001, "Rapide: A Language and Toolset for Simulation of Distributed Systems by Partial Ordering of Events," <http://anna.stanford.edu/rapide> .
- Richard, G. 2000, "Service Advertisement and Discovery: Enabling Universal Device Cooperation," *IEEE Internet Computing.*
- Wang, O. 2003, "Interoperability of COM/DCOM objects with CORBA objects by using DCOM/CORBA Bridge and their performance analysis", <http://www.engr.sjsu.edu/fatoohi/wang-report/abstract.html>

MODELING SEAWATER DESALINATION USING WASTE INCINERATION ENERGY - FUNDAMENTAL MODEL

Ken Udono and Renate Sitte
School of Information Technology
Faculty of Engineering and Information Technology
Griffith University, Gold Coast Campus
PMB 50 Gold Coast Mail Centre
Queensland 9726, AUSTRALIA
Email: k.udono@griffith.edu.au
Email: r.sitte@griffith.edu.au

KEYWORDS

Waste, Energy, Modeling, Dynamic System, Seawater Desalination, Incineration

ABSTRACT

This paper models the potential use of waste-to-energy conversion of municipal solid waste to power seawater desalination, using a dynamic system approach with control theory as basic methodology. Australia has been hit by one of the worst droughts causing water shortage in many places, including the city of Gold Coast. As so many cities, this city is also faced with the handling of waste generated by its inhabitants. Currently most of the waste generated from the city goes into landfills despite that waste can be a source of cheap fuel to obtain power. The Gold Coast has been the fastest growing city in Australia for quite sometime and a long-term solution for both these issues is needed. In this paper, incinerating waste to obtain electricity to power desalination of seawater is investigated. The fundamental model is implemented with a modular hierarchical structure in Matlab/SimulinkTM and the result of the simulation is presented.

INTRODUCTION

This paper models the potential use of Municipal Solid Waste (MSW) incineration to power seawater desalination in order to overcome water shortage and waste landfill issues surrounding Gold Coast city in Australia. Australia has been hit by one of the worst drought ever resulting from climate changes such as El Niño, global warming, etc. The latest drought between 2001 and 2003 was the worst in the recorded history of the Gold Coast city. A year after the drought, the city of Gold Coast is still under water restriction and the city council is forced to consider several solutions to alleviate the repeated shortage of fresh water. Despite of good rainfalls breaking the drought, the main supplier of water i.e. the Hinze dam is less than 50% of its capacity. A fast growing population of the city also adds additional strain to the city's infrastructure in the long term. The Gold Coast city is the sixth largest city in Australia, which has a population of 425,418 (GCCC 2003). The yearly population growth is estimated at

about 13000 and predicted to reach 1.2 million by year 2056 (GCCC 2003). An average daily water usage was of 156 million litres during 2003 with water restriction in place, however on a hot summer day, it can be as high as 300 million litres (GCCW 2003). By 2056, the daily water demand is estimated to increase up to 465 million litres a day, which is over 2.5 times more than the current demand of 185 million litre (GCCC 2003). Therefore a long term solution to the water shortage is mandatory and so far has not been satisfactorily found.

This paper investigates the introduction of desalination to produce fresh water from seawater, while powering the desalination plant by Waste-To-Energy (WTE) conversion of MSW. The model presented in the paper is the fundamental model that simulates how drought and desalinated water affect the water availability and how much of electricity can be potentially obtained from the incineration of the MSW. This study is motivated on waste incineration successfully being practiced in several parts of the world (Austria, Japan, Germany), for obtaining energy for different purposes. According to a report made by the city council (GCCC 1997; GCCC 2002), 437,447 tonnes of solid waste was collected during 2000-2001. 87 % of waste collected have been buried in landfill and only 13% was recovered to be recycled, reused and avoided landfill. This has been despite of almost 70% of the waste generated being combustible and potentially a source of cheap fuel for obtaining energy. Seawater desalination is well proven and widely used method for fresh water production in regions where fresh water is scarce. Saudi Arabia is the largest producer of desalinated water and has 27 desalination plants (SAIS 2003). In Saudi Arabia, the desalinated water is distributed using 2,300 miles of pipelines and fills 70% of drinking water demand in the country. With completion of additional plants, the country is able to produce 3028 million litres of fresh water per day from seawater desalination. Water desalination is an expensive production process, but it ensures continuation of fresh water supply in dry weather conditions.

There are a number of WTE conversion techniques for generating energy from waste, in different stages, ranging from already-in-use to research and

experimental level. Common techniques are thermal conversion including pyrolysis, gasification, and incineration and other types of processes such as landfill gas and ethanol production. Typically in an incineration method, the heat from combustion of MSW is used to raise steam in a boiler. It is then passed through a turbine generator to produce electricity (Baird 1993). There are several seawater techniques currently in use. Some of the major ones are distillation such as Multiple Effect Distillation (MED) and Multistage Flash Distillation (MSF) and membrane methods such as Reverse Osmosis (RO). In RO method, seawater is forced through the semipermeable membrane by mechanical pressure. It is more energy efficient than the distillation method, which uses thermal energy to evaporate the seawater. This is why RO is gaining popularity in recent years, however in oil rich countries like Saudi Arabia, where energy is not an issue, MSF is still the most often practiced method.

The model presented in this paper is an adaptation and extension to the economic model that utilised a dynamic system approach with control theory presented by R. Sitte (Sitte 1998; Sitte 2001; Sitte 2002). This approach has been successfully applied into modelling green energy export as well as the El Niño effect on Australian economy. This methodology allows focusing on a higher level of resolution, without the loss of relevant sub-models, whenever required. This approach gives us the prospect of integrating all three, the water desalination, and the El Niño and green energy effect models. Our model uses simulation to examine how the desalination assists in the city's water supply, while reducing the landfill. The model itself is implemented in Matlab/Simulink™ as a dynamic model simulating the sequence of economic cycle as it is affected by climatic fluctuations. The overall goal is to investigate whether this can be successfully achieved in an Australian city, such as the Gold Coast City while incorporating future population growth, and deficiency of rainfall during drought conditions. The model can be adapted to simulate other cities, by simply changing the local parameters.

FUNDAMENTAL MODEL

This model follows a dynamic system approach with control theory as basic discipline, as opposed to the statistical approach that is traditionally used in economic modelling (Sitte 1998). The benefit is that the high level of resolution can be better maintained and the cause and effect dynamics are more visible than in a statistical model. The long-term behaviour of the system can be observed without being side tracked by minute detail. The aim of this model is to obtain a faster result with accuracy within 5%-10% error rate.

The fundamental model has two major components: Water dynamics and Waste dynamics. The top level of the fundamental system implemented is shown in Figure

1. In the water dynamics section, the flows between blocks represent the amount of water. The waste dynamics section models the power generation from WTE conversion and the power required for the desalination of seawater. The fresh water resulting from desalination in the waste dynamics section is added to the available water in the water dynamics section. This is the only coupling between the two components.

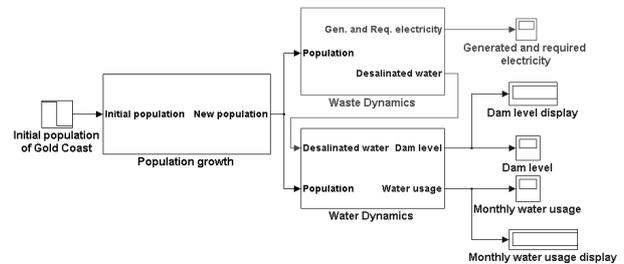


Figure 1: Top Level Featuring Water and Waste Section

The simulation is run for a monthly step size and the population is used to initiate the system. There are various types of data used in this model including water, energy, population, and amongst others. The data related Gold Coast City was taken from Gold Coast City Council (GCCC 2003) and the Australian Bureau of Statistic (ABS 2003). Local weather data such as average rainfall were collected from the Bureau of Meteorology (BOM 2003). The desalination process and incineration related data, which are mostly the result of years of operation experience, were taken from the literature. The following sections provide a brief description of each component.

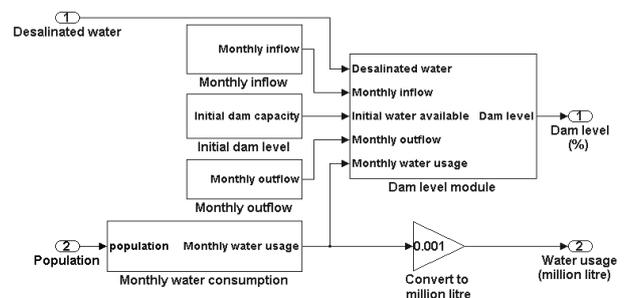


Figure 2: Water Dynamics Model

Water dynamics model

The water dynamics model (Figure 2) involves rainfall-runoff model surrounding city's major source of water, the Hinze dam, to determine the availability of water for the city. However the main focus of this research is on effects of the desalination powered by incineration, but not on building a superior rainfall-runoff model (water balance model). In this fundamental model, the black box modeling approach was chosen for the water balance model using rainfall data, dam level, catchment size etc, because of its simplicity. As only a limited amount of data is currently available, the conceptual

models or physically based distributed model, which should be used only when available data is able to support it, cannot be used in this case (Ragab 1999). An inflow of the dam was approximated as a function of monthly rainfall onto a certain percentage of catchment areas of the dam (in this case 66%). An evaporation is removed and regarded as part of the outflow. To model the rainfall, data of monthly average rainfalls were taken from the Bureau of Meteorology (BOM 2003) and approximated using a best fit cubic polynomial, to be used as a typical monthly rainfall in the model.

$$f(x) = 0.04989x^3 - 0.6083x^2 - 0.4653x + 23.71 \quad (1)$$

Equation (1) above is the polynomial approximation used in the model. The comparison of approximated and actual average rainfall is shown in Figure 3.

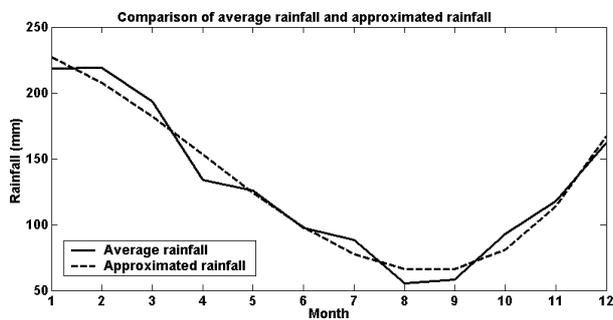


Figure 3: An Approximated and Average Rainfall

Waste dynamics model

The waste dynamics model (Figure 4) analyses and determines the amount of power generated from incineration and the energy required for the desalination plant. The WTE conversion method investigated in this model is a mass burning method of MSW. A physical composition heating value estimation model by K. Amz and A. Ghrarah (Amz and Ghrarah 1991), was chosen to estimate the heating value of waste (Equation (2)). This model uses a weighted (in %) function of heating coefficients of the incinerable components of waste: food, paper and plastic. The relative accuracy of this model was demonstrated by Abu-Qudais and Abu-Qdais (Abu-Qudais and Abu-Qdais 2000).

$$E \text{ (kJ/kg)} = (23(F + 3.6(PA)) + 160(PL)) * 2.324 \quad (2)$$

Where F, PA and PL are the proportion (%) in weight of food, paper and plastic respectively in the waste incinerated. This composition detail for the Gold Coast is unknown due to lack of recorded data. Instead, the waste composition from its neighbouring Brisbane city was chosen to be used because the waste composition the two cities would be similar (GCCC 1997). The waste compositions of Brisbane city as well as other cities are shown in Table 1 for comparison. Using Brisbane data, the heating value was estimated to be 9134.16kJ/kg in our model.

Table 1: Waste Composition from Various Cities

Waste type	Brisbane	Melbourne	Wollongong	Jordan	Ontario
Newspaper	5.20	7.20	12.50	11.45	35.80
Other Paper	13.50	16.70			
Plastic	7.70	7.70	12.80	16.15	7.50
Rubber	1.00	N/A	N/A	N/A	
Glass	10.50	6.90	3.50	2.06	7.80
Aluminium	1.00	0.60	N/A	N/A	N/A
Steel	3.00	3.30	5.70	2.06	14.00
Other Metal	2.10	0.10	0.30		
Food	28.00	31.20	52.10	62.64	16.00
Wood and Garden	21.00	20.70		N/A	N/A
Other Organic	N/A	N/A	10.70	N/A	N/A
Rags	2.00	2.20	N/A	N/A	N/A
Inert	N/A	3.40	N/A	N/A	N/A
Other	5.00	N/A	2.40	5.64	7.00
Reference	GCCC 1997	GCCC 1997	GCCC 1997	Abu-Qudais and Abu-Qdais 2000	Morris 1996

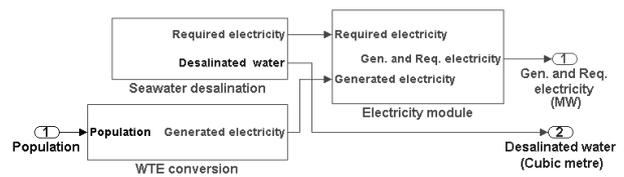


Figure 4: Waste Dynamics Model

The electricity generation from incinerating MSW typically has an efficiency of 10%-15% lower than fossil combustion, which has a 30%-40% efficiency. To estimate the potential electricity generated from incineration of MSW, an average heating value required for producing 1kw of 20089kj from various literature sources was used in this model (Baird 1993; Kathirvale et al. 2003; Morris 1996; Otomoa et al. 1997). The amount of waste generated from the city is estimated using data from the council report in 2002 (GCCC 2002), which is 1.1 tonnes per person per year. At this stage we use the assumption that the entire combustible waste included in the heating value estimation (77% of total waste generated) is incinerated for power generation in regardless of processing capacity of the incineration plant. We do this to examine how waste can power desalination and find the potential of desalination powered by waste incineration.

We also calculate the estimated electricity required for desalinating seawater in the waste dynamics model. Theoretically 0.86kWh of energy is required for conversion of 1m³ of seawater to freshwater (DESWARE 2003). However, the actual power consumption figure can be much higher, up to 20 times more depending on method used. As for our fundamental model, no particular desalination method was chosen in this estimation, but an average power consumption of common desalination methods (10.2kwh/ m³) was used to find the electricity required for desalinating seawater (DESWARE 2003). This is because further investigation is required to find the best suitable desalination method for our case using power from incinerating MSW. There are at least two possible options for implementing desalination powered by waste incineration. Such options are for example locating desalination plant and incineration plant side by

side to use heat directly from incineration plant for distillation desalination process or alternatively locating them remotely to use the electricity generated from the incineration plant for a non-thermal desalination method. At this time our immediate interest is to investigate how much of desalination process can be powered from incinerating the MSW in general term.

SIMULATION EXPERIMENTS AND RESULTS

This section explains the calibration of the model conducted prior to examining the effect of each element on water availability. Several experimental simulations and their results are then discussed by investigating the effects of each chosen parameter for a period of 10 years.

Model Calibration

The purpose of the calibration is to obtain stable conditions represented by a flat (or horizontal wobble) curve of the dam level (%) over a medium time span (in this case for 10 years). This is done by fine-tuning parameters and some of the approximated values in the water dynamics component. The calibration was done using the average rainfall with and without population growth. The initial population was set to 425,418 (year 2001 figure). The calibrations were repeated four times for different initial dam levels: 25%, 50 %, 75% and 100% for a span of 120 months (10 years). Figure 5 shows the dam level output from the calibration using an average rainfall without population growth.

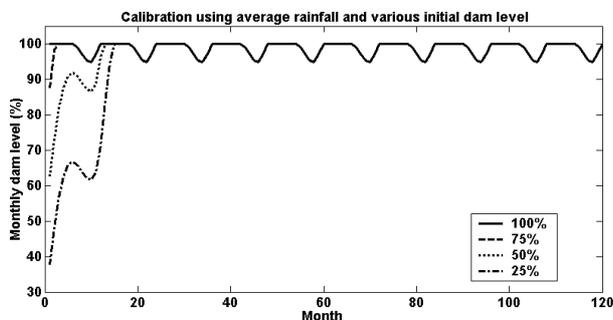


Figure 5: Calibration Results Using Average Rainfall

With this calibration setting, it maintained a relatively flat horizontal wobble after reaching full capacity. The wobble is caused due to variation in the seasonal rainfall on the Gold Coast. Typically winter weather conditions are much drier, causing a definite drop of dam level by about 7% during winter. With an initial level of 100%, it never drops down below 100% until the winter month of July. When the initial dam level was decreased to 75%, it took several months to restore its full capacity. If the initial level was further dropped to 50% and then 25%, the number of months taken to reach full capacity increased to 12 months and 14 months respectively.

Same calibrations were also repeated with projected population growth. Overall, this calibration provided almost identical result to the first calibration without population growth. This indicated that effect of population growth on overall water availability in this simulation duration (10 years) is fairly low, however the effect can become greater for longer term. The only obvious difference between these two simulations was higher amplitude of the wave, which is gradually becoming greater, during wintertime. The higher consumption of water from increased population results this increase of amplitude. Overall these calibrations gave satisfactory result of a slightly oscillating but generally stable dam capacity level as expected.

We verified the short-term behaviour of the water dynamics model, using recorded data of the Hinze dam level and the rainfall during latest drought between February 2001 and October 2003. Figure 6 shows a comparison of our simulated dam level and the actual dam level recorded.

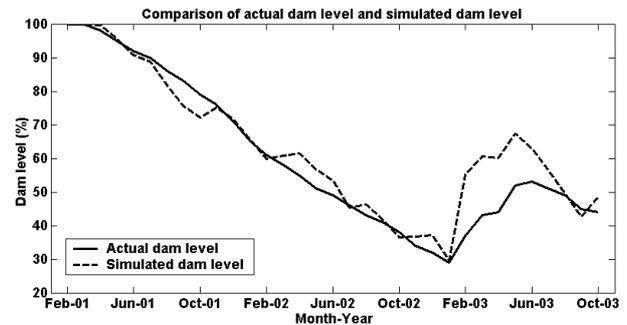


Figure 6: Dam Level Result Using Latest Drought Data

As it can be seen, our model tends to overestimate the dam level, particularly in second half of the graph. This is because due to the long drought condition during 2001-2002, the catchment area had dried out and the rainfall during early 2003 was ineffective. The rainfall was absorbed into the ground before it reached the dam instead of flowing into the dam. However our black box model approximates inflow into dam directly using rainfall without reflecting geographic characteristic or weather conditions of the catchment. This results overestimation of inflow in the case of rainfall after a long period of dry weather condition. Despite of this, the overall error percentage was found to be about 10% although highest error percentage being 50%. At this time, this error rate is acceptable because a vast amount of data is still missing and other modeling methods are also being considered.

Drought effects on water availability

Following the calibration, in order to examine the effects of each chosen parameter on dam level for a period of 10 years, we performed a set of systematic simulations. The first set of experiment was to investigate the effect of drought on the dam level and

the recovery of the dam level from various lengths of drought. To replicate drier weather during drought, the average rainfall was reduced by 50%. This is justified because in the drought year of 2002, the rainfall was about 50% of its normal average. The drought was set to start at beginning of the third year of the simulation with three different lengths of drought: 1 year, 2 years and 3 years. Figure 7 shows the result of these drought simulations.

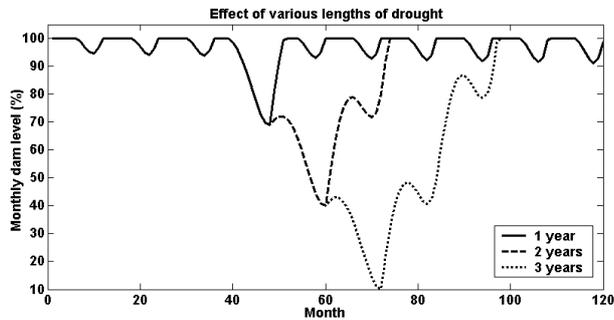


Figure 7: Drought Effects on Dam Level

In all cases, the dam was able to recover from drought within the simulation period. In the case of 1 year drought, it took 4 months to reach full capacity after the end of the drought. For the case of 2 years drought, 14 months were needed to reach full capacity. In the case of 3 years drought, 25 months of recovery were required. The lowest dam levels reached in each case were 68.98%, 40.08% and 10.06% at the last month of drought year respectively. From this simulation, it can be seen that with a drought of 3 years or more there is already a danger of running out of fresh water. Considering that we actually have now less than average rainfall after two years of severe drought, these simulations appear realistic.

Desalination effects on water availability

The purpose of the next two simulations is to examine the influence of a desalination facility on the availability of water. For convenience we measure this availability again on the dam level, although transporting the water from sea level up to the dam might not be very realistic, but these are issues beyond the purpose of our research. Three different capacity sizes of desalination plants, 10,000m³/d, 20,000m³/d and 40,000m³/d were simulated in this experiment. Figure 8 shows the comparison of the dam level using rainfall data between February 2001 and October 2003 with three different desalination plants as well as without any desalination plant. In the case of no desalination as previously shown in Figure 6, it reached lowest dam level of 29.56%, but with introduction of desalination, it improved to 33.6%, 37.6% and 45.4% using 10,000 m³/d, 20,000 m³/d and 40,000m³/d of desalinated water respectively. An average improvement of the dam level from desalination was found to be 2.73% per 10,000m³/d throughout the simulation period.

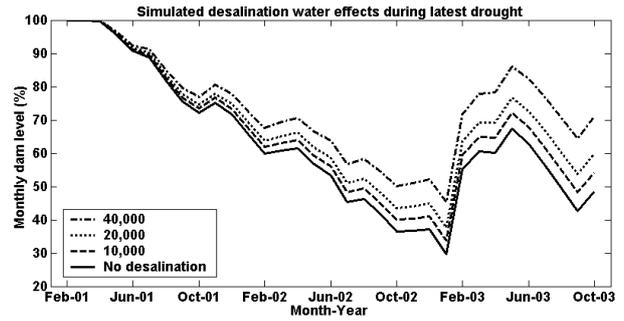


Figure 8: Effects of Desalination During Latest Drought

Figure 9 shows results of the dam level simulation during 3 years drought with same desalination capacity setting. This simulation was conducted to examine how desalinated water improves water availability during 3 years drought, which was previously shown in Figure 7. During the 3 years drought without desalination, it reached the lowest dam level of 10.06%. With inclusion of desalinated water, it was improved to 16.20%, 22.34% and 34.62% respectively. An average dam level improvement per 10,000m³/d during the drought period was 2.82%.

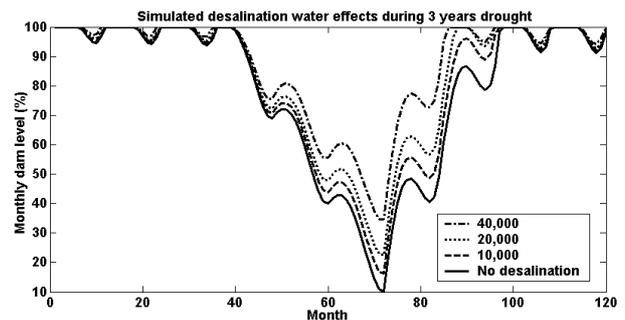


Figure 9: Effects of Desalination on Drought of 3 Years

Electricity generation and requirements

Figure 10 shows the comparison of electricity generated from incinerating MSW and electricity required for desalinating 20,000m³/d of seawater from waste dynamics model. The result is from an early stage of our model development for initial comparison. It showed that it is possible to generate twice as much as electricity required for desalination process from incineration initially. At the end of the simulation, it simulated three times more electricity than electricity required for desalination due to increased MSW generated from population growth. However this may not be realistic, because while there might be sufficient waste the incineration facility may not have capability of processing all of MSW from the city. Moreover, the amount of energy required for desalination can be potentially reduced or possibly increased when the most suitable desalination method is determined at later stage of the research.

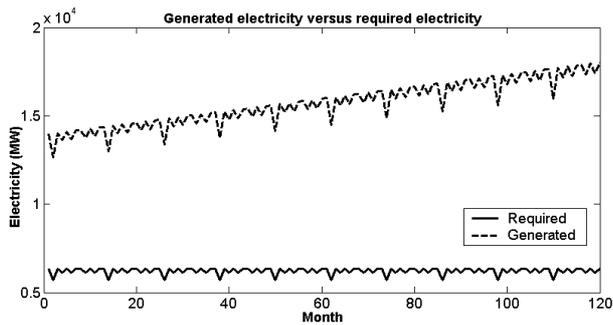


Figure 10: Generated Versus Required Electricity

CONCLUSION

This paper presented our fundamental model for modelling seawater desalination powered by waste incineration using a dynamic system approach, with underlying control theory. The fundamental model was built in Matlab/SimulinkTM. This version of our model was implemented following investigation of desalination and incineration technologies described in the literature and the collection of Gold Coast city data including population, weather, water and waste. The water dynamics model was calibrated to current conditions and several experiments were conducted using a 10 years period. The water dynamics model incorporated population growth and subsequent water demand increase. Our model was found to overestimate the dam level of the Hinze dam due to the approximation method of inflow (rainfall/runoff) under the current absence of an accurate estimate of the catchment area. However, in the verification process using data between 2001 and 2003, the average error was found to be about 10%. The effects of additional water from desalination on the dam level during a drought were also observed using recorded data as well as long drought conditions. The initial results from the waste dynamics model revealed that more than twice of electricity required to desalinate 20,000m³/d of water could be obtained from waste incineration. Future work for the advanced model includes integration of El Niño effect and the economical impact, as well as alternative renewable energy.

REFERENCES

- Australian Bureau of Statistic (ABS). 2003. <http://www.abs.gov.au/>.
- Abu-Qudais M. and H. A. Abu-Qdais. 2000. "Energy content of municipal solid waste in Jordan and its potential utilization". *Energy conversion and Management*, Vol. 41, 983-991.
- Amz K. and A. Ghrarah. 1991. "New approach for estimating energy content of municipal solid waste". *Journal of environmental engineering*, Vol. 117, 376-380.
- Baird S. 1993. "Energy from Waste". *Energy Fact Sheets*, The Energy Educators of Ontario, <http://www.iclei.org/EFACTS/ENRGYWST.HTM>.
- Bureau of Meteorology (BOM). 2003. <http://www.bom.gov.au/>.
- DESWARE. 2003. "Energy requirements of desalination processes". <http://www.desware.net/desware/desa4.asp>.

- Gold Coast City Council (GCCC). 1997. "State of the Environment – City of Gold Coast Benchmark '97". http://www.goldcoast.qld.gov.au/t_std2.asp?PID=2267.
- Gold Coast City Council (GCCC). 2002 Solid Waste advisory Committee. *2002 Vision on Waste: A solid waste management strategic plan for Gold Coast City*. Gold Coast City Council (GCCC). <http://www.goldcoast.qld.gov.au/>.
- Gold Coast City Water (GCCW). 2003. <http://www.goldcoastwater.com.au/>.
- Kathirvale S.; M. N. Muhd Yunus; K. Sopian; and A. Halim Samsuddin. 2003. "Energy potential from municipal solid waste in Malaysia". *Renewable Energy*, Vol. 29, 559-567.
- Morris J. 1996. "Recycling versus incineration: an energy conservation analysis". *Journal of Hazardous Materials*, Vol. 47, 277-293.
- Otomoa S.; U. Mori; A. Terazono; T. Aso; and R. Sameshima. 1997. "Estimation of energy recovery and reduction of CO₂ emissions in municipal solid waste power generation". *Resources, conservation and Recycling*, Vol. 20, 95-117.
- Ragab R.; K. Blyth; M. France; I. Littlewood; C. Loumagne; C. Michel; M. Monfort; C. Perrin and A. Weisses. 1991. "Survey of modelling approaches suitable for AIMWATER". *European research project "AIMWATER"*, September 1999, http://dataserv.cetp.ipsl.fr/AIMWATER/reports/modelsurvey_report.pdf.
- Saudi Arabian Information Source (SAIS). 2003. <http://www.saudinf.com/main/a541.htm>.
- Sitte R. 1998. "Modeling The Propagation of El Niño Climatic Oscillations on Australian Economy – Part I: The basic model". In *Proceedings of the IFAC/SICE Workshop on Control in Natural Disasters* (Tokyo, Japan, Sep. 21-22). Elsevier, 35–38.
- Sitte R. 2001. "Modelling Green-Energy Export in Australian Economy - Part II Advanced model". In *Proceedings of the IFAC Workshop on Modelling and Control in Environmental Issues* (Yokohama, Japan, Aug. 22-23). Elsevier.
- Sitte R. 2002. "Zooming-In Modelling Method For Managing Complexity". *International Journal of Modelling and Simulation*, Vol. 21, No. 4, 104-109.

AUTHOR BIOGRAPHIES

KEN UDONO has received his Bachelor of Information Technology with Honours from Griffith University in 2001. He is currently PhD student in the Faculty of Engineering and Information Technology, Griffith University, Australia under supervision of Dr. Renate Sitte.

RENATE SITTE obtained her PhD (1995) and MPhil (1993) from Griffith University in Microelectronic Engineering. She received her Systems Engineering degree (Ingeniero de Sistemas) in 1985 from the Universidad de Los Andes in Venezuela. She is a Senior Lecturer in the Faculty of Engineering and Information Technology, Griffith University, Australia, where she has been faculty staff since 1986. Her field of research is in modeling and simulation, one major ongoing project is Virtual Reality Prototyping Micro-electromechanical systems (MEMS).

DISTRIBUTED SIMULATION OF AN EMERGENCY SYSTEM FOR THE FLOOD DISASTER IN HAT YAI, THAILAND

Chirawat Wattanapanich
Pichaya Tandayya
Department of Computer Engineering
Faculty of Engineering
Prince of Songkla University
Had Yai, Songkhla 90112 Thailand
Email: s4412013@maliwan.psu.ac.th, pichaya@coe.psu.ac.th

KEYWORDS

Flood, Distributed Simulation, Emergency System, Rehearsal, HLA

ABSTRACT

The paper presents an approach to create a training simulation system for emergency rehearsal in case of flooding at Hat Yai municipality, Thailand. The simulation system applies the concept of demand and supply as well as distributed interactive simulation using High Level Architect Framework.

INTRODUCTION

In many areas, flooding is a natural disaster that causes enormous damage both physically and economically. It has been the cause of catastrophes in the past and at the present. No community areas are absolutely free or safe from flooding. There are still flood problems occurred in many countries including Thailand. It is also not practical to evacuate communities or industrial sites out of the locations in order to avoid the flooding problems when the flooding does not really occur every day. Therefore, human beings need to learn to prepare, prevent and deal with flooding situations.

There are many ways to prevent or reduce damage caused by flooding, for example; on engineering issues such as building dams, installing water pumps, surveying risky areas; and on social issues such as giving knowledge of self care in flooding situations and rehearsing aid units or related organisations. In this paper, we present work on preventing and reducing damage to communities after flooding occurred, focusing on virtual rehearsals of related aid units.

BACKGROUND

High Level Architecture(HLA)

HLA is a software architecture that allows creating distributed simulation. The HLA has

been adopted by the United States Department of Defense (DoD). In September 2000, the HLA was approved as a open standard by IEEE and in mid 2002, the HLA framework has been adopted to full commercial application.

In HLA, a simulation node called a *federate*. By using *OMT (Object Model Template)*, HLA can integrate many individuals models or federates to form a complex system that called a *federation*. Federates can communicate with one another through *Runtime infrastructure (RTI)* that is the HLA middleware. The RTI software allows federates to cooperate together, to synchronize federate events and to form a federation as shown in Figure 1. Information of every federate must be sent to RTI first. Then, RTI will send the information to the destination or desired federates by the mechanism of publication and subscription.

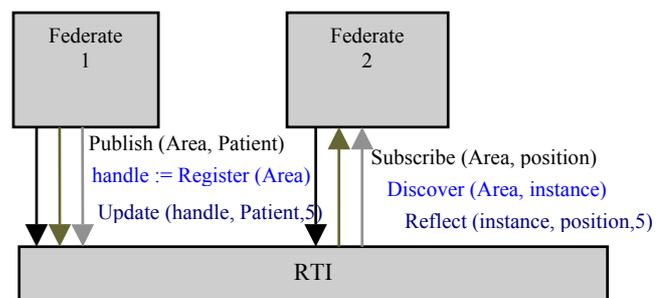


Figure 1. Federate/RTI Interface

Publication and subscription mechanism is analogous to newsgroups because the publishing federate or producer of information must define a means of describing data it is producing and the subscribing federate or receiver must define a means of describing the data it is interested in receiving in form of OMT. (Wyne 1999, Kuhl et al 1999, Fujimoto et al 2000, IEEE STD 1516)

Physical system

This work concerns a case study of Hat Yai municipality in Thailand where there were two recent big floods considered a century case occurred successively in only about a decade (1988 and 2000). Hat Yai municipality with the area of thirty one square kilometers is located in the Intertropical Convergence Zone. 80% of the area is a flat basin. There are two canals called Uhtapao and Toey. The flood type is a flash flood that once the soil is saturate with water, water can be collected and floods the municipality rapidly in a few hours. It normally happens after several days of heavy rain.

The structure of Hat Yai municipality in aiding flooding situations is shown in Figure 2 and consists of the followings.

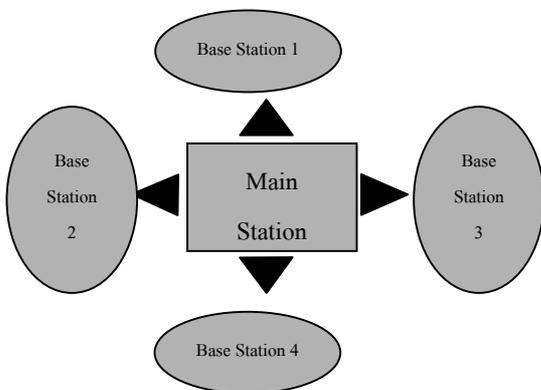


Figure 2. Physical System.

- 1) Director centre (Main Station) and coordinating centres (Base Stations)
- 2) Three types of work units contacting with the inhabitants including parking centre, shelter centre and healthcare centre.
- 3) Specially organised centre for coordinating between other centres and the inhabitants such as flood preventing unit, public relation unit, welfare unit, rescue unit, first aid unit, evacuation unit, security or guarding unit, healthcare unit, aid unit, restoring unit and donation unit.
- 4) Four areas of inhabitants and thirty local communities. (<http://www.hatyaicity.go.th/>)

SIMULATION SYSTEM

From the physical system mentioned above, we modified it by adding and removing some details in order to create a simplified simulation system that can integrate the components together in a better way. Therefore, the simulation model consists of the following.

- 1) Three simulation models including the *control models* that concerns general control of simulated situations at all communities together, the *main station model* or the director centre that acts as the main part for commanding helps and aids for all areas, and *base station model* or coordinating centres for the four areas of inhabitants that helps and aids inhabitants in the area in charge and cooperate with the control model or the main station.
- 2) Three kinds of work units as appeared in the physical system including parking centre, shelter centre and healthcare centre. These are under supervision of its base station.
- 3) Simplified other units including repairing unit, transportation unit, rescue/guarding unit, first aid unit, evacuation unit, aid unit, restoring unit and resource finding unit.
- 4) Four areas of inhabitants and thirty local communities as appeared in the physical system.

The overall of the simulation system is shown in Figure 3. The hierarchy of the simulation models is shown in Figure 4.

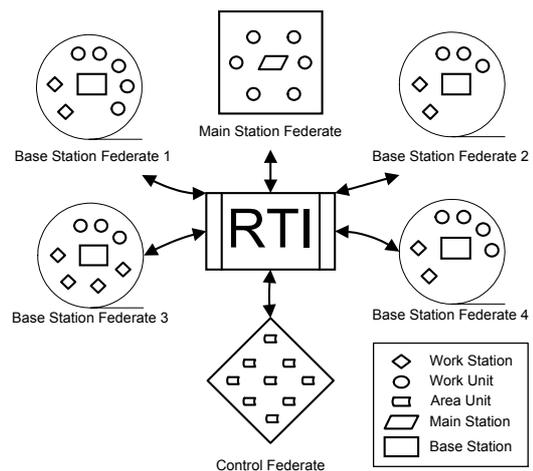


Figure 3. Federation Overview

The control model works as a simulation executive and involves all simulation models. It randomises situations occurring in the simulation as a whole. The controller or the user who works on the control model can add/schedule events at run-time. The main station is also needed in a federation to coordinate works done by the four base stations. However, at a time, there can be at least one base station or more, for more flexibility in increasing or decreasing the number of participating base stations according to the simulation.

Table 1. Interactions in the federation.

	Main Station	Base Station	Work Station	Work Unit	Local Unit
Control Model	S	S	S	S	PS
Main Station Model	PS	S	S	PS	S
Base Station Model	S	PS	PS	PS	S

P: Publication S: Subscription

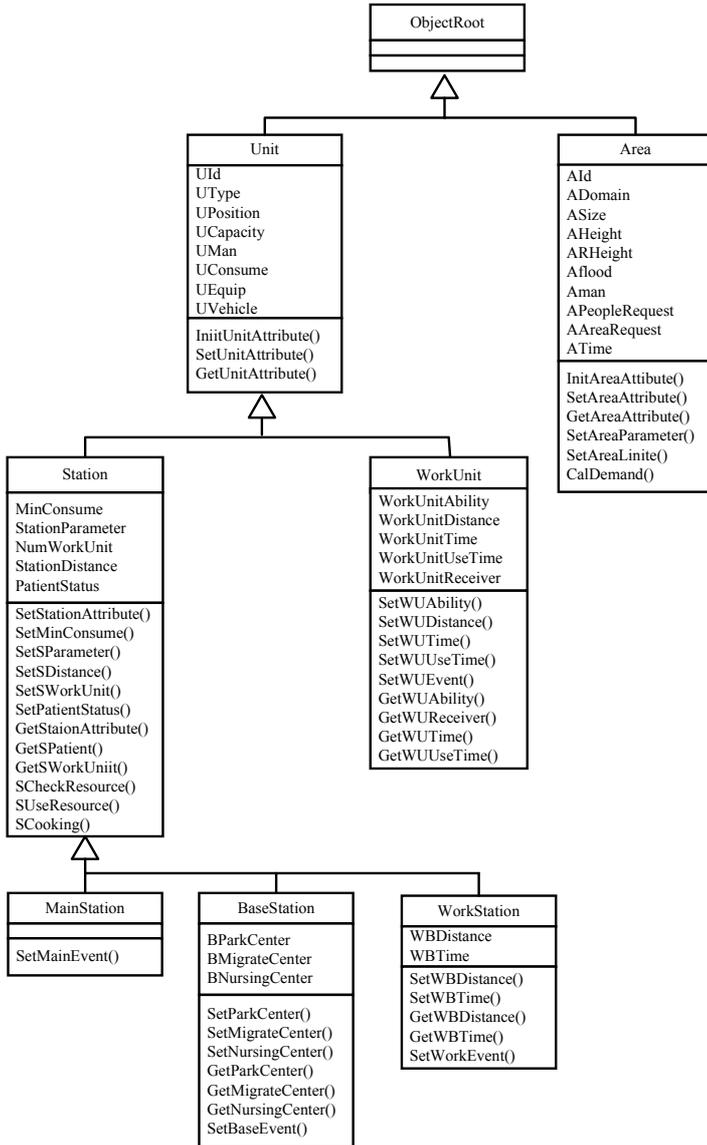


Figure 4. Class Hierarchy

Table 1 shows the interactions in the simulation models in terms of publication and subscription. About exchanging information, the simulation models also have the following relationships.

- 1) The main station directly connects to the control model.
- 2) Each base station connects to the main station.

Therefore, the main station work as an agent or a middle person in connecting each base station to the control model. There is no direct connection from a base station to another base station, as happened in the real organisation, due to authority in distributing work and resources.

RESOURCE MANAGEMENT

Resources exchanged in the simulation can be divided into four different types; human resources or workers, consumable resources, equipment and vehicles. Entities in the consumable group are to be deleted after being used or consumed, and require replacement. The resource management for this group is such that the requiring centre or unit asks for resources from other centres.

1) In case of the stock for the inhabitants, an automatically decreasing stock has been made for simulating real simulations. The stock of consumable resources decreases continuously when it still floods and the inhabitants are stuck in the place. It will soon increase when the level of water decreases that means the inhabitants can collect the resources themselves. From the experience of the Hat Yai municipality, the stuck inhabitants need to live by themselves in order to survive during the first 2 days in case that the life support can not get through the location. This implies that they are required to collect their own life support resources such as food and drug before flooding and after. Equation 1 calculates food decrease at a flooded area.

$$X = (A * AS / PN) * FR \quad (1)$$

When X is collected food

A is a coefficient

AS is the size of the affected area

PN is the number of inhabitants in the area

FR is a food-consuming ratio that has been set of a constant at the begining of flooding simulation and automatically decreases as time advances. When there is no flood anymore, it will then increase.

Note These parameters are adaptable. The food-consuming ratio is added because different areas can have different consuming rates.

2) The decrease of consumable resources in the simulated centres or units is caused by distributing resources to community areas, transferring resources to other centres or units and local consumption. However, the change of resources in the consumable group does not occur in case of dry food for distribution. From

our observation, cooked food has a relationship with raw ingredients and the amount of water used for cooking. To simplify the model, we do not consider seasonal ingredients and oil. Therefore, Equation 2 is used for calculating food transforming.

$$Ax = By + Cz \quad (2)$$

When x is the amount of processed food
 y is the dry food
 z is water
 A, B, C are adaptable coefficients

EMERGENCY SERVICES IN FLOODING SIMULATION

The service simulation is considered in terms of demand and supply. The simulation is driven in two patterns. First, it is stepped by simulation time which is a normal working simulation. Second, it is driven by events happening during the simulation. The first case concerns with all centres and the latter concerns with changes in each community while it is flooding.

Demands for aids

Time advance in the simulation typically sets demands according to the increasing demands of the inhabitants, and leads to more damage at the areas. This can be put into the calculation of local demands in Equation 3.

$$D = D + ((A*People)+(B*Patient))*(C*T) \quad (3)$$

When D is the demand from local people that can be divided into demands of food, dry food, water, drug, life vests and life bags. The specific demands lead to different coefficients in the equation.

A, B, C are adaptable coefficients.

$People$ is the number of inhabitants at the considered area.

$Patient$ is the number of patients at the location.

T is the period that the location has not been visited, calculated from the last time that this location receives help.

Demands of local people increase when time steps. In other words, when time advances and the community has not yet received supplies responded to the demand, the demand will increase, too.

Demands for restoring the area

Flood damage increases exponentially according to the flood height and the height of buildings (Berning et al 2001). Therefore, we

take this conclusion to calculate the demands for restoring the affected area as shown in Equation 4.

$$Z = Z + (X*(AS/PN))*(Y*(FH-AH)) \quad (4)$$

When Z is the demand for restoring the area that can be divided into solving the fuss, restoring building, demolishing ruin, and cleaning.

X, Y are coefficients that are adjustable according to the type of demands

PN is the number of inhabitants in the considered area

AS is the size of the considered area.

FH is the average flood height comparing to the sea level.

AH is the average height of the area considered.

Apart for computing the demands, computing the resource usage time is also another key issue. Some resources are available or to be used within a period of time due to some reasons such as expired date in case of food or availability in case of equipment.

Event driven simulation is adopted in the supplier site. The users need to control the main station and the base stations in order to response to the demands from the affected areas and to manage the resources at the centre and local units involved. If the resources are in sufficient, the users need to find resources from other sources. Demands can be randomed by the simulation or fed by the users at the controller. In addition, when a demand reach a threshold, it can cause some events to happen. The number or amount of resources when reduced below the threshold can trigger some events to happen, too.

LOOK AND FEEL

Situation reports in the simulation are both text and graph based. Figure 5 and 6 show examples of dialog views and simulation reports. In case of text based reports, user can trace the dialog display to check each event occurred during the simulation or the changes to the resources. The tracing commands can be saved into log files. The users can also set current parameters of the simulation.

After finishing the virtual rehearsal, the application will summarise the number of resources left and used, demands, damage, patients and death bodies found at each unit.



Figure 5. Dialog Views during the Simulation

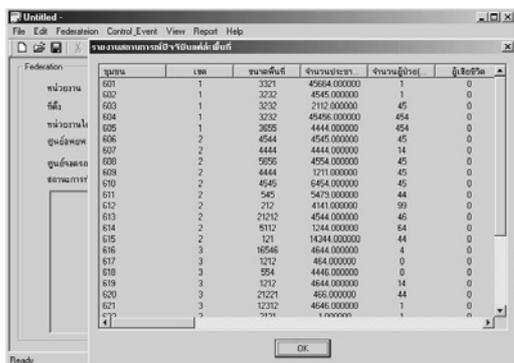
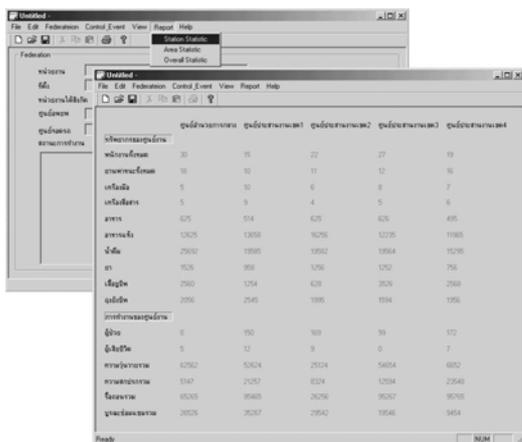


Figure 6. Example of Simulation Reports

CONCLUSION AND FUTURE WORK

This simulation concerns simulating emergency rehearsal in flooding situations in the case study of flooding in Hat Yai municipality, Thailand. The distributed interactive simulation focuses on preparing the work units involved, including staff and interest people, to deal with real situations. The simulation applies the concept of demand and supply. The simulation is simulated into three parts according to the structure of the physical system being simulated. First, the control model acts as a simulation controller that controls the overview of the simulation as a

whole. This model concentrates on demands for serving people and demand for restoring the area. Each community has different demands for resources. Second, simulation model of the director centre called main station and the coordinator centres called base stations. These models concerns suppliers that manage existing resources and find additional resources to serve the demands of their communities. Third, the simulation is driven by both time stepping and events. When a simulations ends, summary of resources usage at each centre and unit, and demands from each community will be made.

Currently, this work is still on progress. More details are still required to serve real organisations.

REFERENCES

Wyne, J., "THE HIGH LEVEL ARCHITECTURE: IS THERE A BETTER WAY?", Winter Simulation Conference Proceedings, 1999, Volume 2, 1999

Berning, C., LA Du Plessis, Viljoen, MF, "Loss function for structural flood mitigation measures", Water SA, Volume 1, No.1, January 2001

Kuhl, F., Weatherly, R., Dahmann, J., "Creating computer simulation systems: introduction to the high level architecture", Prentice-Hall PTR, Prentice - Hall, Inc., 1999

Fujimoto, R.M., "Parallel and distributed simulation systems", A WILEY-INTERSCIENCE PUBLICATION, JOHN WILEY & SONS, Inc., 2000

"IEEE Standard for Modeling and Simulation [M and S] High Level Architecture [HLA] - Federate Interface Specification". IEEE Std 1516.1-2000, 2001

"IEEE standard for Modeling and Simulation (M&S;) high level architecture (HLA)-object model template (OMT) specification", IEEE Std 1516.2-2000, 2001

"IEEE standard for Modeling and Simulation (M&S;) high level architecture (HLA) - framework and rules", IEEE Std. 1516-2000, Sep. 2000

"Hat Yai City Municipality", <http://www.hatyaicity.go.th/>

"Flood Hazard Mapping", http://www.fema.gov/fhm/frm_docs.shtm

"Consumer Advice Disaster Assistance", <http://www.foodsafety.gov/~fsg/fsgdisas.html>

AUTHOR BIOGRAPHIES

CHIRAWAT WATTANAPANICH was born in Trang, Thailand but he grew up in Phattalung. In 2001, he received his BEng in Computer Engineering. Currently, he is pursuing a master's degree and his thesis topic is *Distributed Simulation of an emergency system for the flood disaster in Hat Yai, Thailand*. His personal e-mail address is chiraw@hotmail.com.

PICHAYA TANDAYYA graduated in Electrical Engineering (Communications) from Prince of Songkla University (PSU) in Thailand in 1990. She obtained her PhD in 2001 from the University of Manchester on distributed interactive simulation. She is a lecturer in the Department of Computer Engineering, PSU. Her current research work is concerned with distributed interactive simulation, High Level Architecture, and a Braille keyboard and display unit for the visually-impaired. Her e-mail address is : pichaya@coe.psu.ac.th and her Web-page can be found at <http://fivedots.coe.psu.ac.th/~pichaya>.

Modeling Concepts for the Integrated Reasoning about Complex Systems

Michael Cebulla,
Technische Universität Berlin, Fakultät für Elektrotechnik und Informatik,
Institut für Softwaretechnik und theoretische Informatik,
Franklinstr. 28/29, 10587 Berlin
e-mail:mce@cs.tu-berlin.de

Abstract

In this paper we focus on formal concepts for an integrated modeling of complex systems. We take our examples from the field of sociotechnical systems where we state a great need for model-based reasoning regarding questions of system safety and the definition of long-term management policies. We lay our special focus on systemic capabilities related to context adaptive behavior with special consideration of cognitive parameters. We are convinced that the systemic ability of context adaptivity is insufficiently understood and is of increasing importance in sociotechnical systems as well as in advanced pervasive applications. We think that a model-based approach is well-suited for the improvement of context awareness in complex systems.

Keywords: Requirements Engineering and Specification, Frameworks, Software Architecture.

1 Introduction

Modern systems engineering is more and more confronted with a new quality of contextual embedding. For the specification of systemic behavior a growing number of environmental parameters have to be taken into account. We conceive these parameters as systemic *aspects* and use *ontologies* to achieve a modular way to manage the related knowledge.

Sociotechnical systems as well as advanced applications like ubiquitous services have to reside in complex contexts and to deal with unforeseen situations. In addition they are strongly interwoven with numerous parameters determined by the external environment. One example for these context dependencies is the new impact that human factors have in these kinds of systems. Future ubiquitous services in combination with multimodal interfaces will result in a dependency between user and service whose complexity is unknown up to now. Advanced services will have to possess the capability of *context awareness* in order to adapt their behavior to unexpected situations. Especially they will have to be able to make assumptions concerning the identity of users,

their goals and their knowledge.

This prominent role of context adaptivity is a well known fact in sociotechnical systems. Hybrid teams of humans and devices have to adapt to adverse environmental conditions and have to select those strategies which are optimal in the given situation. Although these systems are of a considerable robustness there is an intensive need for a better understanding of these capabilities which are hard to grasp by formal methods. Due to the complexity of these systems their analysis and understanding is very difficult. But on the other hand for various reasons (safety, efficiency, organizational learning, change management) there is a strong need for a greater transparency of the related processes.

In our approach we suggest a visual notation for the modeling of complex sociotechnical systems as well as context-adaptive systems in general. Starting from software architecture and the experiences of *requirements engineering* [11] we provide concepts which are well suited for the description of specific context adaptive features. Especially the aspects of human cognition and organizational relations are traditionally hard to grasp by formal notations.

We claim that a visual modeling notation significantly increases system transparency, support interdisciplinary system analysis, and is well-suited to support measurements of further education. In addition we argue that domain-specific high level abstractions in general are a strong medium for the design of context aware applications. In the future we plan to provide automated tools for the support of system management.

After some general considerations (section 2) we describe the critical features of sociotechnical systems in section 3. After presenting the basics of our approach (section 4) we introduce some extensions for a cognitive process modeling which is oriented towards the agents capabilities (sections 5 and 6). After demonstrating how to define specific connectors (section 7) we discuss context-adaptive behavior and give some examples for our treatment (section 8).

2 Complexity and Safety

Complex sociotechnical systems have evolved to control high risk technologies by teams of highly qualified specialists. These systems can be defined as *complex* safety-critical systems where *teams* of human operators cooperate with *ensembles* of technical units and devices. Usually, the resulting processes are significantly more complex than in systems consisting solely of technological components because they have to be context aware to a high degree. Examples for this kind of systems are atomic power plants, medical operation theaters and air traffic control.

This new class of system complexity and its related risks have established new requirements for system design and system safety. This is documented by the sad history of catastrophes from Three Miles Island (1979) to Überlingen (2002). The analysis of such complex systems has proven too multi-faceted for the traditional single-disciplinary approach.

A model-based interdisciplinary system analysis is a promising strategy against what Leveson calls *intellectual unmanageability* of high risk systems [6]. The increasing complexity and tight coupling in contemporary high risk systems make a safe and efficient management difficult if not impossible. The main source of failure in complex systems is not human error or an erroneous component, but the *complex interactions* between components which is not understood to a sufficient degree [12]. To increase the level of understanding we choose an model-based approach which is open for results of interdisciplinary research.

In addition we argue that the results of our research and high-level concepts for the various aspects of contextual modeling can be counted as a contribution to an integrated design of very complex systems (i.e. pervasive context-aware services).

3 Sociotechnic Challenges

Of course, the methods of system modeling are not new. Especially in *software engineering* concepts and methods were synthesized to handle the challenges of complexity in development processes. We also heavily rely on the results of systems engineering [14]. The concepts and methods from requirements engineering provide the basic means to manage informal and semiformal knowledge about the target system.

We observed some specific features of sociotechnical systems which can be conceived as challenges for traditional modeling concepts. We claim that the traditional modeling concepts and formal methods as known from software engineering have to be adapted and extended for the specific properties of sociotechnical systems. Speaking generally, just the

merits of formal specification concepts as exactness and well-definedness sometimes prove as shortcomings in the context of sociotechnical systems. It is a generic feature of these processes that they have to deal with vague data, uncertainty and incomplete specifications. Modeling concepts have to adapt to these specific vagueness which can be conceived as important system quality. Paradoxically speaking, too much exactness would lead to less adequate or even wrong specifications.

3.1 Uncertainty

Sociotechnical systems tend to reduce the load of information processing by using vague concepts. So human experts normally don't use exact mathematical numbers but vague expressions from natural languages. We claim that the resulting vagueness is an important precondition for the systems' robustness and safety since it makes their specification less sensitive to contextual changes.

In addition, human actors have to deal with incomplete specifications. In many situation relevant information is not accessible for the human actors. Due to situational time pressure they have to make uncertain decisions based on incomplete information. In our approach we use fuzzy sets and fuzzy logic to specify uncertain information and vague relations [5].

3.2 Adaptive Behavior

One important feature of sociotechnical systems is their *structural dynamism*. The internal structure of systems like the medical operation theatre can be rapidly changing from one phase of the process to another according to environmental changes. For the description of this *structural dynamism* powerful concepts from dynamic architectures [10] are necessary. We handle this problem by introducing *transformation rules* (cf. section 8).

By specifying transformation rules we define the possibilities of configurations to evolve under certain context conditions. Hence, transformation rules describe the adaptive behavior of complex sociotechnical systems in describing the way these systems react to environmental changes by structural mutation.

Sociotechnical systems like the medical operation theatre have distinctive qualities regarding their adaptive capabilities. Thus, their ability to recover in the face of adverse environmental conditions or unexpected events is clearly larger compared to the behavior of traditional component-based systems. Regarding this adaptivity a deeper understanding of sociotechnical processes can contribute to the robustness and flexibility of advanced software-based systems.

3.3 Subjective Evidence

Domain experts tend to disagree about the facts in their field. Sometimes they are not completely sure about the state of affairs. We use fuzzy concepts for the representation of different degrees of subjective evidence or relevance. This enables us to distinguish between different degrees of cognitive adaptation to a given context (section 6).

4 Basic Concepts

We provide concepts for the structural description of complex systems as shown in Figure 1. We provide a visual notation for each of these concepts which allows for an expressive scenario-based modeling. In addition we define a notation for reasoning about systemic features. Our goal is to provide modeling concepts which are meaningful in terms of domain semantics and well-suited for formal reasoning.

For the sake of formal reasoning we base our approach on methods from ontological modeling [7, 3]. Thus, in this section we define a basic ontology for the structural description of complex systems. We are leaving aside all issues which are related to behavioral or temporal specification.

4.1 Agents

We conceive agents as the basic computational elements of every complex systems. Agents interact with each other in order to process requests from their environment.

Definition 1 (Agents) *An agent $\langle name, I, S \rangle$ consists of a symbolic name (denoting his identity), a set of interfaces I and a state S .*

In a description logic example for an agent we first express the hierarchical place of the concept. Then in addition we describe its internal structure as a conjunction of interfaces and features.

```
Anesthesist  $\sqsubseteq$ 
  (AND Agent
    (AND IA1 IA2)
    (AND ( $\leq$  3 acc. $\perp$ )
      ( $\leq$  1 vis. $\perp$ )
      (expertise {low, medium, high})
      (motivation {low, medium, high})))
```

As an example for the visual representation of an agent cf. Figure 5.

4.2 Interfaces

Agents supply their services to their environment by publishing *interfaces*. Using interfaces agents can

be assigned to certain tasks or engage in relations to each other. We describe interfaces as algebraic signatures.

Definition 2 (Agent Signature) *An agent signature is a quadruple $\langle \Sigma, I, O, A \rangle$ where Σ is a data signature $\langle S, Op \rangle$ in the algebraic sense, O is the set of output attributes (the agent variables), I is the set of external attributes that the agent has to read from its environment and A is a set of action names.*

On the semantic level an interface is represented by a conjunction of features.

```
IA1  $\sqsubseteq$  (AND (acc {highPitch, lowPitch})
  (lman {take, give}))
```

In a graph based notation we denote interfaces as *little squares*.

4.3 Linguistic Variables

In this paper we use a simplified version of linguistic variables [5]. Linguistic variables are the essential carriers of data in integrated systems engineering. According to the terminology of description logic they can be conceived as *roles* which are named relations between an agent and a domain.

Thus, as we will demonstrate in examples, it is possible to model the acoustic sense of a human actor by a feature called **acc** which defines a relation between an agent and a domain. We frequently use enumeration types as domains for features.

4.4 Connectors

In complex systems the understanding of agent interactions is a central issue. For the articulation of a system's interaction structure we adopt the concept of *connector* from software architecture [16, 9].

In the context of an ontological approach a connector has to be conceived as a special concept related to interaction.

Definition 3 (Connector) *A connector $\langle name, R \rangle$ consists of a name and a set of roles.*

Similar to agents connectors publish interfaces. They do so to express their requirements concerning services which have to be supplied by the specified agents. This is why we sometimes speak of *roles* when referring to a connectors interfaces.

In our visual notation we use ovals to represent connectors (cf. Figure 8). Concerning a notation which provides better reasoning support connectors are very similar to agents.

4.5 Configurations

Agents constitute *configurations* by interacting using connectors. Connectors express their requirements by publishing interfaces while agents describe their capabilities by publishing interfaces. When a specific selection of an agent’s capabilities satisfies a part of the connectors requirements we say that the concerned interfaces are *matching*.

The typical usage of this concept of matching interfaces can be illustrated by the following example. Since we use connectors for the representation of systemic tasks, interfaces are frequently used for the specification of requirements for the fulfillment of a given task. For each task interface an agent has to be found which supports this interface. The process of this finding usually is called *negotiation* [2]. For the sake of reasoning about matching interfaces we conceive interfaces as concepts in the sense of ontological modeling. We claim that two interfaces match if the concept representing the required interface *subsumes* the provided interface. In terms of model-theoretic semantics we claim that each instance of the provided interface has to be an instance of the required interface. To reason about subsumption *tableau algorithms* can be used which are common in description logics [3].

In our graphical notation we connect matching interfaces using a *link*, which is expressed by a solid line (cf. Figure 8).

4.6 Conceptual Framework

In Figure 1 we show our conceptual framework in a UML-like visual notation. We use the term *conceptual framework* in order to avoid the term *meta-meta-model*.

As we show in Figure 1 we conceive every concept as a subconcept of *Entity* which we express by hollow arrows. Some of the concepts are related by *has-relations* (aggregations) which are expressed by solid arrows. Aggregations may be annotated by cardinality constraints.

5 Availability

Frequently, it is not enough to know that an agent can provide a certain service (as represented by an interface). Moreover, it is necessary to check if the required services are available in the given situation. Consequently we need concepts to reason about an agent’s temporary workload. In order to do this we have to reason about the *actual state* of an agent.

Conceiving the current state of a given agent as a conceptual model we state that the required services which are represented as an additional model are available iff the combination of that model with

the new concept is *consistent*. Of course, with respect to the availability of services the consistency of cardinality constraints (as supported by SHIQ-DL) are of special importance. Again, tableau algorithms can be used to check consistency.

6 Mental Models

In our approach we provide a *cognitive perspective* which takes into account the actors’ subjective motivations and their influence on the global system’s behavior. As we already argued we conceive human factors and cognitive representations as essential contextual features which are crucial for the behavior of sociotechnic systems as well as for advanced pervasive applications. For this sake we use the concept of *mental models*, which describes an actor’s internal representation of his environment.

In our approach a mental model consists of three fuzzy sets (adopting ideas from agent-oriented modeling [17]):

- The agent’s intentions (\mathcal{I}): usually a decision aims at a certain goal. That means that the agent tries to achieve a certain system state by selecting between alternative behavioral options.
- The agent’s beliefs (\mathcal{B}): a decision is highly influenced by the agent’s belief concerning the system’s actual state and its further behavior.
- The agent’s desires (preferences) (\mathcal{D}): usually the agent has a subjective preference for a certain behavioral option which may or may not interfere with the real situation.

We claim that for a proper understanding of the system’s behavior it is necessary to take these factors into account. These factors may influence human behavior to a high degree and thus determine global system behavior. Moreover subjective factors are subject to a large scale of organizational measures like further education and simulator training.

We represent this internal information using fuzzy sets. This gives us the possibility to represent the *subjective relevance* of a proposition by the membership relation μ .

We notate fuzzy sets by using *calligraphic font*. The function of the three fuzzy sets in figure 2 consists in the mental representation of the relevant contextual features [4].

We use a mental model to reason about the adaptation of an actor in a given situation. We define *adaptation* as a relation between a *Context* and an actor. As relevant features of a context we conceive the situational goal (which has to be a strong member of the actors intentions), the behavioral alternatives (which have to be reflected in the actor’s belief),

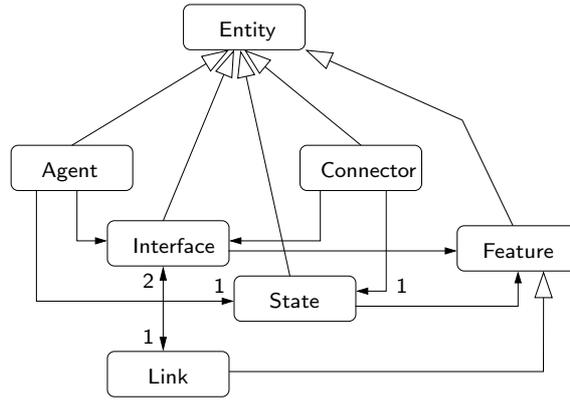


Figure 1: Our Conceptual Framework

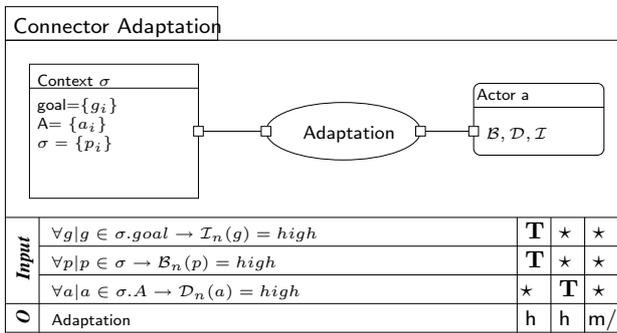


Figure 2: The Concept of Adaptation

and the relevant context conditions (which the actor has to know).

We use fuzzy rules to define a relation between these situational features and actor's mental representations. In general, we claim that an actor's contextual adaptation is good if he has an intensive representation of the relevant features. We use an AND/OR-table to reason about different configurations and their consequences for the quality of adaptation. Thus in the first column of Figure 2 we describe a situation where the actor's adaptation is high because he has strong internal representations of the situational goal and the contextual conditions. In the second row we describe the case that (occasionally) the actor has the adequate preferences. In the rightmost column however we claim that in all other cases the actors adaptation is medium or low (using the \star as wildcard).

For organizational purposes there are some interesting features of contexts. Thus, the cardinality of set A describes the free room of an agent's decision. If it is small this has inhibiting effects on his motivation. If it is too great the agent may be overstrained. Moreover, another important feature of a given context consist in the possibility of differentiation of be-

havioral alternatives.

7 Sociotechnical Connectors

Agents are well-furnished with capabilities to process specific tasks on their own. In complex distributed systems however they have to coordinate their behavior with other agents. Human agents (and in our approach other agents, too) coordinate their activities with each other by the mutual manipulation of their cognitive states. For this sake they rely on interaction and organizational mechanisms of coordination.

We conceive connectors as behavioral specification about the behavior of agents. An important relation between agents is *Communication* (cf. Figure 3). For the definition of this connector again we have to use formulas about the mental models of agents.

We integrate in our specification some claims about the mental models of the actors (as known from speech act theory [15]).

- Perlocution: a result of a communication process is that the message's recipient believes in the content of the message.
- Illocution: for a communication act it is a necessary condition that it belongs to the actor's goals to induce the receiver's conviction that the proposition m holds.

Other parts of the specification we describe the relation between certain parameters of communication (the expertise of the actors) and some global system properties (safety, performance). For example we claim that only highly trained experts can communicate using signals and reach good safety and performance values (cf. column 1 of figure 3).

We conceive organizational relationships are mechanisms that help to compensate weak situational adaptation. While communication is an obvious way

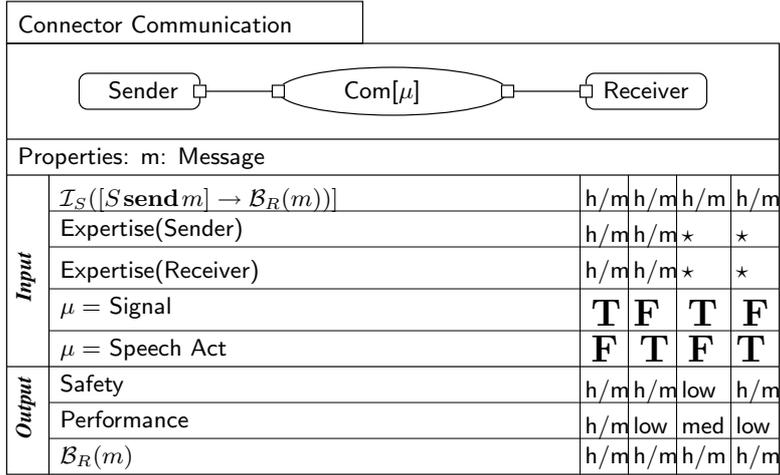


Figure 3: Specification of Connector Communication

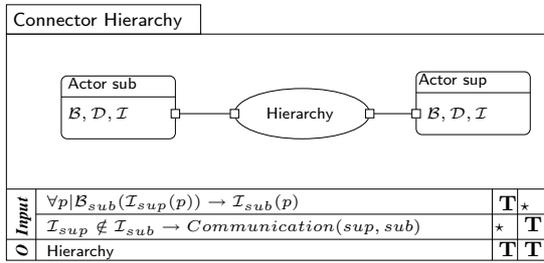


Figure 4: Coordination by hierarchy

for agents to manipulate their mental states there are more subtle mechanisms that have their effect in sociotechnical systems. One of them is hierarchy [8].

As we show in Figure 4 we conceive hierarchy as a mechanisms which asserts that human operators follow the goals of their superiors.

8 Adaptive Behavior

Processes in general as well as sociotechnical processes are structured in *tasks*. Task are logical units of systemic processes. In sociotechnical systems however (as well as in the case of pervasive services) the way a given task is processed may vary in dependence to context conditions. This is a strong adaptive feature of sociotechnical systems: they choose the behavioral alternative which is best regarding the conditions of the actual situations.

For the description of these adaptive features we introduce two levels of abstraction. On the top level we define *abstract tasks* which specify the goals which have to be reached without consideration of context conditions. On the bottom level we define context specific specializations of tasks. In this respect we obviously use ideas taken from *model-driven archi-*

ecture[18].

We use configurations to specify the initial state which is necessary for the processing of a task. Configurations describe the agents which are necessary to process a given task and the way these agents have to interact. We use configuration for the description of abstract as well as of context specific tasks.

To describe the adaptivity of complex behavior we specify *transformation rules* which describe which task specializations are selected in which situations. Triggered by context conditions a certain rule is selected and transforms an abstract configuration into a context specific one. The foundations of this approach are described in [10].

8.1 Example: Coordination

The operators in complex sociotechnical system use technical devices or units to fulfill complex tasks or control complex processes. In order to manage this they have to coordinate their activities. From this point of view coordination is conceived as a compensatory system reaction given adverse environmental conditions.

We separate the aspect of coordination from what we call the *abstract task*, which is defined in terms of the system's primary goal. We discriminate two kinds of coordinative tasks:

- *Transfer tasks* define the need to transfer resources or devices to the place where they are needed. In the operational theatre for example the nurses are heavily involved in passing medical instruments.
- *Communication tasks* define the need for some agents to coordinate their activities by using signs or speech. For instance sometimes the anesthesist has to tell the nurse to pass him an instrument.

In our approach we provide a mechanism to map abstract tasks to context specific tasks. In the example of laryngoscopy implicit coordinative tasks are woven into the specification of the abstract task by a specific transformation rule.

Using this kind of rule-based specification allows us to catch the context-sensitivity of systemic procedures. Which type of procedure is used by the actors in a specific situation is highly dependent on the parameters of the context. One example for such parameters is the *expertise* of the actors: highly trained experts tend to coordinate their activities using signals, while novices normally use convention-based speech acts [15].

In figure 5 we show an example for the description of the coordination aspects of a task. The trigger for the firing of this rule is the precondition that the nurse is the owner of the laryngoscope. This is an adverse context condition since in the specification of the task (not shown in the pictures) it is prescribed that the anesthetist should be the owner of the device.

A special rule describes the selection of interaction type according to the expertise of the actors. The type variable μ is instantiated with type *Signal* if the actors' expertise are high. This rule is very similar to configuration rules in feature modeling [1].

8.2 Example: Observing

In the medical operation theatre during an operation the anesthetist has to control the patient's state and the course of anesthesia. Consequently we define as an abstract task that the anesthetist has to observe the patient. For this sake both, the patient and the anesthetist have to be members of the initial configuration. And they have to be connected by an observation connector.

In Figure 6 for example we use this configuration as a the start configuration of a transformation rule. (For the sake of this presentation we are leaving aside the role of interfaces.) In our first rule we state that the abstract task of observation may be mapped to the context-specific configuration on the right hand side of the rule. In some contexts the anesthetist may delegate the responsibility for the patient's observation to the nurse and communicate with her about the patient's actual state. A precondition for the application of this rule is that the nurse has an adequate qualification.

Another typical way to process the abstract task of patient observation consists in the usage of a monitoring device which controls the characteristic parameters of the patient's state. The corresponding rule is frequently applicated in situations where the patient's state is already stabilized. In many cases anesthetists with low or medium expertise level tend to rely on the monitoring display rather than on their own observations.

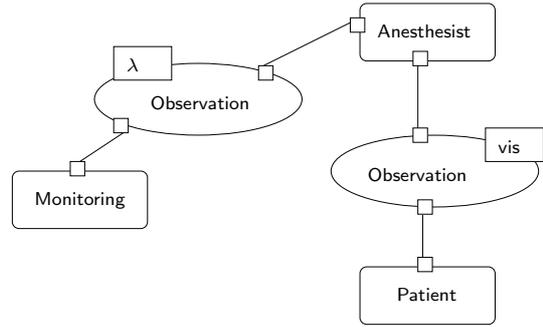


Figure 8: Parametric Configuration

By using simple relations for the sensual capabilities of human agent's we are able to reason about the consequences of task assignments. We are specifying some sensual constraints for human agents.

```
HumanActor ⊆ Agent
HumanActor ⊆ (AND (acc ⊥)
                (lman ⊥)
                (vis ⊥))
```

```
Assignment(x) =
    #(Interfaces.linked ∩ Interface ∋ x)
```

```
Assignment(acc) ≤ 3
Assignment(vis) ≤ 1
Assignment(lman) ≤ 1
```

We represent senso-motoric capabilities as feature variables of agents. The assignment of such a capability is defined by the cardinality of the set of an agent's linked interfaces where the corresponding feature is contained. We claim that for the visual capability of an agent only unary assignments are possible, while the acoustic sense may be used in more than one assignments.

For our example we can conclude that an acoustic observation leaves more flexibility for an actor than a visual engagement. Consequently the configuration in Figure 8 is only legal with $\lambda = acc$. The binding of variable λ with *vis* would violate the assignment constraints. Thus an anesthetist is not able to check another patient when he is occupied with the visual observation of the monitoring device. On the other hand when it is sufficient to control the acoustic alarm tones he can move to another room to assist his colleagues.

9 Human Error

Since we have a special interest for human error we reason about errors using fuzzy fault trees. Using fault trees we can reason about different types of human errors using propositions about mental models.

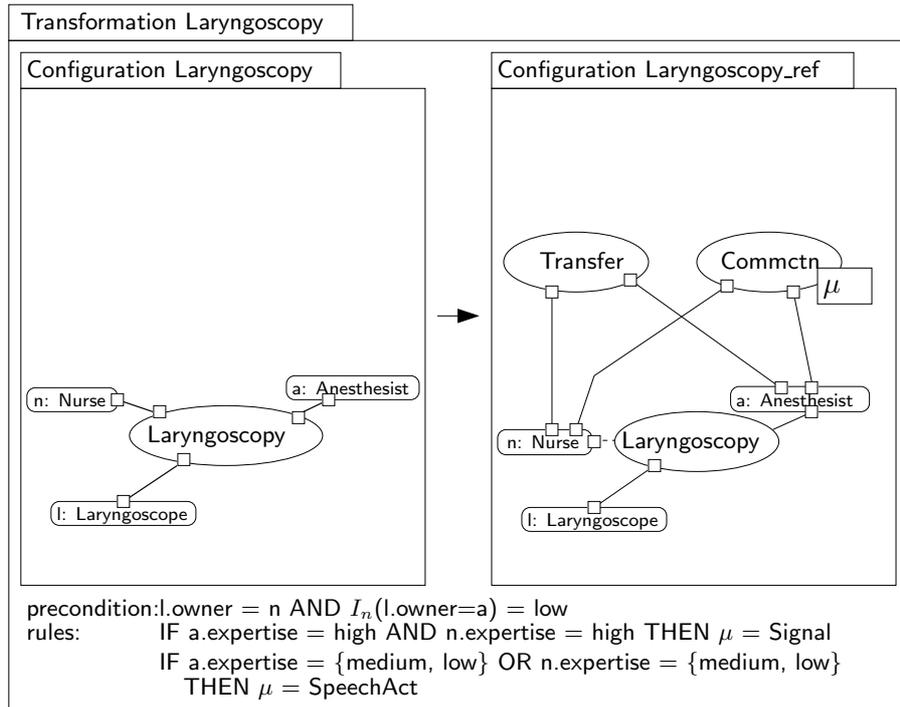


Figure 5: Coordination aspects of Laryngoscopy

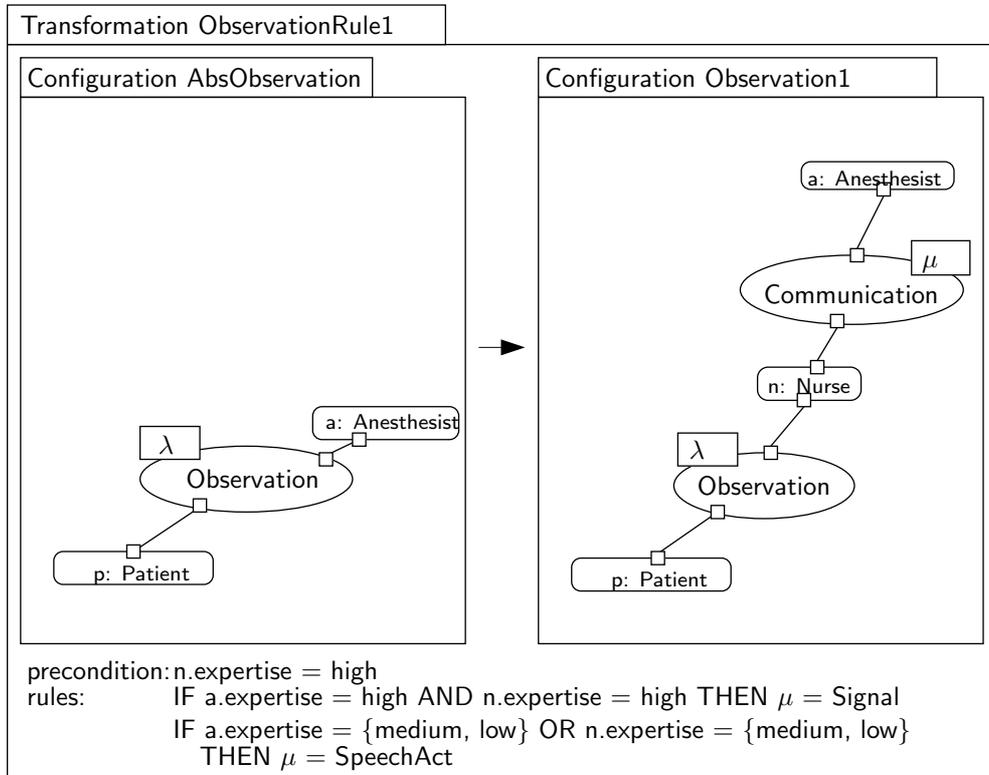


Figure 6: Transformation Rule 1

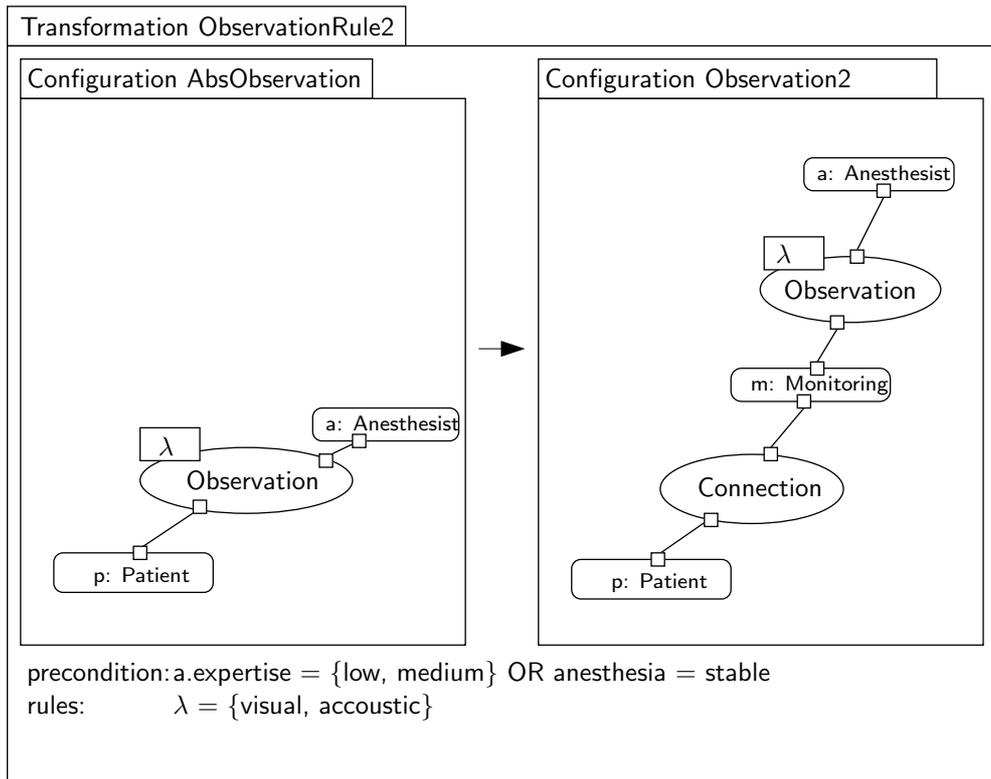


Figure 7: Transformation Rule 2

As shown in figure 9 we distinguish between three cases of human error which are grouped in two sections (following [13]):

- Execution-based errors (slips) occur when the subjective preferences of an actor are not adapted to the context (goals and beliefs may be fitting).
- Knowledge-based errors (mistakes) may occur when the beliefs of an actor are not adapted to the given context. For example a nurse may have the right goals but may not act right because she doesn't see the need to act. Another error-case occurs when an actor's goals do not fit well into the given context.

10 Conclusion

In this paper we provided a visual notation for the context adaptive behavior of complex systems. We started with structural description, dynamic architectures, and rules for reasoning about adaptive system properties. We then took a *cognitive perspective* by introducing the concepts of mental model to reflect cognitive parameters which are highly relevant for systemic behavior.

Using our transformation based approach we employ our notation for the analysis of sociotechnic

adaptivity with special consideration of cognitive context conditions. We claim that the results of this analysis are useful for the design and management of context aware systems in general.

References

- [1] Krzysztof Czarnecki and Ulrich W. Eisenecker. *Generative Programming. Methods, Tools, and Applications*. Addison Wesley, Reading, Massachusetts, 2000.
- [2] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 333–356. Kaufmann, San Mateo, CA, 1988.
- [3] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL, the making of a web ontology language. *Journal of Web Semantics*, 1, 2003.
- [4] Janusz Kacprzyk. *Multistage fuzzy control*. Wiley, Chichester, 1997.
- [5] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall, Upper Saddle River, N.J., 1995.

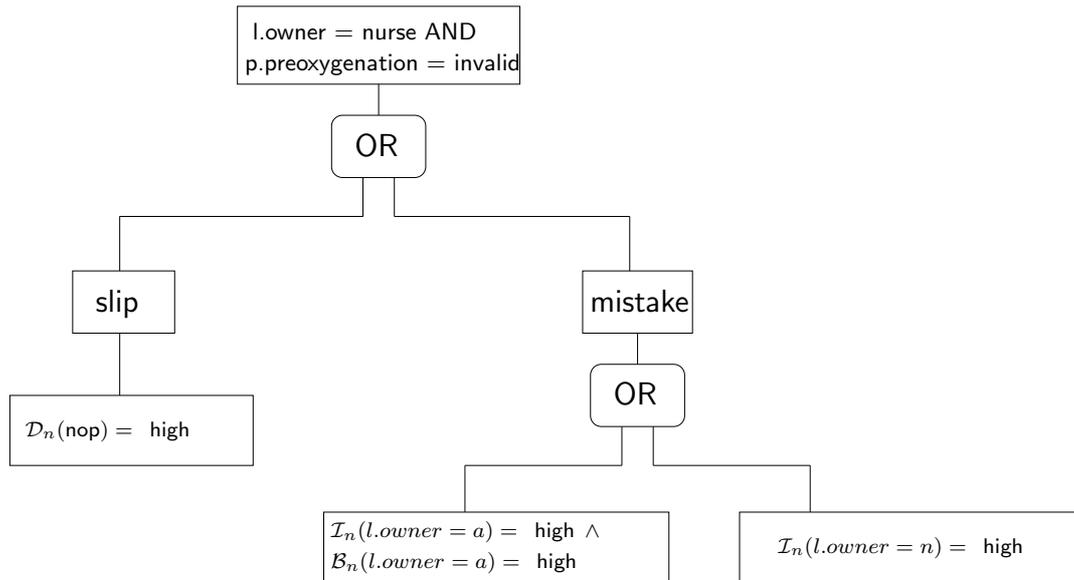


Figure 9: Fault tree for human error

- [6] Nancy Leveson. *Safeware. System safety and computers*. Addison Wesley, Reading, Mass., 1995.
- [7] Deborah L. McGuiness and Frank van Harmelen. OWL Web Ontology Language Overview. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, February 2004.
- [8] Henry Mintzberg. *Structures in fives: designing effective organizations*. Prentice Hall, Englewood Cliffs, 1983.
- [9] Peter Pepper, Michael Cebulla, Klaus Didrich, and Wolfgang Grieskamp. From program languages to software languages. *The Journal of Systems and Software*, 60, 2002.
- [10] Peter Pepper, Ch. Frank, W. Holfelder, D. Jiang, and G. Matylis. Dynamic software architectures for a "sometimes somewhere" telematics concept. Technical report, Technische Universität Berlin, 2003.
- [11] Peter Pepper and Martin Wirsing. A method for the development of correct software. In Manfred Broy and Stefan Jähnichen, editors, *KORSO: Methods, Languages, and Tools for the Construction of Correct Software*, pages 27–57. Springer, 1995.
- [12] Charles Perrow. *Normal Accidents. Living with High-Risk Technologies*. Basic Books, New York, 1984.
- [13] James Reason. *Human Error*. Cambridge Univ. Pr., 1990.
- [14] Andrew P. Sage and William B. Rouse, editors. *Handbook of Systems Engineering and Management*. Wiley, New York, 1999.
- [15] John R. Searle. *Speech Acts*. Cambridge Univ. Pr., 1969.
- [16] Mary Shaw and David Garlan. *Software Architecture. Perspectives on an emerging discipline*. Prentice Hall, Upper Saddle River, N.J., 1996.
- [17] Munindar P. Singh. *Multiagent Systems. A Theoretical Framework for Intentions, Know-how, and Communications*. Springer, Berlin, 1994.
- [18] R. Soley. Model-driven architecture. white paper. <ftp://ftp.omg.org/pub/docs/omg/00-11-05.pdf>, November 2000.

AUTHOR BIOGRAPHIES

MICHAEL CEBULLA was born in Frankfurt/M, Germany and obtained his diploma in Computer Science as well as his Ph. D. in philosophy in Berlin. He worked in industry as a system designer and software developer. His research areas are: systems architecture, modeling concepts for the design and management of complex systems and specification techniques for safety-critical systems. For further information: e-Mail: mce@cs.tu-berlin.de or visit his homepage <http://uebb.cs.tu-berlin.de/~mce>.

USER CLASS BASED QoS DIFFERENTIATION IN 802.11e WLAN

Amit Kejriwal, Ratan Guha
School of Computer Science
University of Central Florida
Orlando, FL-32816 USA
E-mail: {kejriwal, guha}@cs.ucf.edu

Mainak Chatterjee
Dept. of Electrical and Computer Engineering
University of Central Florida
Orlando, FL-32816 USA
mainak@cpe.ucf.edu

KEYWORDS

802.11, 802.11e, QoS, WLAN

ABSTRACT

802.11e is the new extension to the family of 802.11 standards which tries to provide QoS support to data services over the wireless LAN. However, it does not provide any mechanism to differentiate users which becomes a bottleneck for providing differentiated services to users from all sections of the society. In this paper, we study the 802.11e MAC protocol and suggest some enhancements to the current standard that will be able to provide QoS depending on the class a user belongs to in addition to the traffic category used by the user/station. Our algorithm takes into consideration the class of a user and the traffic category before any resource assignment is made. The algorithm has the scope for various tradeoffs that can be provided depending on the importance of the objective function: bandwidth utilization or prioritization of station's ability to transmit. Simulations were conducted to validate our algorithms and test their efficiency.

INTRODUCTION

The need for computing on the move has generated a need for wireless LAN market, with WLANs becoming more and more omni-resent. The mostly commercialized WLAN products available today are based on the IEEE 802.11 standard which has become the most prevailing technology for indoor broadband access for mobile devices. Alongside, the widespread use of networking multimedia applications has brought more requirements to the network and the service providers creating the need for end-to-end quality of service (QoS). The IEEE 802.11 working group is defining a new supplement to the 802.11 MAC called 802.11e which aims at providing toll-quality voice and video services over WLANs [2]. However, 802.11e does not have the option to allow the service providers to differentiate users based on their priority or subscription plans, and hence all users are still treated equally.

Classifying users into different classes and trying to retain them are becoming important strategies for the service providers. Most service providers have started offering differentiated services to users through their pricing plan for voice communication. The users are segmented and offered different values for their services. As the providers start introducing their wireless data services, most likely they will offer similar plans for the data services. In this context, the design of resource control should be based on the mutual benefits of the carriers and users. However, the big question is: what is the quality of wireless data services due to the new impediments created in the wireless networks? Traditionally, today's wireless voice network does not differentiate the QoS among the customers and their voice applications. In other words, the way the wireless network resources are shared between the different voice customers today, does not reflect any bias to the customer's preference or customer's subscription status. The problem lies in the fact that similar design philosophies cannot be used or extended for wireless data networks.

In our opinion, the concept of differentiated quality of service among customers of various classes will be an important parameter in the wireless data networks and services. The objective here is to create different classes of customers based on the selected service packages and provide distinction between the QoS levels like delay or throughput among these classes. In other words, the network must support different customers with different contracts with varying QoS or *service level agreement* (SLA). This motivated the work in this paper.

The rest of the paper is organized as follows. In section 2, we discuss the concept of traffic categories used in 802.11e along with the different timers used for channel access. In section 3, we introduce the concept of user classes and discuss possible scenarios that might arise. Section 4 presents the algorithms to tackle the different scenarios. The simulation is presented in section 5 and conclusions are drawn in the last section.

EDCF: ENHANCED DISTRIBUTIVE COORDINATION FUNCTION

The distributed coordinated function (DCF) is the basis for the 802.11 protocol. An enhanced DCF called EDCF [3] has been proposed for 802.11e which provides QoS based on *traffic categories* (TC) [3]. 802.11e also introduces a concept of QoS-supporting basic service set (QBSS) [3]. QBSS is a BSS that includes the stations that are 802.11e compliant and can support QoS based services. These stations operating in the 802.11e mode are known as enhanced stations.

802.11e allows up to eight traffic categories which are used to differentiate among different traffic and hence priority can be assigned [3]. The traffic category with higher priority will have more probability to access the medium than the traffic category with lower priority [3]. This feature to support traffic priority was missing in the legacy 802.11. Also the DCF inter frame spacing, called DIFS, has been replaced by Arbitration inter frame space (AIFS) in 802.11e [3]. AIFS is different for each traffic category, and depends on the priority of traffic category. The traffic category with the highest priority has the lowest AIFS period and as the priority decreases the AIFS period increases as shown in Figure 1. AIFS value is at least equal to DIFS (i.e., $AIFS \geq DIFS = 34 \mu s$) period for 802.11e to be backward compatible with the legacy 802.11 [2]-[3]. Each traffic category has its own backoff instance which also depends on the priority of the traffic category. EDCF stations waits for a period of AIFS when the medium is idle before starting their backoff function. The contention window (CW) never exceeds the parameter $CW_{max}[TC]$, which is the maximum possible value for CW [2]-[3]. The minimum value depends on the traffic category and can take values between 0 and 255, thus $CW_{min}[TC]=0-255$ [2]-[3].

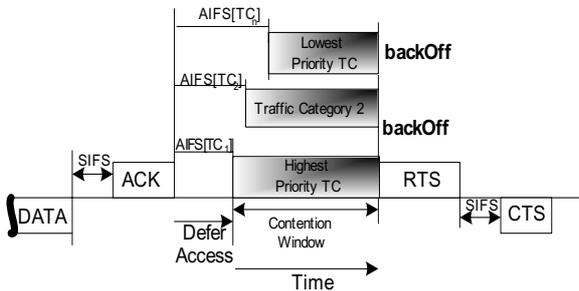


Figure 1: AIFS Classification for Different TC

A station always waits for its AIFS period before it starts transmitting. However, it does not mean that a station can always transmit when the backoff counter reaches zero because the medium can become busy. In that case, the backoff counter is frozen till the next time

medium is free. The station resumes decrementing backoff counter again when the medium become free. If more than one traffic category's backoff counter becomes 0 at the same time, it results in a *virtual collision* [3]. Virtual collisions are avoided by letting the traffic category with the highest priority to transmit. One of the biggest differences between legacy DCF and 802.11e is that in legacy 802.11 the backoff counter is decremented by 1 after the first time slot of DIFS period, while in 802.11e the backoff counter is decremented by 1 from the last time slot of AIFS period [3]. This makes a big difference in the time when a node chooses a backoff instance. Legacy 802.11 increases the range of backoff value by twice after each unsuccessful transmission but in 802.11e the new backoff value is calculated using a parameter called PF (Persistence Factor) which takes different values (1 through 16) for different TC. Thus [3],

$$newCW[TC] = ((oldCW[TC] + 1) \times PF) - 1.$$

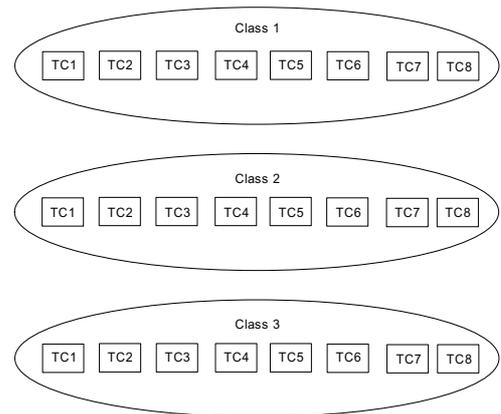


Figure 2: User Class and Traffic Categories

CLASS BASED QoS

802.11e does provide QoS by introducing traffic categories. Since all the traffic categories can potentially be used by every users/stations, there is no scope to differentiate among users. Let us assume that there are multiple classes of users all using the eight traffic categories. Figure 2 shows only 3 classes, namely Class 1, Class 2 and Class 3. We propose that the user classes be taking into consideration while deciding the time for a station to transmit. Normally, the total time that each station has to wait before trying to transmit is AIFS plus the contention window. We suggest that both the AIFS and the CW be made dependent on the class of the user and traffic category. Thus, for a station i to transmit, the time to transmit (TT_i) is given by

$$TT_i = AIFS [Class_i] [TC_i] + CW [Class_i] [TC_i].$$

Possible Scenarios

Before we propose our algorithms, let us introduce the possible scenarios that might arise due to the introduction of these multiple classes. The algorithm used to calculate the time for a station to attempt a transmission, must allow different variations in the priority based on class and the data traffic category used by the station. We consider that Class 1 has the highest priority and Class N has the lowest. In other words, the priority schedule is as follows:

$$\text{Class 1} > \text{Class 2} > \text{Class 3} > \dots > \text{Class N}$$

Scenario 1: This is a trivial scenario where though there are multiple classes; there is no priority differentiation amongst the classes. The priority of a station to transmit is only differentiated by using the traffic category, which is essentially the same as supported by the current 802.11e. Figure 3 shows how the different classes are treated at par.

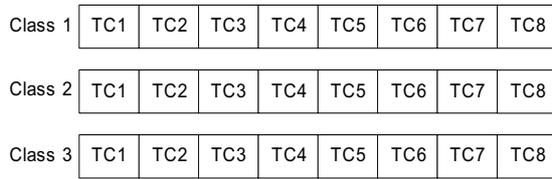


Figure 3: SCENARIO 1

Scenario 2: In this scenario, a higher class has the priority to transmit irrespective of the traffic category as shown in Figure 4.

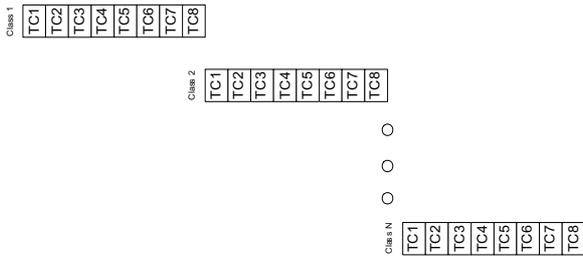


Figure 4: SCENARIO 2

Scenario 3 (General Scenario): In this scenario, only the corresponding traffic category with higher class has a priority over corresponding traffic category in lower class. However, there is a possibility that the priority difference to vary. For example the priority of TC₁ and TC₂ in class 1 may have priority over the TC₁ and TC₂ of class 2 or priority of TC₁, TC₂ and TC₃ in class 1 has priority over the TC₁, TC₂ and TC₃ of class 2 and this relationship continue throughout all the classes and

traffic categories. Figure 5 shows this difference as Δ_j . Δ_j represents the number of traffic categories for a class that have priority over corresponding traffic categories in lower priority class.

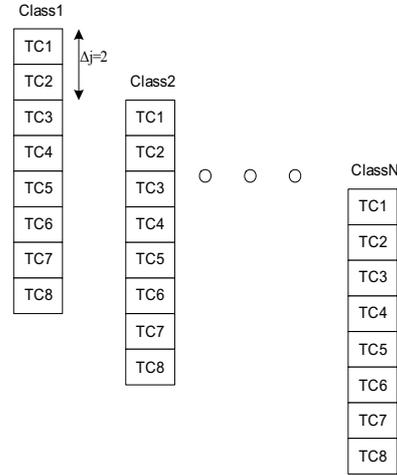


Figure 5: SCENARIO 3, WHEN $\Delta_j = 2$

PROPOSED ALGORITHMS

We first deal with the first two scenarios before we discuss the general scenario. The generalized algorithm for scenario 3 will encompass all the other scenarios just by changing the value of Δ_j .

Algorithm for Scenario 1

This scenario is similar to the one supported by the present 802.11e. In this scenario all users have same priority (only one class exists) but the difference in priority is among the traffic categories [3]. This case can be supported by allowing different AIFS and CW for different TC's priority. If TC₁ has highest priority then it can be scheduled as:

$$\text{AIFS}[\text{TC}_i] < \text{AIFS}[\text{TC}_j] \text{ if } i < j$$

As the CW_{max} is constant for a particular TC, therefore the mean value for the CW will be less for a traffic category with higher priority [3]. Therefore, time to transmit will be

$$\text{TT}_i = \text{AIFS}[\text{TC}_i] + \text{CW}[\text{TC}_i]$$

Thus from the relationship between $\text{AIFS}[\text{TC}_i]$ and $\text{CW}[\text{TC}_i]$ we can say that

$$\text{TT}[\text{TC}_i] < \text{TT}[\text{TC}_j] \text{ if } i < j.$$

Algorithm for Scenario 2

In this scenario all traffic categories of a particular class will have more priority over traffic categories of all other classes of lower priority than that class. To achieve this relationship, first it is important to find out the time each station has to wait to transmit data depending on its class and traffic category.

$$TT_i = AIFS_i [\text{Class}] + CW_i [\text{TC}]$$

In this scenario, AIFS of the station is dependent on the class and CW is dependent on traffic category, thus time to transmit is dependent on both class and traffic category. CW [TC] can be found as it is suggested in 802.11e. To achieve the priority as described above, the relationship between i^{th} and $i+1^{\text{th}}$ class is:

$$AIFS_{i+1} - AIFS_i > CW[\text{TC}_n]$$

$CW[\text{TC}_n]$ is the maximum value of contention window produced by all the traffic categories. Thus, *all* traffic categories of a higher priority class will have more priority than all the traffic categories of a lower priority class.

Let P and Q represent the class of a user and the traffic category of the station respectively. Highest priority implies $P=0$ and highest traffic category implies $Q=0$. As priority decreases for both class and traffic category the value of P and Q increases by 1. Let us denote the time to transmit for a user belonging to class P and with traffic category Q by $TT [P] [Q]$. This scenario will maintain the following relationships:

$$TT[P][Q] < TT[P][Q+1]$$

$$TT[P][Q] < TT[P+1][Q]$$

The value for Q is n for $TT[P][Q]$ and 0 for $TT[P+1][Q]$. In spite of these priorities, there is no way to eliminate the possibility of collisions, as this algorithm is basically collision avoidance and not collision elimination. So the algorithm should also include the penalty for unsuccessful transmission. When the number of unsuccessful transmissions reaches certain threshold, time to transmit by a station shall be increased as a penalty. Since this is a class based service, even after unsuccessful transmission, it may be preferable to have fewer penalties for a station with higher priority. The downside of having lower penalties for station with higher priority is that it might delay the transmission of data or starve the lower priority stations. So to avoid the delay, we do not penalize a station with higher priority until a certain number of retransmissions, say $\alpha_{\text{Threshold}}$. Thus, if the number of retransmissions is below $\alpha_{\text{Threshold}}$ the value of time to transmit remains unaffected. If the number of retransmission is higher than $\alpha_{\text{Threshold}}$ the time to transmit is increased by an

Increment Factor (IF). Increment factor depends on whether the algorithm needs to be fair to all users or it needs to be favorable to the stations with higher priority, and is always greater than 1. Therefore the algorithm can be defined as follows.

$$\text{if (no.of retransmission} < \alpha_{\text{Threshold}})$$

$$TT_i = AIFS_{\text{class}} + CW_{\text{Trafficcategory}}$$

else

$$TT_i = IF \times (AIFS_{\text{class}} + CW_{\text{Trafficcategory}})$$

Algorithm for Scenario 3 (General Scenario)

With the special scenarios discussed, we are in position to discuss the algorithm which will handle any scenario. Both classes and traffic category are interdependent, while deciding time to transmit. As use the same definition for P and Q . The time to transmit for user from class P having traffic category Q is represented by $TT[P][Q]$. The general relationships can be described as:

$$TT[P][Q] < TT[P][Q+1]$$

$$TT[P][Q] < TT[P+1][Q]$$

$$TT[P][Q] < TT[P-1][Q+1]$$

Recall, Δ_j represents the number of traffic categories for a class that have priority over corresponding traffic categories in the next lower class. In general, Δ_j can take values from 1 through n , where n is the number of traffic categories. For example, when $\Delta_j=2$, the 2 consecutive traffic categories have priority over the same traffic categories in the next lower class. While the same two traffic categories of that lower priority class have higher priority over same traffic categories for the classes that have lower priority than itself. In general form of equation in the form of Δ_j the relationship can be expressed as:

$$TT[P][Q + \Delta_j - 1] < TT[P+1][Q]$$

$$TT[P][Q] < TT[P-1][Q + \Delta_j]$$

$$TT[P][Q + \Delta_j - 1] < TT[P-1][Q + \Delta_j]$$

The above inequalities explain the Δ_j relationship with the classes and traffic category. Let us now calculate the contention window. If Q is the priority value of traffic category, r is the priority value of class, and $D_j = \Delta_j$. Initially the value of D is equal to D_j , D is reset to D_j once it reaches 0, p keeps track of TC , i represent the class category for which the algorithm is calculating CW_{max} and j represents the traffic category for which function is calculating CW_{max} .

$$\text{sum} \leftarrow i + (r+1) \times ((j-1)/Dj) + ((j-1)\%Dj)$$

if (D = 0) {

$$p \leftarrow p+1$$

$$D \leftarrow Dj$$

}

$$D \leftarrow D-1$$

$$CW'_{\max} \leftarrow \text{sum} + r \times p \times Dj \times MF + i$$

if(i!=1)

$$CW'_{\max} \leftarrow \text{sum} - i + i \times (Dj-1)$$

MF is the *Main Factor*, where $MF > 1$, defined as the difference in CW'_{\max} value among the traffic categories which are Δj apart within the same class. Greater value of MF will increase the CW'_{\max} value. The value of MF is a tradeoff between the bandwidth utility and providing higher probability to transmit to those stations that have higher priority. The higher value of MF will make sure that station with higher priority may have more probability to access medium but its bandwidth utilization may be very bad if the higher priority stations are not transmitting often. This will happen because even though the medium is free the lower priority stations have to wait for longer time before they can try to transmit as their time to transmit will be large. Value of CW'_{\max} depends on class and traffic category and is related to CW'_{\max} as

$$CW_{\max} = RF \times CW'_{\max}$$

where RF is reducing factor to reduce the value of CW_{\max} as sometimes the function may produce high values depending on the trade-off. Thus,

$$TT[P][Q] = AIFS + CW_{\max}[P][Q]$$

AIFS can be kept at minimum in this case and that is equal to $DIFS=34 \mu s$. Value of AIFS will remain same for all values of class and traffic category. The algorithm will produce different values of CW_{\max} depending on class, traffic category and Δj . The value of CW_{\max} produced by above function are as desired by the algorithm. Thus $TT[P][Q]$ relation will be stored as desired by class, traffic category and Δj .

SIMULATION MODEL AND RESULTS

We simulated our proposed algorithm with varying parameters. We considered 5 classes of users, 8 traffic categories, $MF=1$, $RF=1$ and $\Delta j=2$. From Figure 6 we see that the desired relationship is maintained by the function. Since $\Delta j=2$, the value of CW_{\max} for class 1 traffic category 1 & class 1 traffic category 2 should be less than Class 2 traffic category 1 & class 2 traffic

category. Also the value of CW_{\max} for class 1 traffic category 3 should be greater than class 2 traffic category 1 & class 2 traffic category 1 & class 2 traffic category 2.

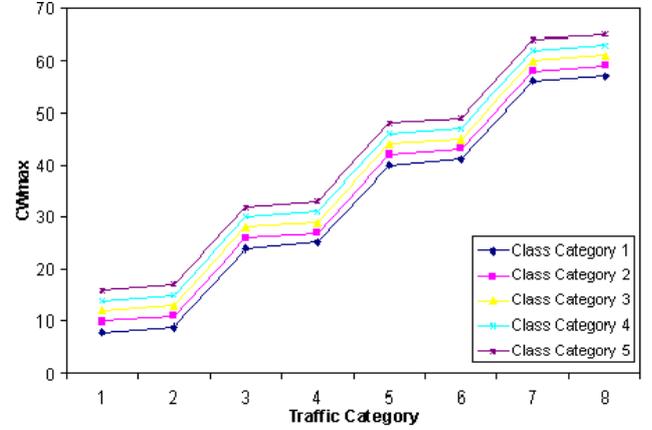


Figure 6: VALUES OF CW_{\max} ; $\Delta j=2$, $MF=1$

Also, in case of $\Delta j=3$ the required relationship is maintained as can be seen from Figure 7. Value of some class and traffic category is chosen to be near each other as to avoid bandwidth loss if there is no transmission from a high priority station.

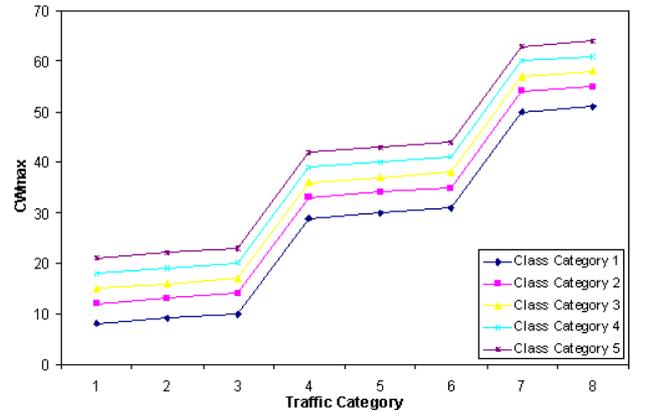


Figure 7: VALUES OF CW_{\max} ; $\Delta j=3$, $MF=1$

The effect of a different value of MF can be seen in Figure 8, where MF was set to 2. The values of class 1 traffic category 3 & class 1 traffic category 4 were 24 and 25 respectively in case of $MF=1$. But for $MF=2$ these values were 34 and 35 respectively. This shows that increase in MF will increase the value of CW_{\max} value for every class and traffic category across the system. Using above shown algorithm the relationship as desired in Scenario 2 can be achieved by putting $\Delta j=8$.

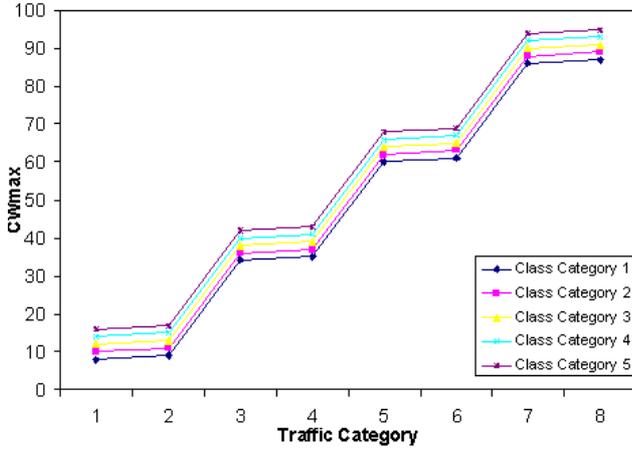


Figure 8: VALUES OF CW_{max}; Δj=2, MF=2

Mean of CW_{max}

The value of CW_{max} maintains the relationship as desired. Since CW_{max} gives the maximum value, a station can choose between 0 and CW_{max}. The mean for uniform distribution will be CW_{max}/2 for all the different value of CW_{max} depending on class and traffic category. In some cases there is a possibility that the mean may be too close for some values, in that case the probability of collision might increase. To reduce this collision probability instead of choosing random values between 0 and CW_{max}, a new function can be defined as a function of the user class, which will choose a random value between X and CW_{max}. X will be dependent on the class of the station. In that case the mean for uniform distribution will be (X+CW_{max})/2. X can be formally defined as

$$X = (P - 1) \times \Delta j$$

This way the mean value of CW can be changed.

Starvation

Algorithm for calculating CW_{max} may initially give an impression that it might produce starvation for the stations that have lower class and traffic priority. But in reality this is not the case since the algorithm is distributive and once a station chooses a value for CW_{max}, it does not reset CW_{max} in any case except collision. A new value is chosen for CW_{max} in case of a collision irrespective of station's priority. At any given instance of time a station with a higher priority will be given lower value of CW_{max} than lower priority station but once the value is chosen for any station it is not reset even if a higher priority station tries to claim the medium at some later instance of time. The value TT_i of lower priority station decreases every time it found the medium to be idle for more than AIFS period and thus

once its back off counter reaches 0 it will be allowed to transmit irrespective of whether a station with higher priority is waiting to transmit and this simple but distributive approach will save stations with lower priorities from starvation.

Retransmissions

802.11e suggests that after an unsuccessful transmission, a new contention window using persistence factor (PF) be calculated. It is suggested that the value of CW be increased as follows:

$$\text{newCW}[\text{TC}] \geq ((\text{oldCW}[\text{TC}] + 1) * \text{PF}) - 1$$

In the solution presented for the class based QoS instead of increasing the value of the contention window after each unsuccessful transmission, we suggest that the value of CW is increased only after a certain number ($\alpha_{\text{Threshold}}$) of retransmission is reached. This way a station will not be penalized after each unsuccessful retransmission but will be penalized only after it consistently results in unsuccessful retransmissions. Also the value of the Persistence Factor will depend both on the class and the traffic category. To make sure that a station which has many consecutive unsuccessful retransmissions failure is not penalized heavily, we propose another variable, α_{max} , which counts the number of times the value of CW has changed consecutively before making a successful transmission. Once that reaches a certain threshold value, $\alpha_{\text{maxthreshold}}$, the value of CW is not increased any more but kept at the last increased value which makes sure that no station suffers from starvation.

if (no. of retransmission < $\alpha_{\text{Threshold}}$)

$$\text{newCW} = \text{OldCW}$$

if (no. of retransmission $\geq \alpha_{\text{Threshold}}$)

and if ($\alpha_{\text{max}} < \alpha_{\text{maxthreshold}}$)

$$\text{newCW}[\text{TC}] \geq ((\text{oldCW}[\text{TC}] + 1) \times \text{PF}) - 1$$

else

$$\text{newCW} = \text{OldCW}$$

The added benefit of using α_{max} and $\alpha_{\text{maxthreshold}}$ is that these values can also be user class and traffic category dependent and thus different values can be assigned to different stations depending on the class and data's traffic category. If the values of PF change depending only on the traffic category the slope of the graph for each class will almost remain same as shown in Figure 10.

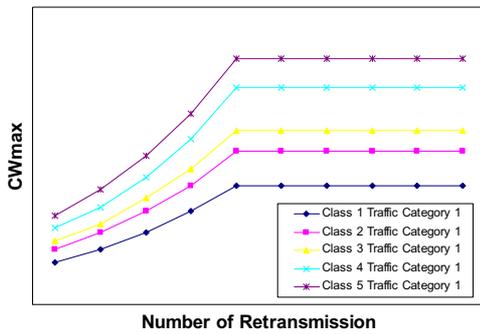


Figure 9: CW_{MAX} : PF IS DEPENDENT ON TC

If the value of PF is dependent on class and traffic category, the value of CW_{max} will have greater slope for lower priority class compared to the higher priority class as shown in Figure 10.

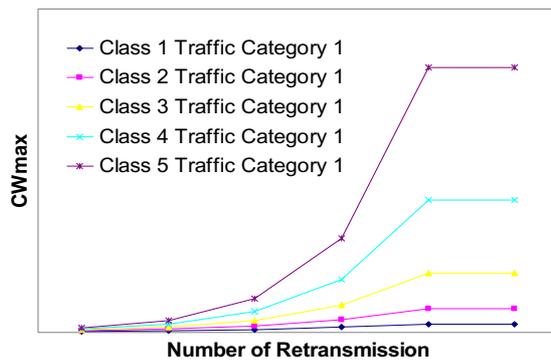


Figure 10: CW_{MAX} : PF IS DEPENDENT ON CLASS AND TC

CONCLUSIONS

In this paper, we proposed algorithms for class based QoS for 802.11e based WLAN services. Different aspects of algorithm can be tuned by controlling parameters like MF and RF to achieve the desired level of system performance. We considered different scenarios including the generalized one. Algorithms are distributive, simple and give flexibility for trade-offs depending on the demand for bandwidth or increasing probability to transmit for higher priority class. The benefit of using above suggested algorithm is that it provides alternate and more flexible way of providing QoS. Simulation results demonstrate that our proposed algorithms do provide the differentiation among different user classes.

ACKNOWLEDGEMENT

This work was partially supported by ARO under grant DAAD19-01-1-0502 and NSF under Grant EIA 0086251.

REFERENCES

- H.L. Truong and G. Vannuccini, "The IEEE 802.11e MAC for Quality of Service in Wireless LANs" in Proceedings of SSGRR2003w, Telecom Italia Learning Services International Conference on advances on the Infrastructure in e-business, e-science and mobile networking, L'Aquila, January 8th 2003.
- Habetha, J. and Mangold, S. and Wiegert, J., "802.11 versus HiperLAN/2 - a comparison of decentralized and centralized MAC protocols for multihop ad hoc radio networks," in *Proc. 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI)*, Orlando, USA, July 2001, pp. 206-211.
- IEEE 802.11 WG, Draft Supplement to STANDARD FOR Telecommunications and Information Exchange between Systems -LAN/MAN Specific Requirements -Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS).
- S. Mangold, S. Choi, P. May, G. Hiertz "IEEE 802.11e - Fair Resource Sharing between Overlapping Basic Service Sets" in proc. of the PIMRC 2002, Lisbon, Portugal, September 2002, pp. 166-171.
- D. Qiao and S. Choi "Goodput enhancement of IEEE 802.11a wireless LAN via Link adaptation" in proc. IEEE ICC 2001, Helsinki, Finland, June 2001, vol. 7, pp. 1995-2000.
- Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, Lothar Stibor "IEEE 802.11e Wireless LAN for Quality of Service" in proc. of the European Wireless, Vol. 1, Florence, Italy, February 2002, pp. 32-39.

AUTHOR BIOGRAPHIES

Amit Kejriwal is pursuing his PhD in computer science at University of Central Florida and is a Sr. Software Developer working on Integrated Criminal Justice System. His interests include ad-hoc networks, network security and wireless networks.

Ratan Guha is a Professor in the School of Computer Science at the University of Central Florida. His research interests include modeling and simulation, distributed computing and networking, and network security.

Mainak Chatterjee is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Central Florida. His research interests include resource management and quality-of-service provisioning in wireless and cellular networks, sensor networks, CDMA data networking media access control protocols, Internet traffic, and applied game theory.

INTERACTIVE SIMULATION OF OBJECT-ORIENTED HYBRID MODELS, BY COMBINED USE OF EJS, MATLAB/SIMULINK AND MODELICA/DYMOLA

Carla Martin
Alfonso Urquia
Jose Sanchez
Sebastian Dormido
Dept. Informática y Automática
E.T.S. Ingeniería Informática, UNED
Juan del Rosal 16, 28040 Madrid, Spain
{carla,aurquia,jsanchez,sdormido}@dia.uned.es

Francisco Esquembre
Dept. Matemáticas
Universidad de Murcia
Campus de Espinardo,
30071 Murcia, Spain
fem@um.es

Jose Luis Guzman
Manuel Berenguel
Dept. Lenguajes y Computación
Universidad de Almería
Ctra. Sacramento s/n,
04120 Almería, Spain
{joguzman,beren}@ual.es

KEYWORDS

Interactive simulation, web-based simulation, object-oriented modeling, hybrid models, quadruple-tank process, Easy Java Simulations.

ABSTRACT

Easy Java Simulations (Ejs) is a freeware, open source, Java-based tool intended to create interactive dynamic simulations. The use of Ejs, together with Matlab/Simulink and Modelica/Dymola allow us to combine the best features of each tool. Ejs capability for building interactive user-interfaces composed of graphical elements, whose properties are linked to the model variables. Matlab/Simulink capability for modeling of automatic control systems and for model analysis. Modelica capability for physical modeling, and finally Dymola capability for simulating hybrid-DAE models. The combined application of these tools to the implementation of interactive simulations is discussed in this manuscript, and a novel modeling methodology adequate for interactive simulation is proposed. It takes advantage of the modeling and simulation capabilities of Modelica and Dymola. The proposed methodology is successfully applied to a case study: the interactive simulation of the quadruple-tank process.

INTRODUCTION

Interactive simulation provides a flexible and user-friendly method to define the experiments performed on the model. During the interactive simulation run, the user can change the value of the model inputs, parameters and state variables, perceiving instantly how these changes affect to the model dynamic. As a consequence, interactive simulation facilitates developing and enhancing the understanding of the model behavior. This capability is especially useful when the model is being used for educational purposes.

Easy Java Simulations (Ejs) is a freeware, open source, Java-based tool intended to create interactive dynamic simulations (Esquembre, 2004; <http://fem.um.es/Ejs/>). Ejs was originally designed to be used by students for

interactive learning, under the supervision of educators with a low programming level. As a consequence, simplicity was a requirement. Ejs guides the user in the process of creating interactive simulations. This process includes the definition of the *model* and the *view*.

Ejs implements its own procedure to define the *model*: a simple structure that the user needs to complete in order to specify the model variables and equations. In addition, Ejs version 3.2 supports the option of describing and simulating the model using Matlab/Simulink: (1) Matlab code and calls to any Matlab function (either built-in or defined in an M-file) can be used at any point in the Ejs model; and (2) the Ejs model can be partially or completely developed using Simulink block diagrams.

The *view* is the user-to-model interface of Ejs interactive simulations. It is intended to: (1) provide a visual representation of the model relevant properties and dynamic behavior; and (2) facilitate the user's interactive actions on the model. Ejs includes a set of ready-to-use visual elements, that the modeler can use to compose a sophisticated view in a simple, drag-and-drop way. The properties of the view elements can be linked to the model variables, producing a bi-directional flow of information between the view and the model. Any change of a model variable value is automatically displayed by the view. Reciprocally, any user interaction with the view automatically modifies the value of the corresponding model variable.

Once the modeler has defined the model and the view of the interactive simulation, Ejs generates the Java source code of the simulation program, compiles the program, packs the resulting object files into a compressed file, and generates HTML pages containing the narrative and the simulation as an applet. The user can readily run the simulation and/or publish it on the Internet.

Working together with Matlab/Simulink significantly improves the Ejs capabilities for model description and numerical solution. However, Simulink modeling paradigm (i.e., graphical block-diagram modeling)

exhibits some limitations (Astrom et al. 1998). It requires explicit state models (ODE) and that the blocks have a unidirectional data flow from inputs to outputs. These restrictions strongly condition the modeling task, which requires a considerable effort from the modeller.

Object-oriented modeling languages facilitate the physical modeling paradigm (Astrom et al. 1998), an attractive alternative to the block diagram modeling. They support a declarative (non-causal) description of the model, which permits better reuse of the models. The modeling knowledge is represented in terms of hybrid-DAE models: differential, algebraic and discrete equations that may change by being triggered by events. As a consequence, the use of object-oriented modeling languages reduces considerably the modeling effort. In order to achieve this goal, the use of Modelica/Dymola together with Ejs and Matlab/Simulink is proposed in this manuscript, in addition to a novel modeling methodology intended for interactive simulation. This methodology is successfully applied to a case study: the interactive simulation of the quadruple-tank process.

COMBINED USE OF EJS, MATLAB/SIMULINK AND MODELICA/DYMOLA

Dymola 5.0 interface to Matlab 5.3 / Simulink 3.0 can be found in Simulink's library browser: DymolaBlock block (Dynasim 2002). This block is an interface to the C-code generated by Dymola for the Modelica model. This C-code contains the numerical algorithms required to simulate the Modelica model, which is formulated in terms of differential-algebraic equations (DAE) and discrete-time equations. DymolaBlock block solution is synchronized with the solution of the other blocks: Simulink integrates DymolaBlock blocks together with the other blocks.

Double-clicking on the DymolaBlock block opens a form where the name and file of the Modelica model name are specified. In order to embed the Modelica model within a Simulink block, the computational causality of the Modelica model interface needs to be explicitly set. The input variables are supposed to be calculated from other blocks, while the output variables are calculated from the Modelica model. DymolaBlock block can be connected to other Simulink blocks and also to other DymolaBlock blocks.

Ejs 3.2 supports the option of describing and simulating some parts of the model (or the complete model) using Matlab/Simulink. Ejs synchronizes the numerical solution of the model part described using Ejs procedure and the part described in terms of Simulink blocks. In order to simulate the Ejs part of the model, Ejs implements a set of built-in ODE solvers, and it allows the modeler to program and use his own numerical algorithms. Matlab code is calculated in the Matlab workspace. Simulink part of the model is simulated by Matlab/Simulink, using Simulink's numerical algorithms.

MODELING FOR INTERACTIVE SIMULATION USING MODELICA LANGUAGE

A novel modeling methodology intended for interactive simulation is proposed. It takes advantage of the modeling and simulation capabilities of Modelica and Dymola. The proposed methodology takes into account the common characteristics of the models intended for interactive simulation. As a result of the user interaction, these models need to support instantaneous (discontinuous) changes in the state, and in the value of the input variables and parameters. In addition, several choices of the state variables need to be simultaneously supported, in order to provide alternative ways of describing the state changes.

The model shown in Fig. 1 will be used to illustrate the discussion. The voltage applied to the pump (v) is an input variable. The cross-sections of the tank (A) and the outlet hole (a), the pump parameter (k) and the gravitational acceleration (g) are time-independent properties of the physical system.

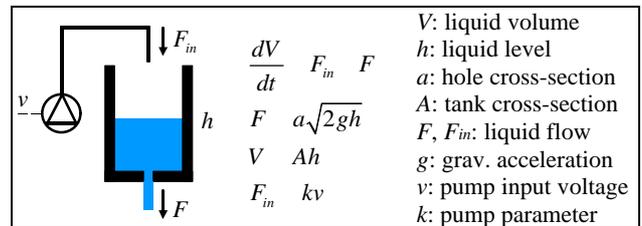


Figure 1: Model of a Process

Interactive Changes on the Model State

Two input variables to the DymolaBlock block are used to model the interactive changes in the state: *Istate*[:] and *CKstate* (see Fig. 2). The array *Istate* contains the values used to reinitialize the model state. The signal *CKstate* is used to trigger the state re-initialization event. The state re-initialization is performed using a built-in Modelica operator: *reinit(x,expr)*. It re-initializes a state variable (x) with the value obtained of evaluating an expression ($expr$), at an event instant. The output-variable array of the DymolaBlock block (*O*[:]) contains the variables representing the actual value of the state, in addition to the other variables linked to the view elements. Ejs uses the value of this output array (*O*[:]) to refresh the simulation view. When the user changes the value of one or several state variables, Ejs writes the new state in *Istate* array and changes the value of *CKstate* (from one to zero or from zero to one) in order to trigger the re-initialization event.

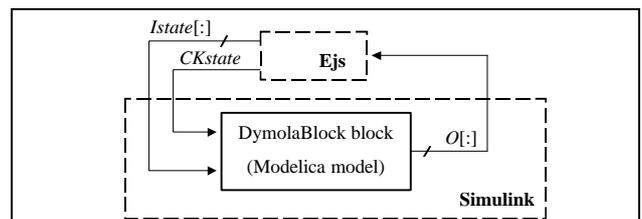


Figure 2: Interactive Change of the State

Different choices of the model state-variables are possible. For instance, possible choices in the model of Fig. 1 include: $e_1 = h$, $e_2 = V$ and $e_3 = F$; where e_i represents one particular choice of the state variables. If the user wants to change interactively the level value (h), the appropriate choice is $e_1 = h$. Likewise, if the user wants to change V , then the right choice is e_2 ; and if he wants to change F , then e_3 . The model should support the following feature: every time the user needs to change the state value, he decides to represent it in terms of a change in either the volume or the height or the flow. Different choices are possible during a given interactive simulation run.

In general, an interactive model is required to support changes in its state value that correspond with different choices of the state variables. An approach to fulfill this requirement is the following. Modeling the interactive model as composed of several instantiations of the physical model, each one with a different choice of the state variables. Modelica capability for state selection control allows the implementation of this approach conveniently. In addition, the interactive model needs to describe the information flow among these different instantiations of the physical model. These topics will be discussed later.

Changes on the Interactive Parameters

Time-independent properties of the system are usually represented by model parameters. However, sometimes one of the interactive-simulation goals is studying the dependence between the model dynamic behavior and the value of these properties. In this case, the property value can be instantaneously changed by the user's action, remaining constant between consecutive interactive changes. These variables of the interactive model are called *interactive parameters*. DymolaBlock block inputs should include the variables required to describe the changes in the interactive-parameter values.

Changes in the value of interactive parameters can have different effects depending on the state variable choice. Consider an instantaneous change in the cross-section (A) of the model in Fig. 1 (for instance, the user interactively changes A from 1 cm^2 to 3 cm^2). If the state variable is the volume (V), then the change in A produces an instantaneous change in the value of the liquid height (h) and flow (F), while the liquid volume remains constant. On the contrary, if the state variable is the height or the flow, these magnitude values do not change as the result of an instantaneous change in A . In this case, the volume does change.

In general, interactive models need to support instantaneous changes in the value of the interactive parameters, for different choices of the state variables. This requirement can be accomplished by using the adequate physical-model instantiation (that with the

required state selection) for executing the instantaneous change in the parameter value and for solving the re-start problem. Next, these calculated values are used to re-initialize the other physical-model instantiations. This action guarantees that all physical-model instantiations describe the same trajectory.

State Selection Control

Modelica 2.0 supports the user's control on the state variables selection, via the *stateSelect* attribute of Real variables (Otter and Olsson 2002). This attribute values include "never" (the variable will never be selected as state variable) and "always" (the variable will always be used as a state). This Modelica feature allows controlling the model state selection by means of a Boolean array. State selection of the model in Fig.1 can be accomplished as shown below, by means of the Boolean vector *isState*. For instance, if *isState* is set to the value {false,true,false} when instantiating the physical-model, then the volume is selected as a state variable.

```

model tank
  Real h (stateSelect = if hIsState
    then StateSelect.always else StateSelect.never);
  Real volume (stateSelect = if volumeIsState
    then StateSelect.always else StateSelect.never);
  ...
end tank;

model pipe
  Real F (stateSelect = if FIsState
    then StateSelect.always else StateSelect.never);
  ...
end pipe;

partial model physicalModel
  parameter Boolean[3] isState;
  tank tank1 ( hIsState = isState[1],
    volumeIsState = isState[2] );
  pipe pipe1 ( FIsState = isState[3] );
  ...
end physicalModel;

```

Conceptual Foundations of State Selection Control

Dymola 5.1 (Mattsson et al. 2000; Dynasim 2002) supports this Modelica capability: the user's control on the state-variable selection. The continuous part of a Modelica model is mapped to a DAE of the form $\mathbf{f} \dot{\mathbf{x}}_1, \mathbf{x}_1, \mathbf{x}_2, t = \mathbf{0}$, where \mathbf{x}_1, t are variables appearing differentiated, and \mathbf{x}_2, t are pure algebraic variables. Conceptually, once the state-variables (i.e., \mathbf{x}_s, t) have been selected, Dymola transforms automatically this DAE into the state space form:

$$\begin{aligned} \dot{\mathbf{x}}_s &= \mathbf{f}_1(\mathbf{x}_s, t) \\ \mathbf{x}_n &= \mathbf{f}_2(\mathbf{x}_s, t) \end{aligned} \quad (1)$$

The tank model shown in Fig. 1 will be used to illustrate the conceptual foundations of this transformation. According to the model description in Fig. 1: $\mathbf{x}_1 = V$.

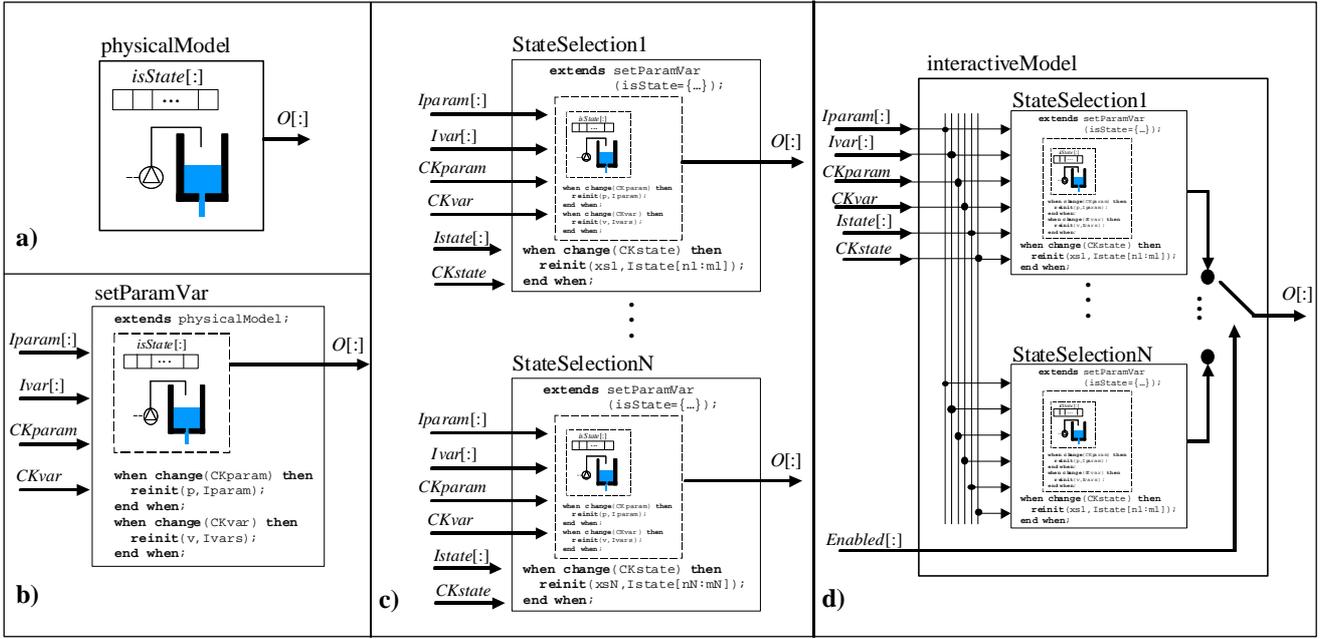


Figure 3: Schematic Description of the Proposed Modeling Methodology for Interactive Simulation

In order to reformulate the model as shown in Equation (1), with \mathbf{x}_s h , a dummy variable \mathbf{a} and an equation ($\dot{h} \mathbf{a}$) are included in the model (H. Elmquist: personal communication, 1995). These changes do not modify the model: the equation $\dot{h} \mathbf{a}$ is used to calculate \mathbf{a} . The index of the extended model is greater than one (Brenan et al. 1996). The model index can be reduced to one or zero by applying the Pantelides algorithm (Pantelides 1998) and by including the differentiated equations in the model (Mattsson and Soderlind 1992). Then, the dummy derivatives are defined so that the user's state selection is fulfilled. The result of these manipulations is shown below, where the variable to evaluate from each equation is written within square brackets.

$$\begin{array}{rcl}
 \dot{V} & F_{in} & F \\
 F & a\sqrt{2gh} & \\
 V & Ah & \\
 F_{in} & kv & \\
 \dot{h} & \mathbf{a} & \\
 & & F \quad a\sqrt{2gh} \\
 & & F_{in} \quad kv \\
 & & derV \quad F_{in} \quad F \\
 & & V \quad Ah \\
 & & derV \quad \dot{A}h \quad A[\dot{h}] \\
 & & \dot{h} \quad \mathbf{a}
 \end{array}$$

Interactive Parameters and Input Variables

This method to formulate the model according to a given state selection can require differentiating an interactive parameter or an input variable. In the previous example, the time-derivative of the tank cross-section needs to be calculated. If the interactive parameter A is defined as an input variable, then an error is produced: Dymola cannot differentiate an input variable.

A valid approach consists in defining the interactive parameters as constant state-variables (i.e., with zero time-derivative). The interactive changes in the value of these parameters are implemented by re-initializing their values. An analogous reasoning is applicable to the input variables of the physical model, and an analogous solution is implemented. Four input variables to the DymolaBlock block are used to model the changes in the value of the interactive parameters and input variables: two signal arrays ($Iparam[:]$, $Ivar[:]$) containing the new values, and two signals ($CKparam$ and $CKvar$) for triggering the re-initialization events.

Modeling Methodology for Interactive Simulation

Summarizing the previous discussion, the proposed methodology consists in the following steps:

1. Object-oriented modeling of the physical system: model *physicalModel*. The Boolean vector *isState* controls the state selection (see Fig. 3a).
2. Define the model *setParamVar* (see Fig. 3b). It inherits *physicalModel* and contains the when-clauses to change the value of the interactive parameters ($p[:]$) and input variables ($v[:]$).
3. Define as many models (*stateSelection1*, *stateSelection2*, ...) as different state-variable choices needed (e_1, e_2, \dots). Each of these models inherits *setParamVar* (*isState* array is set to the value adequate in each case) and contains a when-clause to change the value of the corresponding state-variable array (see Fig. 3c).
4. Define the model *interactiveModel*, composed of all the models defined in the previous step (*stateSelection1*, ..., *stateSelectionN*). The value of the input array *Enabled*[1:N] is set by Ejs, and it selects which output is connected to the output signal ($O[:]$).

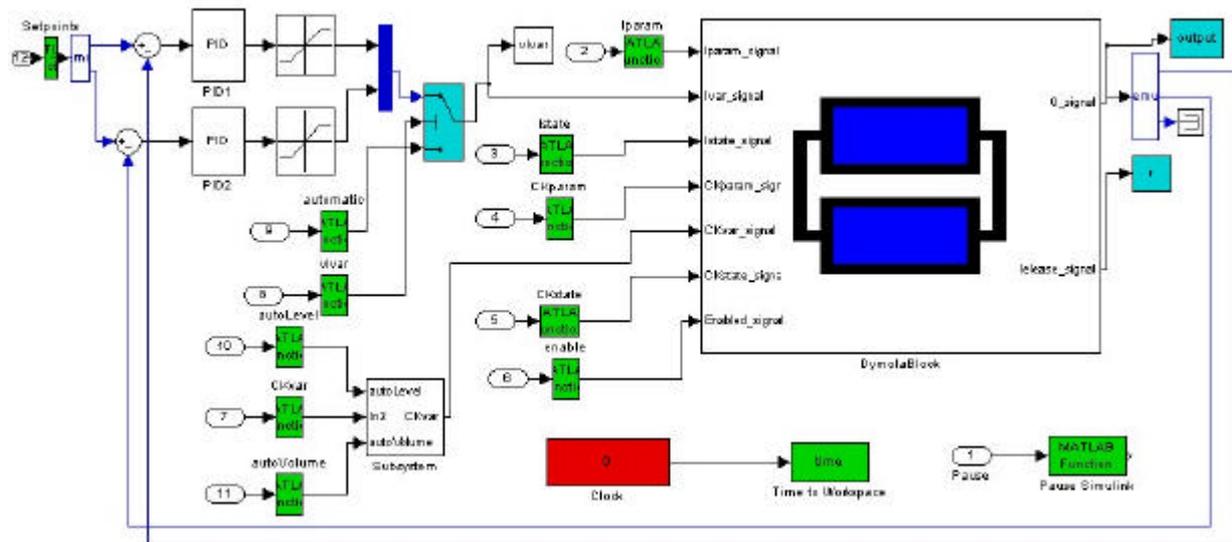


Figure 4: Simulink Model of the Quadruple-Tank Process and its Control System

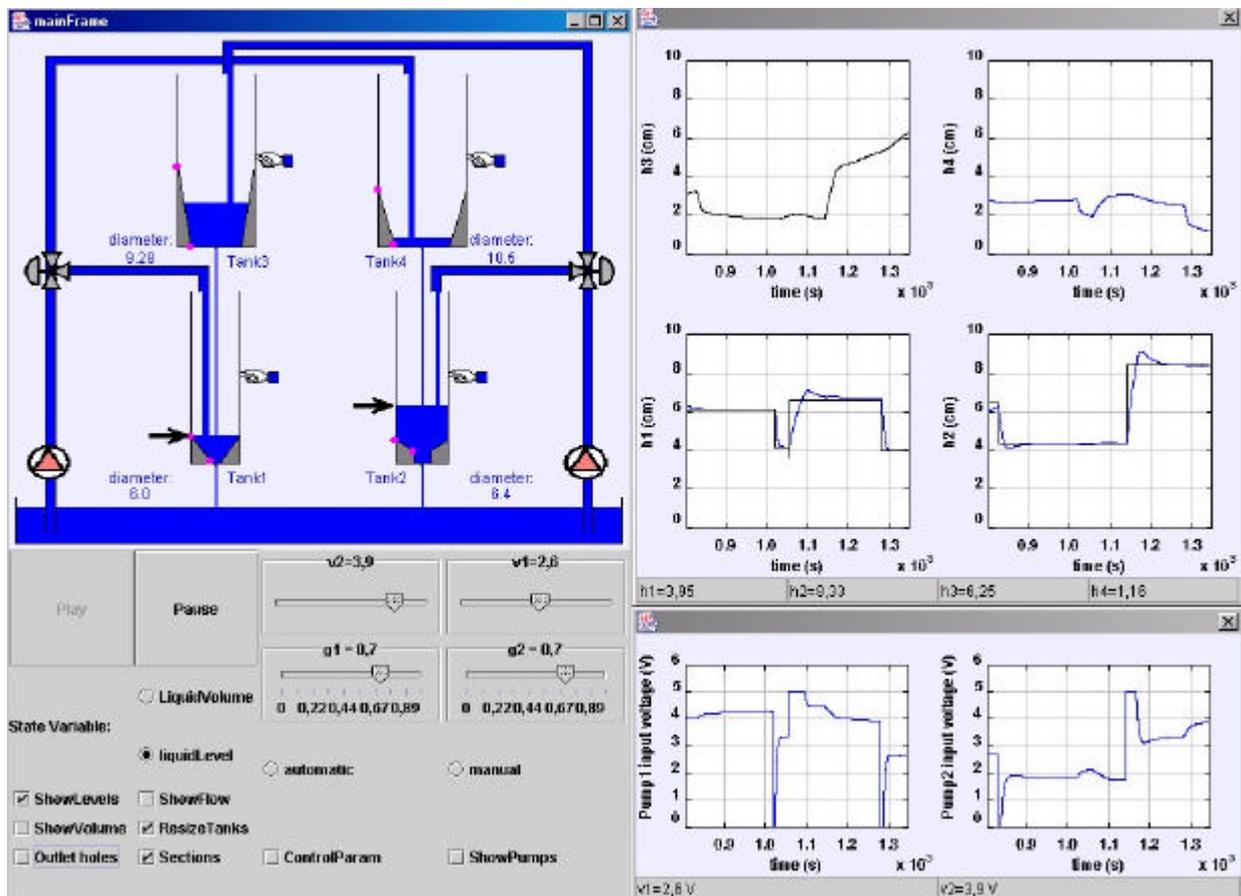


Figure 5: View of the Quadruple-Tank System

The obtained Modelica model is schematically represented in Fig. 3d. It is suitable to be embedded in a DymolaBlock block and connected to Ejs for interactive simulation.

CASE STUDY: INTERACTIVE SIMULATION OF THE QUADRUPLE-TANK PROCESS

The interactive simulation of the quadruple-tank process has been implemented (see Fig. 5), by the combined use

of Ejs, Matlab/Simulink and Modelica/Dymola. This process has been used to illustrate different aspects of the multi-variable control theory (Johansson 2000; Dormido and Esquembre 2003): the goal is to control the level of the lower two tanks with two pumps. Two different control strategies have been implemented in this case study: manual control and decentralized PID. Switching between these control strategies can take place during the simulation run. The parameters of the PID controllers can be changed interactively.

The simulation supports interactive changes in the tank physical parameters: cross-section, shape and cross-section of the outlet hole. Tank cross-section and shape changes can take place under one of two alternative conditions: the liquid volume inside the tank is kept constant or the liquid height is kept constant. In addition, the simulation supports interactive changes in the amount of liquid stored inside the tanks. These changes can be defined in terms of the liquid height or the liquid volume.

The physical model of the quadruple-tank process has been implemented in Modelica language. It is analogous to the model shown in Fig. 1. Mass balance and Bernoulli's law are applied to model the tanks and the flows. The process model (DymolaBlock block) and the control system are shown in Fig. 4. The automatic control system consists in two PID blocks followed by saturators. Depending on a variable that control a switch, the pump voltage values sent to Dymolablock are the ones obtained by PID controllers or the ones introduced by the user.

The simulation view is shown in Fig. 5. The main window (on the left side of Fig. 5) contains the schematic diagram of the process (above) and the control buttons (below). Both of them allow the user to experiment with the model. The liquid levels, the tank cross-sections and the level setpoints represented in the schematic diagram are linked to the respective model variables: their values can be interactively changed by dragging with the mouse. The sliders placed under the schematic diagram allow changing interactively the PID parameters and the cross-sections of the outlet holes. The control buttons allow choosing the state variables (liquid volumes or heights). They also open and close graphic plots of the liquid levels, volumes and flows, and plots of the voltages applied to the pumps. Some of these plots are displayed on the right side of Fig. 5.

CONCLUSIONS

The feasibility of combining Ejs, Matlab/Simulink and Modelica/Dymola for implementing interactive simulations has been discussed. This approach takes advantage of Ejs capability for building interactive user-interfaces, Matlab/Simulink capability for modeling of automatic control systems, Modelica capability for physical modeling, and Dymola capability for simulating hybrid-DAE models. In addition, a novel modeling methodology adequate for interactive simulation has been proposed, and it has been successfully applied to a case study: the interactive simulation of the quadruple-tank process.

REFERENCES

Astrom, K.J.; H. Elmquist; and S.E. Mattsson. 1998. "Evolution of Continuous-Time Modeling and Simulation". In *Proceedings of the 12th European Simulation Multiconference* (Manchester, UK).

- Dormido, S. and F. Esquembre. 2003. "The Quadruple-Tank Process: An Interactive Tool for Control Education", In *Proceedings of the 2003 European Control Conference*, (Cambridge, UK).
- Dynasim. 2002. "Dymola. User's Manual. Version 5.0a". Dynasim AB, Lund, Sweden.
- Esquembre, F. 2004. "Easy Java Simulations: a software tool to create scientific simulations in Java" *Computer Physics Communications*, 156, 199-204.
- Johansson, K.H. 2000. "The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero". *IEEE Transactions on Control Systems Technology*, Vol. 8, No. 3 (May), 456-465.
- Mattsson, S.E.; H. Olsson; and H. Elmquist. 2000. In *Proceedings of the Modelica Workshop 2000*, (Lund, Sweden), pp. 61 - 67.
- Otter, M. and H. Olsson. 2002. "New features in Modelica 2.0". In *Proceedings of the 2nd International Modelica Conference*, (Oberpfaffenhofen, Germany), 7.1 - 7.12.

AUTHOR BIOGRAPHIES

CARLA MARTIN received her M.S. degree in Electrical Engineering in 2001 from University Complutense of Madrid (UCM), Spain. She is currently completing her doctorate studies at UNED, Spain. (carla@dia.uned.es)

ALFONSO URQUIA was born in Madrid (1969). He received his M.S. degree in Physics in 1992 from UCM and his Ph.D. in 2000 from UNED. His work experience includes six years as an R&D engineer at AT&T and Lucent Technologies - Bell Labs. Currently he is working as an associate professor at UNED. (aurquia@dia.uned.es)

JOSE SANCHEZ received his Computer Sciences degree in 1994, from Madrid Polytechnic University and his Ph.D. in Sciences from UNED in 2001. Since 1993, he has been working at UNED Department of Computer Sciences and Automatic Control as an Assistant Professor. (jsanchez@dia.uned.es)

SEBASTIAN DORMIDO received his Physics degree from UCM (1968) and his Ph.D. from Country Vasc University (1971). In 1981, he was appointed Full Professor of Control Engineering at UNED Faculty of Sciences. Since 2002, he is President of CEA-IFAC. (sdormido@dia.uned.es)

FRANCISCO ESQUEMBRE was born in Alicante (1963) and received the Ph.D. degree in Mathematics in June 1991, from the University of Murcia, Spain, where he works since 1986, holding a permanent job as Assistant Professor since 1994. (fem@um.es)

JOSE L.GUZMAN was born in Almería, Spain. He received his Computer Science degree in 2002 from the University of Almería. Currently, he is working in his Ph.D at the University of Almería. (joguzman@ual.es)

MANUEL BERENGUEL is associate professor in the University of Almería, Spain. (beren@ual.es)

A NEW METHOD FOR ORDERING SPARSE MATRICES AND ITS PERFORMANCE IN CIRCUIT SIMULATION

Gunther Reißig

Otto-von-Guericke-Universität Magdeburg
Chair of Systems Theory (J. Raisch), FEIT-IFAT
PF 4120
D-39016 Magdeburg, Germany
URL: <http://www.reiszig.de/gunther/>

ABSTRACT

Local algorithms for obtaining a pivot ordering for sparse symmetric coefficient matrices are reviewed together with their mathematical background and appropriate data structures. Recently proposed heuristics as well as improvements to them are discussed, and their performance, mainly in terms of the resulting number of factorization operations, is compared with that of the Minimum Degree and the Minimum Local Fill algorithms. It is demonstrated that a combination of Markowitz' algorithm with these symmetric methods applied to the unsymmetric matrices arising in circuit simulation is capable of accelerating the simulation significantly.

1. INTRODUCTION

Simulating an electrical circuit requires the solution of numerous linear equations of the form

$$Ax = b, \quad (1)$$

where A is a real or complex $n \times n$ matrix [Chua et al., 1987, Feldmann et al., 1992, Nagel, 1975]. Solving (1) is also necessary in other types of analyses, such as DC- and small signal analysis, and the efficiency by which this is done determines the quality of simulation tools as a whole to a great extent.

In many professional simulation tools, including TITAN [Feldmann et al., 1992], (1) is solved directly without pivoting for numerical accuracy [Hajj et al., 1981, Tan, 1986, Nagel, 1975], despite the fact that A is unsymmetric and indefinite. (To investigate the merits of that approach is beyond the scope of this paper.) In fact, one assumes that it is numerically feasible to solve

$$(PAP^T)y = Pb \quad (2)$$

by Gaussian elimination for any $n \times n$ permutation matrix P and then set $x = P^T y$. That is, the diagonal entries may be chosen as pivots, in any order [Hajj et al., 1981, Nagel, 1975].

Usually, A is very large and extremely sparse, i.e., only few of its entries are nonzero. However, new nonzeros are introduced during the elimination phase:

The coefficient matrix PAP^T is successively overwritten by the matrix $L + U - \text{id}_n$, where

$$LU = PAP^T \quad (3)$$

and L and U are factors of PAP^T , L is unit lower triangular, U is upper triangular, and id_n is the $n \times n$ identity matrix [Chua and Lin, 1975, Golub and Van Loan, 1993, Nagel, 1975].

In general, the matrix $L + U - \text{id}_n$ may have nonzeros at positions where the coefficient matrix PAP^T has zeros. Those newly created nonzeros are called *fill-ins*, and their total is called *fill* [Duff et al., 1986, George and Liu, 1989].

As the coefficient matrices of all linear equations to solve usually have the same zero-nonzero pattern, the same permutation matrix P may be used throughout the simulation [Nagel, 1975, Chua and Lin, 1975]. With the amount of fill created, the computational complexity of solving equation (1) heavily depends on P . Thus, a deliberate choice of P is extremely important for the overall performance of circuit simulation software.

Unfortunately, minimizing the fill appears to be harder than solving the linear equations at hand without taking advantage of their sparsity [Yannakakis, 1981]. Therefore, various heuristics have been proposed, which aim at acceptable low fill rather than minimum fill [Duff et al., 1986]. Among them, there are what is called *local* algorithms, which mimic Gaussian elimination based on the zero-nonzero pattern of the coefficient matrix alone, where it is assumed that nonzeros do not accidentally cancel out.

Those local methods always choose a pivot that minimizes some scoring function and thereby determine an overall pivoting order.

MARKOWITZ was the first to propose heuristics for fill-in minimization in Gaussian elimination [Markowitz, 1957]. In his *Minimum Local Fill (MF)* algorithm, the score of a diagonal entry is the number of fill-ins that would be created in the next elimination step if that en-

try was chosen as the next pivot. The following scoring function was also proposed in [Markowitz, 1957]:

Let c_i and r_i be the number of off-diagonal nonzeros in the i th column and row, respectively, of the zero-nonzero pattern that has been obtained from the preceding eliminations. The product $c_i r_i$ is called the *Markowitz product* and is an upper bound on the number of fill-ins introduced when the i th variable in the current scheme is eliminated next using the i th equation. To chose a pivot with minimum Markowitz product has become known as *Markowitz' algorithm*.

If A is structurally symmetric, i.e., $A_{i,j} \neq 0$ iff $A_{j,i} \neq 0$, Markowitz' algorithm is called *Minimum Degree (MD) algorithm* [George and Liu, 1989].

The computational cost of the MF algorithm may be two orders of magnitude higher than that of a version of the MD algorithm [Ng and Raghavan, 1999], which is a serious drawback. In addition, it has been found especially in circuit simulation that the MF algorithm saves at most 5% in factorization operations compared with other algorithms [Nagel, 1975].

Only recently, it has been found that the MF algorithm as well as newly proposed variants of both the MF and MD algorithms yield significantly better orderings than the MD algorithm on certain test suites of symmetric matrices [Cavers, 1989, Lustig et al., 1992, Mészáros, 1998, Rothberg and Eisenstat, 1998, Ng and Raghavan, 1999]. The running times of some of those algorithms are just in the order of those of the MD algorithm [Cavers, 1989, Mészáros, 1998, Rothberg and Eisenstat, 1998, Ng and Raghavan, 1999].

The savings in fill of those variants heavily depend on the field of application and increase with increasing problem size [Cavers, 1989], but, unfortunately, extensive tests and comparisons of those variants on matrices derived from circuits of a size representing today's simulation tasks are still missing.

A first step towards efficient algorithms for high-quality ordering of sparse, unsymmetric matrices in circuit simulation was presented in [ReiBig and Klimpel, 2001] and tested in [ReiBig, 2001]:

Initially, diagonal entries with Markowitz product zero are chosen as pivots, as many as possible. The corresponding elimination steps do not create any fill, hence the zero-nonzero pattern obtained after those steps is that of a submatrix \tilde{A} obtained from A by removing the pivot rows and columns. Now, a symmetric ordering method is applied to $|\tilde{A}| + |\tilde{A}|^T$, thereby completing the pivoting order for A . That way, much of the unsymmetry of A is removed by the initial steps of Markowitz' algorithm so that \tilde{A} will be symmetric or nearly so.

It was shown in [ReiBig, 2001] that the orderings obtained from recently proposed ordering heuristics for symmetric matrices combined with the above method from [ReiBig and Klimpel, 2001, ReiBig, 2001] require up to 31% fewer factorization operations than those obtained from the Minimum Degree algorithm, in some cases at virtually no extra computational cost, when applied to

symmetrized Jacobians of modified nodal equations.

The purpose of this paper is to demonstrate that, in terms of the resulting number of factorization operations, the method from [ReiBig and Klimpel, 2001] outperforms the MD and Markowitz' algorithms even if applied directly to the unsymmetric matrices that arise in circuit simulation. In fact, our tests on a test suite of 15 matrices extracted from the circuit simulator TITAN of Infineon Technologies shows that the method we propose is capable of saving up to 72% (38% on average) of factorization operations if compared to Markowitz' algorithm.

The remaining of this paper is structured as follows. In section 2, we review PARTER's interpretation of Gaussian elimination in terms of graphs [Parter, 1961] and the data structure underlying local symmetric ordering methods. These concepts have already been the basis for the definition of recently proposed ordering heuristics in [ReiBig, 2001]. For the convenience of the reader, we compile them in Section 3.

In Section 4, we compare the performance of the MD and MF algorithms and the algorithms from Section 3 on a test set of 15 unsymmetric Jacobians of modified nodal equations extracted from the circuit simulator TITAN [Feldmann et al., 1992] of Infineon Technologies.

2. DATA STRUCTURE OF LOCAL SYMMETRIC ORDERING METHODS

Throughout this section, A is a structurally symmetric $n \times n$ matrix, i.e., $A_{i,j} \neq 0$ iff $A_{j,i} \neq 0$, for all i and j , $1 \leq i, j \leq n$.

2.1. Gaussian elimination in terms of graphs

A *graph* is an ordered pair (V, E) of a finite set V of *nodes* and a set E of *branches*, $E \subseteq \{\{v, w\} \subseteq V \mid v \neq w\}$.

Two nodes $v, w \in V$ are *adjacent* in the graph G , $G = (V, E)$, if $\{v, w\} \in E$. For $w \in V$ and $W \subseteq V$, $\text{adj}_G(w)$ and $\text{adj}_G(W)$ denote the *adjacent set* of w and W , respectively, i.e.,

$$\begin{aligned} \text{adj}_G(w) &= \{u \in V \mid \{u, w\} \in E\}, \\ \text{adj}_G(W) &= \bigcup_{u \in W} \text{adj}_G(u) \setminus W. \end{aligned}$$

For $X \in V \cup \mathcal{P}(V)$, we denote the *degree of X in G* by $\text{deg}_G(X)$,

$$\text{deg}_G(X) = |\text{adj}_G(X)|,$$

where $|\cdot|$ denotes cardinality, and $\mathcal{P}(V)$ is the *power set* of V .

A node $v \in V$ and an branch $e \in E$ are *incident* if $v \in e$.

The *graph of A* , denoted $\mathcal{G}(A)$, is the graph (V, E) defined by

$$\begin{aligned} V &= \{1, \dots, n\}, \\ E &= \{\{i, j\} \mid A_{i,j} \neq 0, i \neq j\}. \end{aligned}$$

The *elimination graph* G_v is obtained by *eliminating* $v \in V$ from $G = (V, E)$, i.e., by removing v and its incident branches from G , and connecting all nodes previously adjacent to v [Parter, 1961], thereby creating a set $\text{fill}_G(v)$ of new branches or *fill-ins*.

Choosing pivots down the diagonal in PAP^T for some $n \times n$ permutation matrix P corresponds to selecting nodes of $\mathcal{G}(A)$ in the order $\pi(1), \pi(2), \dots, \pi(n)$ for some bijection $\pi : V \rightarrow V$, which we call the *pivoting order*. Thus, local ordering algorithms are equivalent to the selection of nodes as follows:

Input: Graph $G = (V, E)$, scoring function s .
Step 1: $S := \emptyset$.
Step 2: Pick $v \in V \setminus S$ with $s(v, G) = \min_{w \in V \setminus S} s(w, G)$.
Step 3: $S := S \cup \{v\}$, $\pi(|S|) := v$, $G := G_v$.
Step 4: If $S \neq V$, goto Step 2.
Output: Pivoting order π .

In particular, if the input graph G equals $\mathcal{G}(A)$, the above algorithm is the MD and the MF algorithm for $s(v, G) = \deg_G(v)$ and $s(v, G) = |\text{fill}_G(v)|$, respectively.

2.2. Clique representations of elimination graphs

A *clique* is a graph in which any two distinct nodes are adjacent. The node set of a clique will also be called a clique.

A set $\mathcal{C} \subseteq \mathcal{P}(V)$ is called a *clique representation* of the graph $G = (V, E)$ if $E = \{\{v, w\} \subseteq C \mid C \in \mathcal{C}, v \neq w\}$. In other words, clique representations of $\mathcal{G}(A)$ correspond to coverings of the nonzeros of A by full symmetric minors. Note also that E is a (trivial) clique representation of G .

If \mathcal{C} is a clique representation of G and $v \in V$, then the set \mathcal{C}' ,

$$\mathcal{C}' = \{C \in \mathcal{C} \mid v \notin C\} \cup \{\text{adj}_G(v)\} \quad (4)$$

is a clique representation of G_v .

2.3. Indistinguishable nodes

The nodes $v, w \in V$ are *indistinguishable* [George and Liu, 1989], $v \sim_G w$, if

$$\text{adj}_G(v) \cup \{v\} = \text{adj}_G(w) \cup \{w\}. \quad (5)$$

If $v \sim_G w$, then $\deg_G(v) = \deg_G(w)$ and $\text{fill}_G(v) = \text{fill}_G(w)$. Moreover, for any reasonable scoring function, it should suffice to determine the score of one of v and w only. Hence, we may maintain the quotient graph G/\sim_G rather than G itself, where $G/\sim_G = (V/\sim_G, E')$ and

$$E' = \{\{[v]_{\sim_G}, [w]_{\sim_G}\} \mid \{v, w\} \in E, [v]_{\sim_G} \neq [w]_{\sim_G}\}.$$

3. IMPROVED SCORING FUNCTIONS

Let G be the current elimination graph, $G = (V, E)$, \sim be the equivalence relation of indistinguishable nodes, and let \mathcal{C} be a clique representation of G/\sim . For simplicity, we denote the class $[v]_{\sim}$ by $[v]$.

For each node $[v]$ of G/\sim , let $\kappa_{[v]}$ be the list of cliques containing $[v]$. Let $c_{[v]}$ be the number of cliques in that list that have been created by eliminating nodes and assume that those created cliques are located at the beginning of the list.

We define scoring functions in terms of G/\sim . For $W \subseteq V/\sim$, we define

$$\|W\| = \sum_{[w] \in W} |[w]|.$$

3.1. Bounds on the local fill

The scoring function of the MD algorithm represents an upper bound on the number of fill-ins introduced by eliminating a node from G , since

$$\xi : x \mapsto (x^2 - x)/2$$

is monotonic on the set of nonnegative integers and $|\text{fill}_G(v)| \leq \xi(\deg_G(v))$ for all $v \in V$.

That bound may be improved since some of the $\xi(\deg_G(v))$ potential fill-in branches in G_v are branches of G that are easy to identify:

First, if v is indistinguishable from w in G , $v \neq w$, then eliminating v from G does not create any new branches adjacent to w in G_v . Hence, the *external degree* $\deg_G([v])$ of v represents an upper bound on $|\text{fill}_G(v)|$,

$$|\text{fill}_G(v)| \leq \xi(\deg_G([v])). \quad (6)$$

Taking the external degree as scoring function usually produces less overall fill than taking the degree [George and Liu, 1989].

Further, if $C \in \mathcal{C}$ is a clique, then the elimination of $[v]$ from G cannot create any new branch $\{[u], [w]\} \subseteq C$. Even if only those cliques are considered that contain $[v]$, the bound in (6) may be improved in several ways:

The *Approximate Minimum Local Fill (AMF)* algorithm of ROTHBERG and EISENSTAT uses the upper bound s_{AMF_0} ,

$$s_{AMF_0}([v]) = \begin{cases} 0 & \text{if } c_{[v]} = 0 \\ \xi(\deg_G([v]) - \xi(\|\kappa_{[v]}(1) \setminus \{[v]\}\|)) & \text{otherwise} \end{cases}$$

as its scoring function, thereby taking into account the most recently eliminated clique only [Rothberg and Eisenstat, 1998]. We will denote that scoring function that takes in account the largest, rather than the most recently created, clique, by s_{AMF_1} .

NG and RAGHAVAN propose to consider all cliques in $\kappa_{[v]}$ rather than just one [Ng and Raghavan, 1999].

3.2. Looking ahead

While local ordering algorithms usually consider the fill introduced in the next elimination step only, the concept of indistinguishability provides a simple means to look some steps ahead:

Since the total number of fill-ins created when all nodes in $[v]$ are eliminated from G immediately upon each other is just $|\text{fill}_G(v)|$, ROTHBERG and EISENSTAT consider dividing fill bounds by $|[v]|$ [Rothberg and Eisenstat, 1998]. Their (*Approximate*) *Minimum Mean Local Fill* ((A)MMF) heuristics are based on the scoring functions s_{MMF^α} and $s_{AMMF_0^\alpha}$,

$$s_{MMF^\alpha}([v]) = |\text{fill}_G(v)|/|[v]^\alpha,$$

$$s_{AMMF_0^\alpha}([v]) = s_{AMF_0}([v])/|[v]^\alpha$$

with $\alpha = 1$. According to [Rothberg and Eisenstat, 1998], a version with $\alpha = 1/2$ “produced slightly better results”.

We denote by $s_{AMMF_1^\alpha}$ that scoring function that results from application of the above trick to s_{AMF_1} .

3.3. Further variants

Among the various improvements of the MD and MF algorithms that we do not discuss in this paper are the tie breaking techniques of [Cavers, 1989, Mészáros, 1998], the *Modified Multiple Minimum Degree* (MMMD) algorithm of [Ng and Raghavan, 1999], and the *correction terms* of [Rothberg and Eisenstat, 1998, Ng and Raghavan, 1999].

4. COMPUTATIONAL RESULTS

In addition to the techniques described in Section 2, our implementation of the MD and MF algorithms as well as their variants described in Section 3 includes further techniques, such as *element absorption*, *incomplete score update*, and *multiple elimination*, see [George and Liu, 1989].

The code was compiled with the cc compiler (Workshop Compilers 4.2 30 Oct 1996 C 4.2) using the options “-fast -fsimple=2 -xtarget=ultra -xarch=v8ultra” under SunOS Release 5.7 and run on one of the CPUs (sparcv9+vis, 400 MHz clock rate, 4 MB cache, 17.4 SPECint95, 25.7 SPECfp95) of a SUN Enterprise E4500 workstation with 6 Gbytes of memory.

We chose the implementation of Markowitz’ algorithm of the circuit simulator TITAN [Feldmann et al., 1992], version 6.1a, as our reference algorithm.

Our primary measure for comparing ordering algorithms is the number

$$\sum_{i=1}^n c_i(1 + r_i) \quad (7)$$

of factorization operations determined by the pivoting orders obtained, which represents divisions and multiplications, where c_i and r_i are the number of off-diagonal

Problem characteristics						Markowitz’ A	MMD	AMMF ₁ ^{1/2}	MMF ₁ ^{1/2}
Matrix	$\frac{n_0}{10^3}$	$\frac{m_0}{10^3}$	$\frac{m_1}{m_0}$ in %	$\frac{u_0}{m_0}$ in %	$\frac{u_1}{m_1}$ in %				
bag	144	995	93	7.3	0	$3.5 \cdot 10^7$	0.95	0.81	0.68
cor	123	892	74	26	0	$1.8 \cdot 10^6$	0.87	0.87	0.87
eng	5	42	80	20	0	$1.1 \cdot 10^6$	0.96	0.97	0.88
jac	1	22	83	17	0	$1.0 \cdot 10^6$	1.12	1.28	0.92
m8	11	237	83	17	0	$3.9 \cdot 10^7$	0.77	0.56	0.43
m14	16	418	84	16	0	$7.1 \cdot 10^7$	0.79	0.64	0.55
m24	26	710	86	13	0	$4.1 \cdot 10^8$	0.94	0.72	0.49
m40	156	2039	88	12	0	$4.2 \cdot 10^9$	0.81	0.61	0.48
sei	4	292	96	3.7	0	$2.5 \cdot 10^8$	0.61	0.34	0.28
buc	10	78	58	51	17	$8.4 \cdot 10^5$	0.92	0.86	0.69
gue	89	549	69	32	2.3	$1.6 \cdot 10^7$	0.82	0.88	0.43
te	60	398	64	36	0.032	$7.6 \cdot 10^5$	1.00	1.01	0.99
tei	175	1206	67	33	0.0099	$4.7 \cdot 10^6$	1.08	1.04	0.86
xch	11	84	57	52	17	$8.3 \cdot 10^5$	1.02	0.82	0.73
X	16	170	67	40	11	$2.8 \cdot 10^6$	0.86	0.76	0.63
geometric mean							0.89	0.78	0.62

Table 1: Circuit matrices and performance of Markowitz’ algorithm and its combination with the MMD, AMMF₁^{1/2} and MMF₁^{1/2} algorithms defined in section 3.2. n_0 , m_0 , and u_0 denote the number of rows, nonzeros, and structurally unsymmetric nonzeros, respectively. m_1 and u_1 denote the number of nonzeros and structurally unsymmetric nonzeros, respectively, in the matrices remaining after the initial steps of Markowitz’ algorithm. o denotes the number of factorization operations (7), for the combinations divided by the corresponding value for Markowitz’ algorithm. The values in the row “geometric mean” are geometric means of the ratios reported in the respective column. The values of n_0 and m_0 are rounded to multiples of 1000, all ratios are rounded to a precision of 10^{-2} , and the remaining quantities are rounded to two decimal digits.

nonzeros in column and row i of the factors L and U , respectively, of PAP^T , and P is the permutation matrix corresponding to the pivoting order.

Our test suite of input data consists of 15 matrices extracted from the circuit simulator TITAN of Infineon Technologies. These matrices are listed under the names “bag” through “X” in Tab. 1.

It is well known that ordering heuristics are sensitive to permutations of the rows and columns of the input matrices. Therefore, for the combinations of Markowitz’ algorithm with symmetric ordering methods to the problems “bag” through “sei” listed in Tab. 1, we report the arithmetic mean of the number of factorization operations over 11 runs as in the previous paragraph. For Markowitz’ algorithm, and for the problems buc through X of Tab. 1, we report the result of one run only.

From the problem data presented in Tab. 1, we see that the initial steps of Markowitz’ algorithm only slightly reduce the dimension of the problem, but remove 4%–43%

of the nonzeros. Moreover, while 3.7% – 52% of the nonzeros of the original matrices are structurally unsymmetric, those initial steps remove most or all of them. In particular, the remaining matrices for the problems “bag” through “sei” are symmetric, and only about 0.01% – 17% of the nonzeros of the other remaining matrices are structurally unsymmetric.

For problems leading to symmetric remaining matrices, it is obvious that the advantages of the symmetric ordering methods investigated over the basic MD heuristic carry over to the combination of those methods with Markowitz’ algorithm. The results presented in Tab. 1 show that these advantages carry over to the combination even if the remaining matrices are structurally unsymmetric.

After all, the combination of Markowitz’ algorithm with the MMD, $\text{AMMF}_1^{1/2}$, and $\text{MMF}^{1/2}$ algorithms leads to 11%, 22%, and 38% fewer factorization operations than Markowitz’ algorithm alone, although for some of the problems from Tab. 1, MARKOWITZ’ algorithm yields better orderings than both MMD and $\text{AMMF}_1^{1/2}$.

Furthermore, it is evident that the running time of the above combination with the MMD algorithm should never exceed that of an analogous but unsymmetric implementation of Markowitz’ algorithm. In fact, the code of Markowitz’ algorithm we used was always much slower than its combination with both the MMD and the $\text{AMMF}_1^{1/2}$ heuristic.

In general, permuting coefficient matrices to block triangular form [Duff et al., 1986], of which removal of pivots with zero Markowitz product – the technique we applied – is a first step, followed by ordering and factoring the diagonal blocks can speed up the solution of linear equations even further. However, we found that the effect of that improvement is insignificant if applied to circuit matrices: For the circuit problems bag through sei from Tab. 1, removal of pivots with zero Markowitz product leads to irreducible matrices in all cases except m40. For those problems from Tab. 1 that lead to reducible matrices, the dimension and the number of nonzeros, respectively, of the largest diagonal block would always be greater or equal to 97.6% and 98.7%, respectively, of the corresponding numbers for the whole matrix. Furthermore, the number of structurally unsymmetric nonzeros would be reduced in two cases only and by approximately 0.1%.

A final decision on which of those heuristics is best would not only depend on the kind of circuits to be simulated, but also on the computer architecture and the specific numerical factorization algorithm used [Lustig et al., 1992] and is beyond the scope of this paper. However, let us show by an example that the savings in factorization operations of the $\text{MMF}^{1/2}$ over the $\text{AMMF}_1^{1/2}$ heuristic may very well reduce the overall simulation time:

The largest CPU time for the $\text{MMF}^{1/2}$ algorithm was 2h43min, achieved for the “m40” example for which ap-

plication of the $\text{AMMF}_1^{1/2}$ algorithm took only 47sec.. However, a transient simulation of that example with TITAN [Feldmann et al., 1992], version 6.1a, using the pivot ordering obtained from Markowitz’ algorithm, spent over 138h on factorizations, so that the savings of the $\text{MMF}^{1/2}$ algorithm in factorization operations of 21% relative to the $\text{AMMF}_1^{1/2}$ heuristic should outweigh the larger CPU time of the former.

5. CONCLUSIONS

We have reviewed recently proposed local symmetric ordering methods, i.e., methods for obtaining pivot orderings for sparse symmetric matrices, as well as some straightforward improvements to them. We have shown that for the purpose of circuit simulation, a combination of Markowitz’ algorithm with symmetric ordering methods yields pivot orderings significantly better than those obtained from Markowitz’ algorithm alone, in some cases at virtually no extra computational cost and that that combination is capable of accelerating circuit simulation significantly.

We could not make a final decision on what ordering algorithm is best. That decision would not only depend on the kind of circuits to be simulated, but also on the computer architecture and the specific numerical factorization algorithm used and is beyond the scope of this paper.

However, we think that a further improvement of the running times of those symmetric methods that are based on exact local fill counts, for example, through a combination of multiple elimination with the fill updating method of WING and HUANG [Wing and Huang, 1975, Vlach and Singhal, 1983], would make them superior to any other local ordering method known today when combined with Markowitz’ algorithm as described in [Reißig and Klimpel, 2001] and applied in this paper.

6. ACKNOWLEDGMENT

I thank P. I. Barton (Cambridge, MA), G. Denk (München), and Uwe Feldmann (München), for their valuable hints and comments.

7. REFERENCES

- [Cavers, 1989] Cavers, I. A. (1989). Using deficiency measure for tiebreaking the minimum degree algorithm. Technical Report 89-2, Dept. Comp. Sci., The Univ. Of British Columbia, Vancouver, B. C., Canada V6T 1W5.
- [Chua et al., 1987] Chua, L. O., Desoer, C. A., and Kuh, E. S. (1987). *Linear and Nonlinear Circuits*. McGraw-Hill.

- [Chua and Lin, 1975] Chua, L. O. and Lin, P.-M. (1975). *Computer-Aided Analysis of Electronic Circuits*. Prentice-Hall, Englewood Cliffs, NJ.
- [Duff et al., 1986] Duff, I. S., Erisman, A. M., and Reid, J. K. (1986). *Direct methods for sparse matrices*. Oxford University Press.
- [Feldmann et al., 1992] Feldmann, U., Wever, U. A., Zheng, Q., Schultz, R., and Wriedt, H. (1992). Algorithms for modern circuit simulation. *Archiv für Elektronik und Übertragungstechnik (AEÜ)*, 46(4):274–285.
- [George and Liu, 1989] George, A. and Liu, J. W. (1989). The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1):1–19.
- [Golub and Van Loan, 1993] Golub, G. H. and Van Loan, C. F. (1993). *Matrix Computations*. The John Hopkins Univ. Press, 2 edition.
- [Hajj et al., 1981] Hajj, I. N., Yang, P., and Trick, T. N. (1981). Avoiding zero pivots in the modified nodal approach. *IEEE Transactions on Circuits and Systems*, 28(4):271–278.
- [Lustig et al., 1992] Lustig, I. J., Marsten, R. E., and Shanno, D. F. (1992). The interaction of algorithms and architectures for interior point methods. In Pardalos, P. M., editor, *Advances in optimization and parallel computing*, pages 190–204. North-Holland.
- [Markowitz, 1957] Markowitz, H. M. (1957). The elimination form of the inverse and its application to linear programming. *Management Science*, 3:255–269.
- [Mészáros, 1998] Mészáros, C. (1998). Ordering heuristics in interior point LP methods. In Giannessi, F., Komlósi, S., and Rapcsák, T., editors, *New trends in mathematical programming*, pages 203–221. Kluwer Acad. Publ.
- [Nagel, 1975] Nagel, L. W. (1975). SPICE 2: A computer program to simulate semiconductor circuits. Technical Report ERL-M520, Univ. of Calif. Berkeley, Electronic Res. Lab., Berkeley, CA.
- [Ng and Raghavan, 1999] Ng, E. G. and Raghavan, P. (1999). Performance of greedy ordering heuristics for sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 20(4):902–914.
- [Parter, 1961] Parter, S. V. (1961). The use of linear graphs in Gauss elimination. *SIAM Review*, 3:119–130.
- [Reißig, 2001] Reißig, G. (2001). On methods for ordering sparse matrices in circuit simulation. In *Proc. 2001 IEEE Int. Symp. on Circuits and Systems (ISCAS), Sydney, Australia, May 6-9*, volume 5, pages 315–318.
- [Reißig and Klimpel, 2001] Reißig, G. and Klimpel, T. (2001). Verfahren zum computergestützten Vorhersagen des Verhaltens eines durch Differentialgleichungen beschreibbaren Systems. Patent DE 101 03 793 A 1. (“Method for computer-aided prediction of the behavior of a system described by differential equations”, in German).
- [Rothberg and Eisenstat, 1998] Rothberg, E. and Eisenstat, S. C. (1998). Node selection strategies for bottom-up sparse matrix ordering. *SIAM Journal on Matrix Analysis and Applications*, 19(3):682–695.
- [Tan, 1986] Tan, G.-L. (1986). An algorithms for avoiding zero pivots in the modified nodal approach. *IEEE Transactions on Circuits and Systems*, 33(4):431–434.
- [Vlach and Singhal, 1983] Vlach, J. and Singhal, K. (1983). *Computer methods for circuit analysis and design*. Van Nostrand Rheinhold.
- [Wing and Huang, 1975] Wing, O. and Huang, J. (1975). SCAP - a sparse matrix circuit analysis program. In *Proc. 1975 IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pages 213–215.
- [Yannakakis, 1981] Yannakakis, M. (1981). Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79.

8. AUTHOR BIOGRAPHIES

GUNTHER REIßIG received his Dipl.-Ing. (EE), Dipl.-Math. and Dr.-Ing. (EE) degrees from Technische Universität Dresden, Dresden, Germany, in 1991, 1995, and 1998, respectively. He was awarded the “Förderpreis” of the Information Technology Society, Germany, for his Dr.-Ing. thesis. From 1998 to 2000, he was with SIEMENS AG (München, Germany) and Infineon Technologies (München, Germany), where he contributed to the improvement of the diagnosis, VHDL, and sparse matrix capabilities of the circuit simulator TITAN. From 2000 to 2002, he was with the Department of Chemical Engineering of the Massachusetts Institute of Technology as a Postdoctoral Research Fellow. He is now with the chair of systems theory at the Department of Electrical Engineering of the Otto-von-Guericke Universität Magdeburg (Magdeburg, Germany) as an assistant. At present, he is particularly interested in the structural approach to systems analysis as well as discrete-continuous systems.

libcppsim: A SIMULA-LIKE, PORTABLE PROCESS-ORIENTED SIMULATION LIBRARY IN C++

Moreno Marzolla

Dipartimento di Informatica
Università Ca' Foscari di Venezia
via Torino 155, 30172 Mestre (VE), Italy
e-mail: marzolla@dsi.unive.it

KEYWORDS Process-Oriented Simulation, Simulation Library, C++.

ABSTRACT

In this paper we describe the design and implementation of `libcppsim`, a general-purpose, process-oriented simulation library written in C++. `libcppsim` provides a set of classes for implementing simulation processes, scheduling primitives, random variate generation and output data analysis functions. The main simulation entity provided by the library is the simulation process; the basic process scheduling primitives are modeled upon those provided by SIMULA's simulation class. The modular object-oriented design of `libcppsim` allows users to extend its functionalities with minimal effort. In order to improve efficiency, simulation processes are not implemented as operating system threads; instead, we use coroutine objects which implements a cooperative quasi-parallel process environment. A portable implementation of coroutines is provided in order to allow `libcppsim` to be used on different platforms.

1 INTRODUCTION

Simulation is a general modeling technique which can be used for analyzing systems which can not be described and evaluated analytically. Many tools and languages have been developed in order to facilitate the implementation of simulation models. SIMULA [Dahl and Nygaard, 1966] has been one of the first successful general-purpose programming languages providing special features targeted at simulation implementation purposes. SIMULA provides a set of basic facilities for describing a simulation model as a set of cooperating simulation processes executing in a quasi-parallel system. These facilities allow the description of many typical simulation models, and could also be used as building blocks for more sophisticated applications; DEMOS [Birtwistle, 1979] was one of the alternative simulation systems developed in SIMULA.

Traditional general-purpose programming languages can be used for implementing simulation programs as well. Support for simulation facilities is usually pro-

vided as libraries which can be linked to user's applications. Examples of simulation libraries based on general-purpose languages include CSIM19 [Schwetman, 2001] and C-Sim [Hlavička et al., 1999] (based on the C language), C++Sim [Little and McCue, 1993], OMNET++ [Varga, 2001] and DESP-C++ [Darmont, 2000] (based on the C++ language) and JavaSim [Little, Little] (written in Java). More sophisticated and user-friendly simulation tools have also been developed to make the implementation of simulation models easier for non-programmers. Such simulation environments are usually limited to particular models, such as those based on Queuing Networks or Petri Nets; they provide users with visual editors for graphically building the model.

Simulation languages and libraries can be based on an event-oriented or process-oriented paradigm. Event-oriented simulation models are described in term of the events which can change the system's state. A process-oriented simulation model is represented as a collection of concurrently executing simulation processes, each one with its own thread of execution. A simulation process is made of two parts: a set of local variables, and a sequence of actions to execute.

It should be noted that many existing simulation languages and tools suffer either a long learning time, poor performances, or can only be used for special-purpose models as they lack generality. Simulation libraries based on the event-oriented paradigm, while usually easier to learn, are not suited for models with a high number of different event types, as the resulting simulation programs would be very difficult to understand and debug. On the other hand, implementing the process-oriented simulation paradigm on the top of a conventional, general-purpose programming language can be difficult due to limitations (the main being the lack of support for coroutines) of the host language. Nevertheless, using a standard programming language is desirable because modelers are more likely to be already acquainted with it and do not require to learn some new idiom.

In this paper we describe `libcppsim`, a process-oriented, discrete simulation library written in C++. Our goal is to provide a simulation library based on a widely used and efficient object-oriented programming language. Object-orientation has long been recognized to be very useful for modeling complex systems; it is

not a surprise that SIMULA was one of the first object-oriented programming languages. `libc++sim` implements a limited set of simulation primitives which can be learned quickly and can be immediately used for implementing simulation models. The provided general framework can be extended if necessary with more complex, high-level functionalities.

Efficiency was one of our main concerns. For that reason the pseudo-parallel simulation process system is not based on the threading model of the underlying Operating System (OS). Threads are not handled efficiently in every OS, as thread switching may incur in the same overhead as process switching. Moreover, complex simulation models made of tens or thousands of simulation processes would probably overflow the resources (process or thread table size) of many operating systems. We address this problem by implementing the *coroutine* primitive as a C++ class. Coroutines implements a simple, user-level cooperative multitasking environment; while this lacks the features of multitasking, it is exactly what is needed to implement process-oriented simulations.

This paper is organized as follows. Section 2 describes how coroutines and simulation processes are implemented in `libc++sim`, and the process scheduling facilities are illustrated in Section 3. Random variate generation is described in Section 4 and output data analysis in Section 5. In Section 6 we show a usage example of the library, and conclusions are reported in Section 7.

2 SIMULATION PROCESSES

Coroutines [Marlin, 1980] are a programming construct which can be useful for implementing process-oriented simulation languages and libraries. Coroutines are blocks of code with associated state. A coroutine can suspend itself by calling another coroutine; however, unlike traditional subroutine call in which the caller is suspended until the callee terminates, coroutines may be reactivated in any order.

SIMULA has coroutines as a built-in facility; unfortunately, the C and C++ programming languages, which are widely used and for which efficient compilers are available, do not provide coroutines natively.

In C/C++, each function is associated to a data structure called *activation record* which is stored in the runtime (LIFO) stack. The activation record contains informations about the state of the routine, such as the value of local variables and the return address used to resume the execution after a function call. When a function call occurs, a new activation record is created and put on the top of the stack. All variables locally defined by the called routine are stored on the newly created activation record. When a routine terminates, its activation record is pushed from the stack. LIFO handling of the activation records does not work anymore with coroutines, because the currently active coroutine may not be the one associated with the topmost activation record in the stack. This implies that the order of activation of coroutines is not given by the LIFO stack handling. Also, it is not possible to grow

the stack of all the routines whose activation records are not on at the topmost position.

This problem can be solved in different ways. The first approach is based on the “copy-stack implementation” described by Helsgaun [Helsgaun, 1999]. The stack of the currently operating coroutine is kept in the C++ runtime stack. When a coroutine suspends, the runtime stack is copied in a buffer associated with the coroutine. The buffer is allocated in the dynamic heap, so it does not interfere with the normal stack operation. A coroutine is resumed by copying the content of that buffer back to C++’s runtime stack and setting the program counter to the saved execution location.

A second approach consists of making use of the context handling facilities provided by most Unix SysV-like environments. The OS provides functions allowing user-level context switching between different threads of control within a process using the `getcontext()`, `setcontext()` and `makecontext()` functions. In this approach each coroutine has its own run-time stack in a pre-allocated block in the heap.

Both approaches have advantages and disadvantages. The “copy-stack” approach is the most portable, as it can be implemented in every modern OS which implements the `setjmp()` and `longjmp()` system calls (almost everyone does). However, it requires a copy of the C++ runtime stack to be saved and restored every time a coroutine passes control, which may cause a considerable overhead. On the other hand, the approach based on the context handling facilities is simpler to use and more efficient, as the stack content is not copied. This approach is, however, less portable as not every OS implements the required system calls. Also, the context-based approach does not allow the coroutine context to grow past the dimension defined when the context is allocated. The user is required to set the maximum dimension of the context in advance; choosing a too small buffer results in erroneous runtime behaviors caused by stack corruption which can be difficult to debug. A complete discussion is given in [Helsgaun, 1999], and detailed portability considerations of different context manipulation strategies are described in [Engelschall, 2000].

The `libc++sim` library implements both stack-handling mechanisms, and the user can choose which one to use at compile time. As a general rule, the “copy-stack” variant is useful when developing the simulation program, as stack corruption errors do not occur and it is possible to measure the maximum stack size required by the application; then, when the program is verified, production runs may be done with the more efficient context-based approach.

Once coroutines are available, it is easy to define simulation processes on top of them. Fig. 1 shows the class hierarchy related to simulation process implementation. A simulation process is represented by the `process` class, and is derived from the `coroutine` class; a simulation process is a coroutine whose state must be preserved across invocations.

Each simulation process has a unique identifier and a

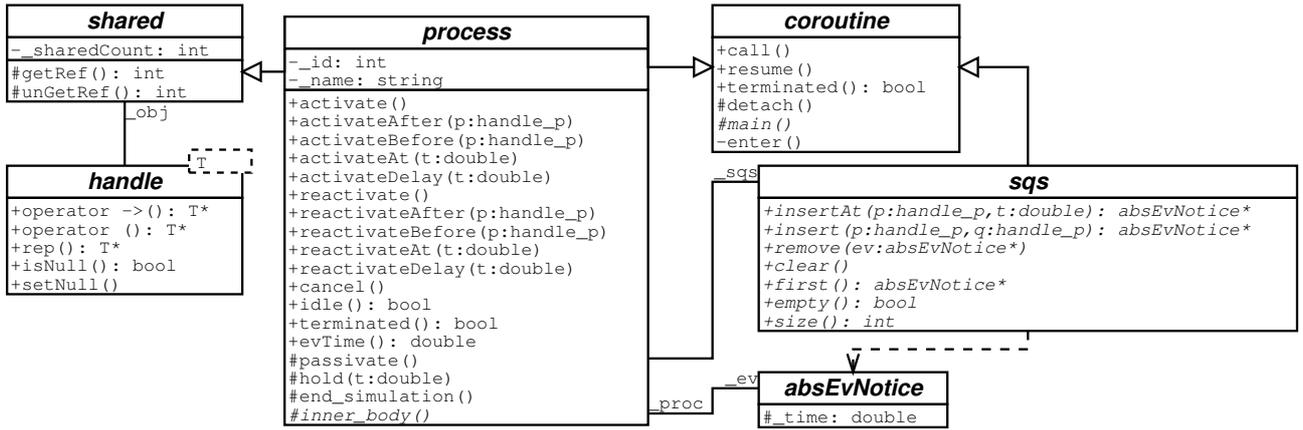


Figure 1: libcppsim Process Class Diagram

user-defined name. The process class has methods providing SIMULA-like scheduling primitives: it is possible to suspend the current process for a given amount of time, schedule the activation of another process in the future or cancel a pending process from the sequencing set. The actions performed by a simulation process are specified by defining the pure virtual method `inner_body()`. Fig. 2 shows the C++ interface of the process class.

The `handle<T>` class implements a “smart pointer” to an object of type T , where the class T must inherit from class `shared`. A smart pointer [Stroustrup, 1997] is a pointer to an object which keeps count of the number of times the object is referenced. When the last smart pointer is deallocated, the referenced object is automatically destroyed. Smart pointers are used for implementing a garbage collection mechanism. Our implementation of the `handle` class differs from the one proposed in [Stroustrup, 1997] as we store the reference count in the object rather than in the `handle`. This allows to preserve the counter even if the `handle` is converted to a C-style pointer and then back to an `handle`.

3 THE SEQUENCING SET

The Sequencing Set (SQS) is a data structure containing the list of simulation processes to be executed; the list is sorted by nondecreasing activation time order. Each simulation process contained in the Sequencing Set (SQS) is associated with an event notice. The event notice basically contains the simulated timestamp at which the process is to be resumed, and references to that process. Additional informations may be present in order to make insertion or removal of event notices more efficient.

libcppsim provides two different implementations of the SQS data structure. The first is defined in class `sqsDll`, and is based on a doubly linked list of event notices. This data structure is very simple to implement, but insertions of event notices require linear time on average to be performed. A more efficient implementation is given by class `sqsPrio`, based on balanced search trees. The expected insertion time in this case is proportional to

the logarithm of the SQS size. More efficient data structures (e.g. calendar queues [Brown, 1988]) can be implemented by supplying a corresponding class, without other modifications being required.

4 RANDOM VARIATE GENERATION

Simulation programs require an efficient and statistically robust mechanism for producing streams of pseudo-random numbers with given distribution. libcppsim defines an abstract templated class `rng<T>` representing a pseudo-random number generator of numbers of type T . It is possible to generate streams of random integers, real or boolean values, by setting T to the appropriate datatype in a subclass.

It turns out that the basic ingredient for generating stream of pseudo-random numbers is a good uniform generator $RN(0, 1)$ over the interval $[0..1]$. We chose algorithm MRG32k3a from [L’Ecuyer, 1999], which is known to have long period and good statistical properties. We implemented different random variate generators using algorithms described in [Banks, 1998, Ch. 5]. These generators include the Uniform, Exponential, Gamma, Weibull, (Truncated) Normal, Bernoulli and Empirical distributions.

5 OUTPUT DATA ANALYSIS

Simulation results are sequences of observations for quantities of interest. Appropriate statistical techniques should be applied to analyze these sequences, as the observations are in general autocorrelated. libcppsim implements a set of classes dealing with collection of statistics; the relevant portion of the class diagram is shown in Fig. 3.

All types of statistical variables inherit from the `var<Tin, Tout>` abstract base class. This class represents “variables” which can be updated many times by feeding values of type T_{in} , and compute some function from the input data producing a result of type T_{out} . Each variable has two pure virtual methods: `update()` and `reset()`.

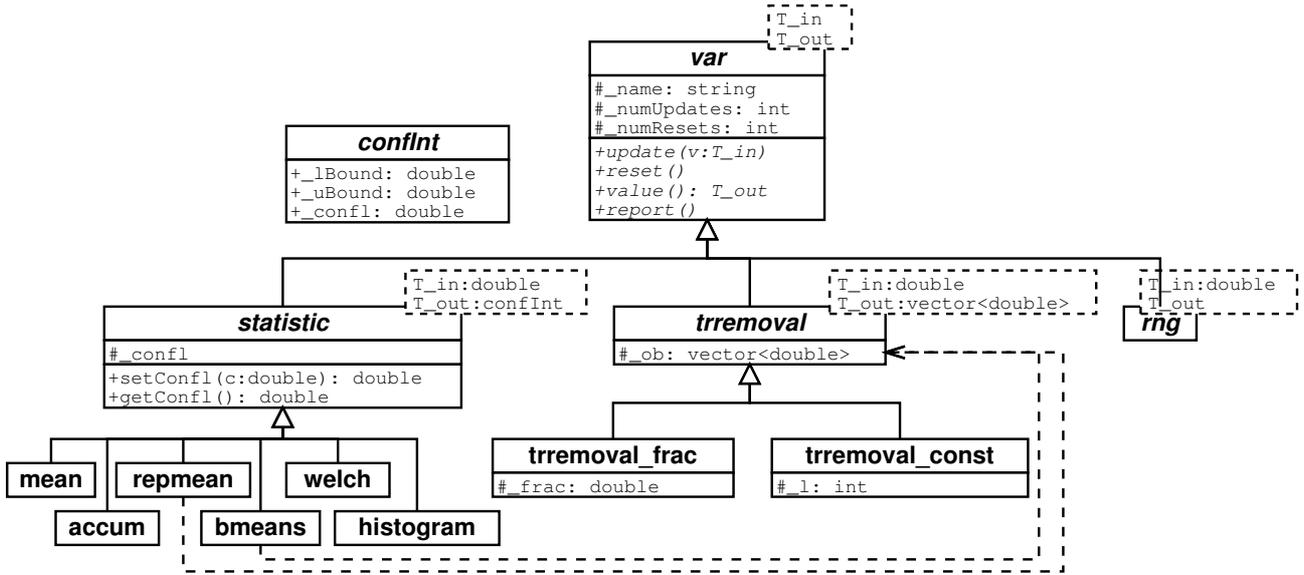


Figure 3: libcppsim Statistics Class Diagram

These are used to insert a new value and reset the variable to a known state. The result can be computed by invoking the `value()` method. Variables are not stateless; thus, it is possible that successive repeated invocations of the `value()` method return different results. This is useful for representing a random stream as a variable which is initially updated with the seed of the pseudo-random number generator, and whose `value()` method returns the next pseudo-random number from the stream, at the same time updating the seeds. Class `var` defines also a pure abstract `report()` method which can be used to display a report about the content of the class.

The statistics class represents a base class for a number of predefined statistic functions. In addition to the methods and attributes defined in its parent class `var`, statistics contains an attribute representing the confidence level of the computed result. Statistics are special kind of variables returning objects of type `confInt` (confidence intervals). The following statistics can be computed:

mean This class is used to compute a confidence interval for the mean of a sequence of *statistically uncorrelated* observations.

accum This class is used to compute the time-weighted sum of observations Y_1, Y_2, \dots, Y_n , $n > 1$ with timestamps respectively t_1, t_2, \dots, t_n . The result is $A = \sum_{i=1}^{n-1} Y_i(t_{i+1} - t_i)$

repmean Computes the mean of a sequence of observations using the method of independent replications. The simulation must be repeated R times, each run using different random streams and independently chosen initial conditions. Averages across replications can be used to obtain an accurate estimator of the sample mean [Banks, 1998].

bmeans The method of *batch means* divides the output data from one replication into a number of batches,

and then treating the means of these batches as if they were independent to compute the sample mean [Banks et al., 2001].

welch This class implements the graphical procedure of Welch [Welch, 1983] for identifying the length of the initial warm-up period. Values from R independent replications are collected. The values across different replications are first averaged and then smoothed by computing moving averages. The plot of the moving average becomes approximately constant after the warm-up period is over.

histogram This class computes an histogram profile from a sequence of values. The user supplies an expected lower bound and upper bound for the observed values, and the number of cells (bins) in which the histogram will be divided.

The `trremoval` class is used to model algorithms used for removing the initialization bias from a sequence of observations. The simulation run produces the sequence of observations Y_1, Y_2, \dots, Y_n for some parameter of interest. Using the whole sequence to compute the statistics is often dangerous, as there is a bias associated to artificial or arbitrary initial conditions. One method to overcome this limitation is to divide the sequence of observations into two phases: an initialization phase which includes observations Y_1, Y_2, \dots, Y_d , followed by a data-collection phase $Y_{d+1}, Y_{d+2}, \dots, Y_n$. Observations from the initialization phase are discarded, and only those in the data-collection phase are used for computing the required statistics. The parameter d , $0 < d \ll n$ is called *truncation point*.

Classes inheriting from `trremoval` are used to estimate the truncation point d . Currently, there is no automatic, general and correct method for detecting the length of the

```

typedef handle<process> handle_p;

class process :
  public coroutine, public shared {
public:
  process( const string& name );
  virtual ~process( );
  //// Modifiers ////
  // put in front of the sqs
  void activate( void );
  // activate AFTER q
  void activateAfter( handle_p& q );
  // activate BEFORE q
  void activateBefore( handle_p& q );
  // activate at time t
  void activateAt( double t );
  // activate with delay dt
  void activateDelay( double dt = 0.0 );
  void reactivate( void );
  void reactivateAfter( handle_p& q );
  void reactivateBefore( handle_p& q );
  void reactivateAt( double t );
  void reactivateDelay( double dt = 0.0 );
  // remove from the sqs
  void cancel( void );
  //// Accessors (const) ////
  // Is this process idle?
  bool idle( void ) const;
  // Is this process terminated?
  bool terminated( void ) const;
  // The next reactivation time
  double evTime( void ) const;
protected:
  //// Modifiers ////
  void hold( double dt );
  void passivate( void );
  void stop_simulation( void );
  //// Accessors ////
  // Currently executing process
  handle_p& current( void ) const;
  // Current simulation time
  double time( void ) const;
  // Simulation process' body
  virtual void inner_body( void ) = 0;
};

```

Figure 2: Interface of the process Class.

initialization bias. Different approaches have been proposed in the literature [White Jr. et al., 2000], although their effectiveness has not been agreed upon. For that reason, simulation packages usually implement the strategy of removing a prefix of fixed length (eg, 20%) from the sequence of observations; this approach is implemented by class `trremoval_frac`. This class can be used if the exact length of the warm-up period is known by the modeler.

6 EXAMPLE

We show in Fig. 4 a simple usage example of the `libcpps` library. The `main()` function performs a set of initialization tasks. The first is the creation of a new simulation context with an associated SQS object which is used by all processes. As in the original SIMULA language, it is possible to nest simulation contexts, allowing independent simulations to be executed inside other sim-

ulations. Once a simulation context has been created, it is possible to instantiate and schedule for activation one or more simulation processes.

```

#include <iostream>
#include "cpps.h"

class job : public process {
public:
  job( const string& name ) :
    process( name ) {
    cout << "Created job '"
         << name << "' " << id() << endl;
  };
  virtual ~job( ) {
    cout << "Destroyed job '"
         << name() << "' " << id() << endl;
  };
protected:
  void inner_body( void ) {
    hold( 10 );
  };
};

class source : public process {
public:
  source( const string& name ) :
    process( name ) { };
  virtual ~source( ) { };
protected:
  void inner_body( void ) {
    handle<job> j;
    for ( int i=0; i<10; i++ ) {
      hold( 5 );
      j = new job( "job" );
      j->activateAfter( current() );
    }
    stop_simulation();
  };
};

int main( void )
{
  simulation* sim = simulation::instance();
  // Instantiates a simulation context
  sim->begin_simulation( new sqsDll() );
  // Creates a simulation process
  handle<source> src = new source( "source" );
  // Schedules the process for execution
  src->activate();
  // Runs the simulation
  sim->run();
  // Cleans up the SQS
  sim->end_simulation();
  return 0;
};

```

Figure 4: Usage Example of the `libcpps` Library

7 CONCLUSIONS

In this paper we described a SIMULA-like process-oriented simulation library implemented in C++. The library is based on the coroutine abstraction implemented as a C++ class, and has been successfully compiled on various versions of the Linux OS and other flavors of Unix including Digital/Compaq OSF and NetBSD. The library provides facilities such as simulation processes, random variate generators and basic statistical functions. The `libcpps` library has been used to implement UML- Ψ , a process-oriented simulation tool for performance evaluation of UML specifications [Balsamo and

Marzolla, 2003; Marzolla, 2004].

Due to its object-oriented nature, the library can be easily extended with additional functionalities. We are currently improving the library in several directions. First, we are implementing more high-level synchronization mechanisms (such as semaphores and mailboxes) which can be useful for developing certain complex simulation models. We are also implementing more statistical functions in order to provide the modeler with a set of more advanced mechanisms for evaluation of the simulation results.

Availability

Full source code of the `libcpsim` simulation library is available on the author's web page at <http://www.dsi.unive.it/~marzolla>

Acknowledgments

This work has been partially supported by MIUR research project FIRB "Performance Evaluation of Complex Systems: Techniques, Methodologies and Tools".

REFERENCES

- Balsamo, S. and M. Marzolla (2003, June 9–11). Simulation modeling of UML software architectures. In D. Al-Dabass (Ed.), *Proc. ESM'03, the 17th European Simulation Multiconference*, Nottingham, UK, pp. 562–567. SCS–European Publishing House.
- Banks, J. (Ed.) (1998). *Handbook of Simulation*. Wiley–Interscience.
- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol (2001). *Discrete-Event System Simulation*. Prentice Hall.
- Birtwistle, G. (1979). *DEMOS—A system for discrete event modelling on Simula*. MacMillan Press.
- Brown, R. (1988, October). Calendar queues: A fast 0(1) priority queue implementation for the simulation event set problem. *Comm. of the ACM* 31(10), 1220–1227.
- Dahl, O.-J. and K. Nygaard (1966, September). SIMULA—an ALGOL-based simulation language. *Comm. of the ACM* 9(9), 671–678.
- Darmont, J. (2000, January). DESP-C++: a discrete-event simulation package for C++. *Software–Practice and Experience* 30(1), 37–60.
- Engelschall, R. S. (2000, June 18–23). Portable multithreading—the signal stack trick for user-space thread creation. In *Proc. USENIX Ann. Tech. Conference*, San Diego, CA.
- Helsgaun, K. (1999). A portable C++ library for coroutine sequencing. *Datalogiske Skrifter (Writings on Computer Science)* 87, Roskilde University, Denmark.
- Hlavička, J., S. Racek, and P. Herout (1999, December). C-Sim v.4.1. Research Report DC-99-09, DCSE CTU Prague Publishing, Prague.
- L'Ecuyer, P. (1999, January). Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* 47, 159–164.
- Little, M. C. JavaSim Home page. <http://javasim.ncl.ac.uk/>.
- Little, M. C. and D. L. McCue (1993, July). Construction and use of a simulation package in C++. Computing Science Tech. Report 437, University of Newcastle upon Tyne.
- Marlin, C. D. (1980). *Coroutines: A Programming Methodology, a Language Design and an Implementation*, Volume 95 of LNCS. Springer-Verlag.
- Marzolla, M. (2004, February). *Simulation-Based Performance Modeling of UML Software Architectures*. PhD thesis, TD-2004-1, Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy.
- Schwetman, H. (2001). CSIM19: a powerful tool for building system models. In *Proc. 33rd Winter Simulation Conference*, pp. 250–255. IEEE Computer Society.
- Stroustrup, B. (1997). *The C++ Programming Language* (3rd ed.). Addison–Wesley.
- Varga, A. (2001, June 6–9). The OMNET++ discrete event simulation system. In *Proc. ESM'01, the 15th European Simulation Multiconference*, Prague, Czech Republic, pp. 319–324. SCS–European Publishing House.
- Welch, P. D. (1983). The statistical analysis of simulation results. In S. Lavenberg (Ed.), *The Computer Performance Modeling Handbook*, pp. 268–328. Academic Press.
- White Jr., K. P., M. J. Cobb, and S. C. Spratt (2000). A comparison of five steady-state truncation heuristics for simulation. In *Proc. 32nd Winter Simulation Conference*, Orlando, FL, pp. 755–760. SCS International.

AUTHOR BIOGRAPHY



MORENO MARZOLLA received MSc and PhD degrees in Computer Science in 1998 and 2004 respectively, both from "Ca' Foscari" University of Venice. From 1999 to 2001 he was a software engineer at the Italian National Institute for Nuclear Physics. He currently holds a post-doc position at the University of Venice. His research interests include performance evaluation of software systems, distributed systems monitoring and parallel and distributed simulation. His web page is at <http://www.dsi.unive.it/~marzolla>

PARAMETER TUNING IN MODELLING HUMAN BEHAVIOURS BY USING OPTIMIZATION TECHNIQUES

Agostino G. Bruzzone
McLeod Institute of Simulation Science
Via Opera Pia 15, 16145 Genova, Italy
agostino@itim.unige.it
www.simulationscience.org

Marina Massei, Simone Simeoni, Danila Carini
Liophant Simulation
Via Molinero 1, 17100 Savona, Italy
{*marina.massei, simone.simeoni, danila.carini*}@liophant.org
www.liophant.org

Matteo Brandolini
BRB Studio
Palazzina Uffici, Nuovo Porto di Voltri, 16100 Genova Voltri, Italy
matteo.brandolini@brbstudio.com
www.brbstudio.com

ABSTRACT

The authors propose an approach for developing complex models integrating different components and including human behaviour; in these problems the data uncertainty as well as their availability is a major criticality; the paper proposes an approach for a mutual validation of the model and a fine tuning process based on integrating simulation with optimisation techniques.

INTRODUCTION

Modelling human interactions is a very challenging problems especially if it have to be faced in relation to other issues such as technical/operative aspects (i.e. modelling team working within a production cell or simulating a group reactions in a battlefield). The paper proposes the issue of modelling human behaviours with special attention to their interactions while working/operating in teams; the complexity of such models are related to the multidisciplinary aspects involved. The knowledge of these processes as well as the quantitative data are very difficult to be caught and normally are affect by high degrees of uncertainty.

Currently most of these aspects are modelled in existing simulators as component based on estimations provided by subject matter experts (SME); therefore a proper VV&A (Verification, Validation and Accreditation) of the resulting simulators is complicated by individual estimations, complexity in reproduction of boundary conditions and difficulties in quantitative measuring of both input/output affecting such aspects. Considering the human behaviour just as an object model to be integrated in a wider simulation the stochastic nature of several external factors is a further complexity element for proper design and testing of this components.

However human behaviour and team working performances are often affecting overall performance and some kind of modelling could be requested for constructing a correct model (i.e. fear effect on troops attacking under enemy fire or people efficiency in a company just acquired and subjected to personnel reduction procedure). In particular considering industrial plants, production facility or business process it is evident that the introduction of this aspects could be pretty useful, while it is very important to avoid the introduction of just noise factors in simulation when it is impossible to proceed to their proper identification, definition, measure and testing. The paper will concentrate in proposing a general methodology and a more specific approach for proceeding in modelling in this area of interest.

GENERAL APPROACH FOR COMPLEX HUMAN BEHAVIOUR MODELS

The first general question to be solved is the evaluation of the importance of Human component modelling in the real system and the fidelity level requested. A good example to be used for clarify this aspect could be related to the case of modelling a production facility; in this experience is often requested to model the pause/coffees of the workers; in this case we can adopt different approaches such as:

- § Considering just net available time of the workers without break times
- § Generating events related to the coffee breaks randomly for the same amount
- § Construct a model that consider the aggregation phenomena and chatting aspects

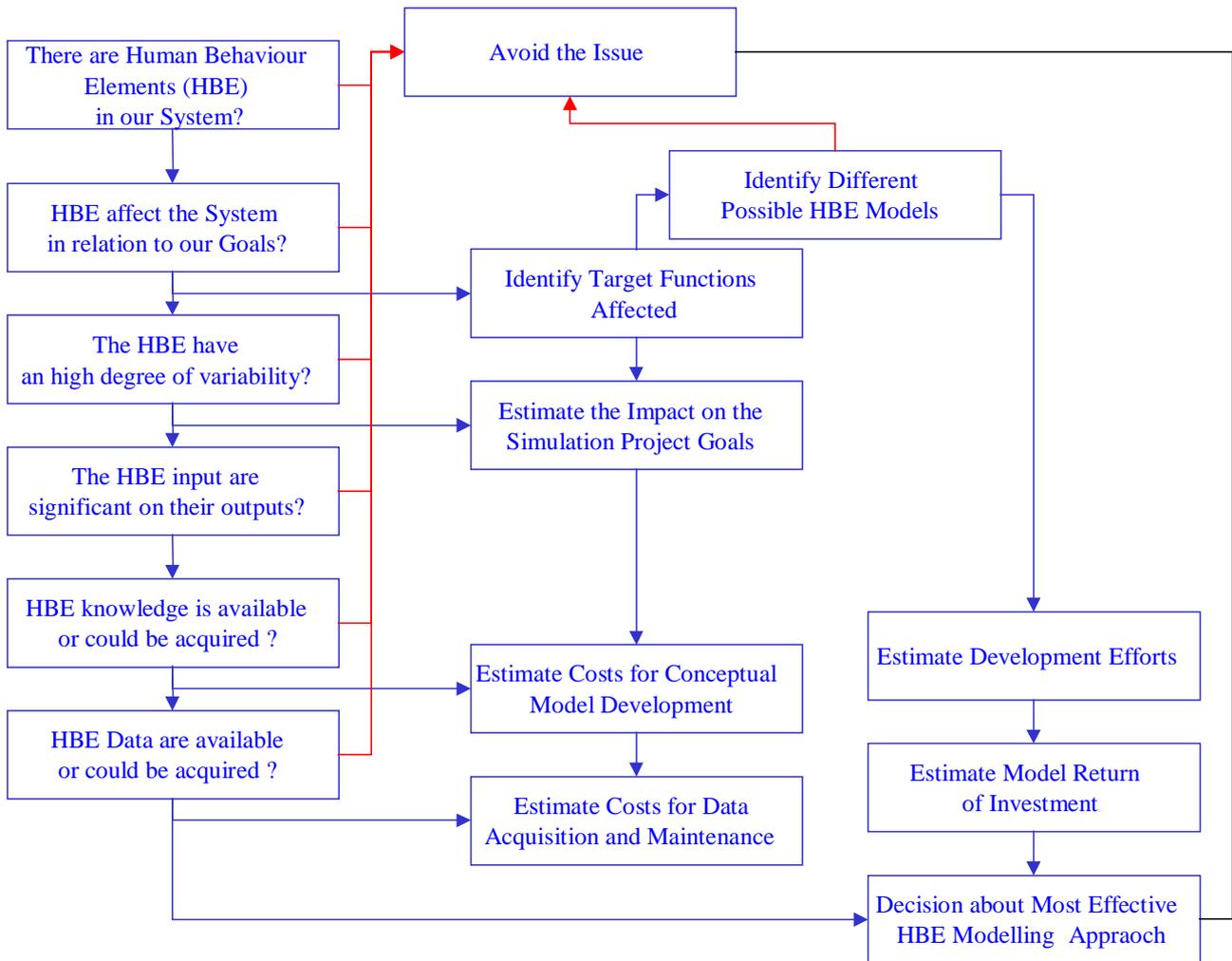


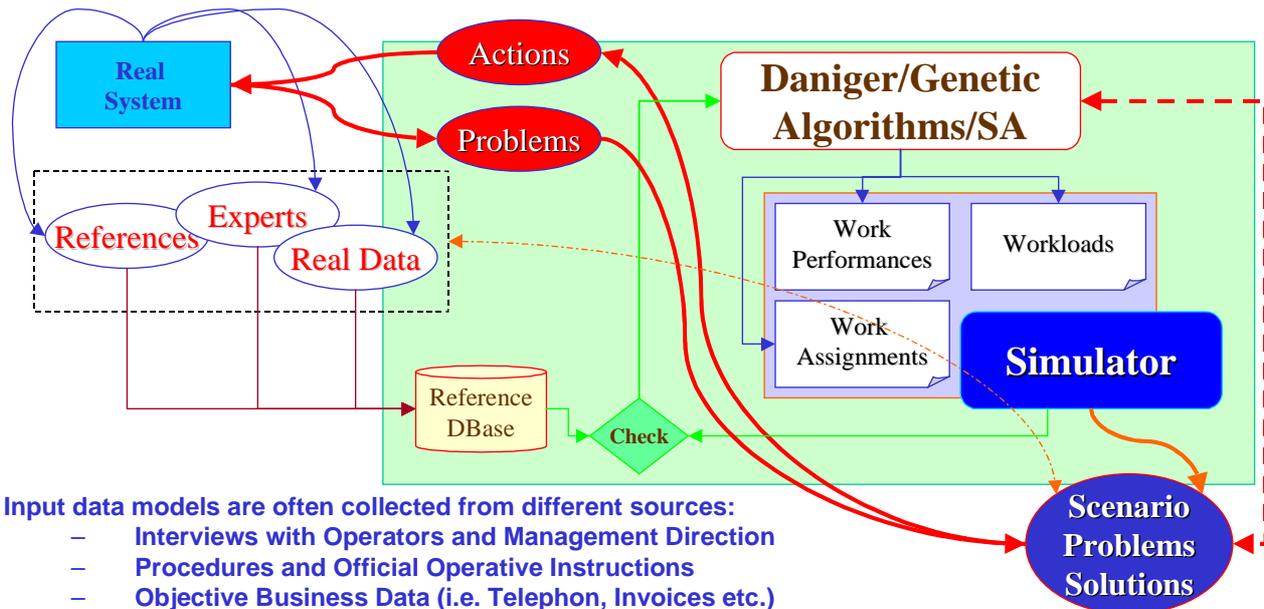
Figure 1: Procedure for Evaluating Best Modelling Approach related to Human Behaviour Elements

The approach to be chosen should be based on the benefit it provides to the usability of the model and to the profitability of the results provided; so it is strictly related to the context (i.e. in a regular facility it could be a simple approximation, while in a reality where there are high rate of absenteeism this factor could require additional attention). Another important aspect it is to identify how much this components affect the overall performance. Therefore the critical point it is to estimate the costs and benefits related to each model; as much as we detail this component as much as additional knowledge and data are requested to feed the model this normally provide additional development time, additional testing time and especially higher costs for maintaining the model in the future. In effect it could be critical to develop a very effective model and expend a lot of resources in his VV&A, if just after one year being changed procedures, people or other boundary conditions the human behaviour model become obsolete and requires another big effort; in this case it could be much better to approach the case by developing a simplified model that takes into account just parameters that could be update with reasonable efforts. In general the overall procedure it is summarised by flow chart in figure 1. If we consider to operate in an application case

where human factors are significant and needs to be integrated with overall processes and to create a detailed model component the suggestions proposed by this approach is to proceed through a "reverse engineering" problem in best fitting the relative parameters for the model.

PARAMETER TUNING

Often the human behaviour component can be developed based on available experiences and bibliography, however a critical aspect is always connected with the fine tuning of the factors affecting each model (i.e. people attitudes, psychological indexes, etc.). Obviously simulation is a wonderful approach for estimating the consequence of some configuration, while it is pretty difficult to identify "a priori" the proper set of input that represent our system. Due to these consideration a possible approach is to investigate a specific scenario where several overall performance is available and to proceed by "reverse engineering" in the identification of the proper configuration that respect such result.



In order to do a correct reverse tuning it was used a reverse engineering technique for critical model parameters (i.e. *Frequency and Operations Time/Performance and Tasks Assignment*) as for reference data (*Balance, timetable cards, Objective Data*)

Figure 2: Overall Best Fitting Procedure for Human Resource Modelling in BPR application

For example if we have the case related to the business processes in a company we can try to set the psychological factors and human parameters of the available personnel trying to reproduce the last year overall behaviour based on the knowledge of general data (working hours, effective work carried out, costs, extra time, success rate in different business processes etc.).

Obviously these parameters involves a very large number of factors; to provide an example a company reorganization involving administrative and commercial offices introducing these aspects could require to tune over 300 parameters for just a small department of 20 people (i.e. different operation service time identification, work sharing configuration, etc).

Therefore the best approach is to adopt some automated procedure based on optimisation that drive the simulation input in order to obtain a proper overlapping with the real value of overall target function on the reference scenario.

This approach could be obtained especially by techniques based on approaches that avoid gradient methodologies in favour of area or stochastic optimisation; in effect the stochastic nature of these phenomena introduces complexity even in the optimum definition.

The very complex relations among factors based on highly non linear relations involve risk of being engulfed in local minimum configurations.

Considering these aspects it is preferable the use of nongradient algorithms for optimisation of the best fitting procedure.

The detailed methodology for this tuning process could be based on the use of genetic algorithms and/or stochastic adaptation in order to face the high dimension of the analysis space and the complexity of the fitness functions; a general architecture is summarised in the figure 2.

APPLICATION EXAMPLES

The authors experienced different case where human behaviour had significant influence over the target functions, in the following some examples are summarized

Human Behaviour in Health Care

The organization of Surgery Divisions in an Hospital introduces different aspects related to human behaviour such as :

- Team Harmony
- Individual Attitudes

In the model developed these factors was introduced as characteristics of the single operator based on interview with the responsible of each Surgery Division; the simulator developed was used as reference for a planning system devoted to increase throughput and reduce costs with heavy constraints in term of resources and technical parameters.

The introduction of human factor it was possible to improve the efficiency of the system by combining in efficient way the resources guaranteeing best harmony

during surgery operations; the comparison obtained between simulated planning and effective operations was very satisfactory.

A similar approach was used to organise hospital Analysis Department where the different profiles was used to consider the behaviour and the interactions during diagnostic meetings among technicians and doctors.

Quality of Life in Vessels and Oil Platforms

Ships and vessels represents a challenge for guarantee efficient quality of life; today new developments in technologies, in automation and especially in system reliability allows to reduce/change processes and systems; in military ships this allows to reduce drastically personnel, increasing the living spaces and providing more support to other services (i.e. fuel, weapons etc.)

The authors was involved in several projects where simulation models was requested to estimate these benefits in projects related to ships and oil platforms.

One project was devoted to create a ship restaurant model for estimating the benefits for introducing new technologies (i.e. passing from kitchens to catering for fast ferries); it was necessary to consider the interference and behaviour of people during the meals in term of grouping attitudes in references to different cultural framework; in this context detailed micrologistics models was developed in order to reproduce the interference and interaction among people; the human attitude was based on bibliography experiments and introduced in the model, while validation on existing case study allowed to complete the best fitting of these parameters.

Public Facility Evacuation

The authors developed in the past models for studying crowd control especially during emergency; the study was applied to stations and museums in order to evaluate the effectiveness of emergency exits and evacuation procedures. In this context the human factor was mostly focusing on the people behaviour that is related to friendship, family structure and other relations keeping together groups and or requesting single individuals to search their relatives.

In this case we model the people as single individuals with specific attributes and types that define their behaviour and reactions to emergencies.

Business Process Re-engineering

Another interesting case study related to the development of integration of human behaviour components in a simulation model was recently developed for studying business reorganisation. In our case the human parameters was concentrating in two major aspects:

Team Working Performance Reaction to Reorganization

In effect the model was devoted to estimate the benefits of a business reorganization in a medium size company, so the impact of the process over the people and their attitude in reacting to the reengineering heavily affects the overall company performance.

In this case most of the parameters was provided by the human resource division and by interview with managers, parameter best fitting was achieved by automated optimization using as reference value people workload extracted from work tables recorded by the collection system respect the simulation estimation.

In this application case it was performed a double stages best fitting devoted to a general tuning followed by a more detailed fitting on the more significant factors.

Different Techniques was used for the best fitting and the Experimental results obtained allowed to complete the model validation as well as the checkup for this proposed approach. Some results about experimental results are proposed in figure 3 as example.

In effect the process is based on the integration between the simulator and the optimisation algorithms; currently the authors implemented, in reference to this BPR applications, a system working in Office Suite™ using different optimisation techniques (i.e. Genetic Algorithms, Random Search, Stochastic Adaptive Search and custom systems); therefore in addition to regular optimisation techniques a customised approach developed from "branch and bound" was used in order to try to create an ad hoc robust system: Tiger DNL is in effect an Optimisation techniques implemented for comparing different approaches devoted to identify cluster of input for proper tuning of the parameters that is inspired by Tiger life cycle.

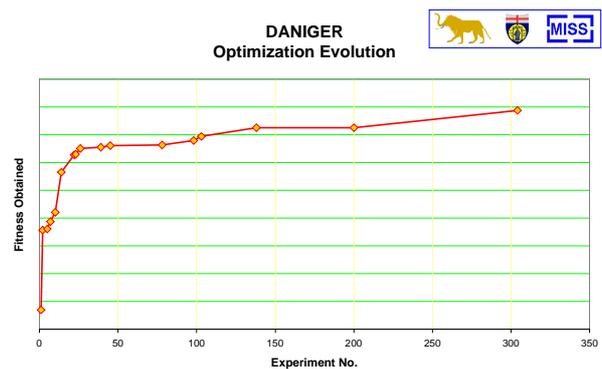


Figure 3: Fitness Function Obtained during Best Fitting

The fitness function for automated optimisation is based on output of simulation runs operating on a scenario with over 150 variables to be tuned representing human factors that affect the work efficiency of administrative department people.

In effect the target function for best fitting is based on the measure of the sum square error between reference target functions over pre-existing scenario and simulated data.

CONCLUSIONS

In human behaviour related to industrial facilities and business process it is very critical the proper identification of the final goal of simulation project; the final objectives become in effect the reference baseline for defining the fidelity and detail level for these aspects considering the benefits obtainable by including these components respect simplified modelling.

The factors to be considered in order to estimated the benefits are maintainability, usability and effectiveness. Obviously a very critical aspect is related to the possibility to tune the human parameters considering the complexity of measuring and validating these aspects; the use of best fitting procedures integrating optimisation techniques within the simulation and using reference baseline on quantitative measure related to historical scenario is the innovative methodology proposed; the approach is very promising and effective and allows to complete robust validation and verification.

The example proposed confirm that the human factors are often introduced in model in explicit way, and that the benefits obtainable are significant.

REFERENCES

- § Amico V., Guha R., Bruzzone A. G. (2000) "Critical Issues in Simulation", Proceedings of Summer Computer Simulation Conference, Vancouver, July
- § Bara B. G., (1995), "Cognitive Science: A Developmental Approach To The Simulation Of The Mind", Lawrence Erlbaum Associates Ltd
- § Berkowitz L. (2000) "Causes and Consequences of Feelings", Cambridge Press, Cambridge
- § Bruzzone A. & Signorile G., (1999) "Crowd Control Simulation in Java Based Environment", Proceeding of Websim99, San Francisco, January
- § Bruzzone A., Revetria R., (2003) "VV&A for innovative concept design in new vessels simulation and Virtual Prototypes", Proceedings of SCSC 2003, Montreal, July 20-24
- § Bruzzone A.G. (1995) "Adaptive Decision Support Techniques and Integrated Simulation as a Tool for Industrial Reorganisation", Proc. of ESS95, Erlangen, October 26-28
- § Bruzzone A.G. (1996) "Object Oriented Modelling to Study Individual Human Behaviour in the Work Environment: a Medical Testing Laboratory ", Proc. of WMC'96, San Diego, January
- § Bruzzone A.G. (1996), "Smart Integrated Interface for Evaluating and Supporting Human Performances", ESPRIT ITBRWG 8467 Simulation for the Future: New Concepts, Tools and Application (SiE-WG), January 29, Brussels
- § Bruzzone A.G., Mosca R., Brandolini M., Briano C., Spirito F., Coppa A. C. M. (2001) "Advanced Modeling for Customer Behaviour Analysis ", Proc. of MIC2001, Innsbruck Austria, February
- § Bruzzone A.G., Mosca R., Briano C., Brandolini M. (2000) "Models for Supporting Customer Satisfaction in Retail Chains", Proceedings of HMS2000, Portofino, October 5-7
- § Bruzzone A.G., Mosca R., Revetria R. (2002) "Supply Chain Management Dynamic Negotiation using Web Integrated Logistics Designer (WILD II)", Proceedings of MAS2002, Bergoggi October
- § Bruzzone A.G., Rivarolo D. (1997) "Virtual Reality as the Starting Point for the Functional Analysis of Public Plant Design Services", Proceedings of High Performance Computing, HPC'97, Atlanta, Georgia,
- § Giribone P., Bruzzone A.G. (1994) "Man Made Disasters - Chemical Plant Accidents", Proceedings of SEM94, San Diego, April 10-15
- § Giribone P., Bruzzone A.G., Devoti A. (1998) "Combined Simulation of the Internal Transportation in Medical Environment", Proceedings of Medical Sciences Simulation Conference, San Diego, January 11-24
- § Harrington H. J., Tumay K., (2000), "Simulation Modeling Methods: To Reduce Risks and Increase Performance", McGraw-Hill, New York
- § Klaus G.N., Troitzsch K. G., (1999), "Simulation for the Social Scientist", Open University Press,
- § Lord R. G., Klimoski R. J., Kanfer R., (2002), "Emotions in the Workplace : Understanding the Structure and Role of Emotions in Organizational Behavior", Jossey-Bass, San Francisco
- § Mac Leod D., (1995), "The Ergonomics Edge: Improving Safety, Quality, and Productivity", John Wiley & Sons, Inc., New York
- § Mosca R., Bruzzone A.G. & Costa S. (1996) "Simulation as a Support for Training Personnel in Security Procedures", Proc. of Military Government and Aerospace Simulation, New Orleans LA, April
- § Mosca R., Giribone P. & Bruzzone A.G. (1995) "AI Techniques to Optimize the distribution of Fire-Fighting Squads during an Accident", Proc. of Simulators International XII, Phoenix, April 9-13
- § Mosca R., Giribone P. & Bruzzone A.G. (1995) "Sensitivity Analysis In Interactive Stochastic Simulators", Proceedings of MIC95, Innsbruck,
- § Rasmussen J., Pejtersen A. M., Goodstein L.P., (1994), "Cognitive Systems Engineering", John Wiley & Sons, Inc., New York
- § Reason J., (1990), "Human Error", Cambridge University Press, Cambridge
- § Roger C., Schank R. C., (1997), "Virtual Learning: A Revolutionary Approach to Building a Highly Skilled Workforce", McGraw-Hill, New York
- § Sarason I.C., Spielberg C.D.(1989), "Stress and Anxiety", Hemisphere Publishing Co., Washington
- § Watts D. J. (1999) "Small worlds" Princeton University Press, Princeton

BIOGRAPHIES

Agostino G. Bruzzone is full Professor in the Department of Production Engineering at the University of Genoa where he specialises in the field of simulator-based applications for industrial plants, developing new methodologies and intelligent system integration techniques.

Currently, he is serving as Industrial Relations Chair in SCS Europe and as Director of Technical Council for Management and Forecasts Application in the Board of Directors of the Society for Modelling & Simulation International.

He is Director of the McLeod Institute for Simulation Science involving over 20 centers worldwide.

He is also a founding member and the president of the Liophant Simulation Club. He has written more than 100 scientific papers in addition to numerous technical and professional reports in partnerships with major companies (i.e. IBM, Fiat Group, Contship, Solvay) and agencies (i.e. Italian Navy, NASA, National Center for Simulation, US Army).

Matteo Brandolini got his degree in Management Engineering and Logistics Engineering at the University of Genoa. He is a founding member of the Liophant Simulation Club and has been involved in several simulation projects related to military and industrial businesses. Presently, he is working as consultant in the fields of Port Environment, Professional Training, Retail Reorganization, ERP projects, E-Commerce and Project Management with major Italian companies. Matteo Brandolini is currently senior partner of BRB Studio a consulting firm devoted to apply advanced modelling and simulation techniques in different sectors.

Danila Carini obtained her degree in Management Engineering at University of Genoa with a thesis on “BPR Organizational Modelling in SME”. She had professional experiences based on Research Grants for Developing Business Process Simulators in Universidad Panamericana, Mexico City And for visiting simulation facilities involved in Human Behaviour Modelling at the ICT, University of Southern California. She acted as Marketing Analysis Project Coordinator of an International Team composed by Engineering & MBA Students in Vernazza ltd during IEPAL International Program.

Marina Massei completed her thesis in Genoa University on File Structures & Data Warehousing. She operates with the team of Prof. Mosca and Prof. Bruzzone, DIP University of Genoa as project controller. She participated in the organization of SIREN Courses, HMS & MAS International Workshops and IEPAL Meetings. She is involved in the organization of International Scientific Events (i.e. SCSC2003, Montreal and SCSC2004 San Jose` CA) and in the coordination of a Simulation Technical Council specialized in Advanced Techniques for Management and Forecasts (i.e. SIMPLEST).

Simone Simeoni completed the Management Engineering degree in Genoa University working in cooperation with Ansaldo Energia. He experienced visits to major companies in Simulation field in Europe, North America and Asia; he was active in CAE Montreal and Ford Motor Company in Detroit. He is currently involved as Project Manager representative in over 50 advanced projects in a billion dollars company operating as member of DIP special research team.

VISUALISATION AND EDUCATION

DEBRIS DISPERSION MODEL USING JAVA3D

Rajkumar Thirumalainambi, Ph.D.
SAIC @ NASA Ames Research center
Mail Stop 269-2, Moffett Field
California, USA 94035
rajkumar@mail.arc.nasa.gov

Jorge Bardina, Ph.D.
NASA Ames Research center
Mail Stop 269-2, Moffett Field
California, USA 94035
Jorge.E.Bardina@nasa.gov

KEYWORDS

Orbital dynamics, Debris Dispersion,
Visualization in 3-dimensional, Java3D

ABSTRACT

This paper focused in web-based simulation of Shuttle launch operations and debris dispersion. Java 3D graphics provides geometric and visual content with suitable mathematical model and behaviors of Shuttle launch. Because the model is so heterogeneous and interrelated with various factors, 3D graphics combined with physical models provides mechanisms to understand the complexity of launch and range operations. The main focus in the modeling and simulation covers orbital dynamics and range safety. Range safety areas include destruct limit lines, telemetry and tracking and population risk near range. If there is an explosion of Shuttle during launch, debris dispersion is explained. The shuttle launch and range operations in this paper are discussed based on the operations from Kennedy Space Center, Florida, USA.

INTRODUCTION

With the complexity of operations occurring in launch and range in present and future spaceports, simulation technologies will be critical to train staff and develop proper procedures and to understand complexity. As part of this focus area, advanced simulation technologies would be developed that accurately represent the performance of Shuttle launch and range. The primary objective is to simulate Shuttle launch and range operations in a distributed collaborative virtual environment with capabilities of command and control with suitable visualization and rendering. This initiative is focused on four primary focus areas (Bardina and Rajkumar 2003): (i) weather modeling (ii) tracking (trajectory) and telemetry (Jensen et. al 1962) (iii) range safety (toxic gas dispersion, debris dispersion, human health risk

assessment) (iv) decision modeling for process operations (Shuttle discrete/continuous event model). The weather modeling represents an integrated suite of weather models, real-time monitored data from various agencies and decision making capabilities. The tracking and telemetry technologies focus area will develop advanced capabilities compatible with emerging spacecraft designs. These capabilities would provide low cost, highly reliable and accurate surveillance and tracking systems. The range safety and traffic management focus area would develop a distributed electronic data architecture that would enable a higher level of integration of range information models. The range dispersion modeling system would install state-of-the-art equipment to identify specific chemicals, depict plume volumes and process in real time. With the wide array of processes involved in preparing spacecraft and payloads for space flight, simulation technology would prove to be critical in optimizing processes in spaceport operations. These simulation technologies may range from discrete event (DE) simulators from modeling payload processing tasks to continuous event (CE) simulators utilizing computational fluid dynamics (CFD) to simulate the flow properties of cryogenic fluids. This focus area would utilize discrete and continuous event simulations to increase cycles of learning, improve efficiency, and reduce costs. This focus area seeks to develop the capability to simulate all of the processes involved in launch operation. The simulation technologies help reduce the operational costs of launch since the learning cycles occur more quickly and at a lower cost than training with the actual hardware.

Simulation would develop new technologies and models that reduce the conservatism embedded in operational models and guidelines, while providing the accuracy necessary to ensure safe and cost effective launch operations. The present range decision models need upgrading to ensure technology limitations do not interject

excessive conservatism into decision making. Excess conservatism could result in unnecessary delays or postponement of launches or operational activities.

LAUNCH AND RANGE OPERATIONS

Range safety personnel evaluate vehicle design, manufacture, and installation prior to launch; monitor vehicle and environmental conditions during countdown; monitor the track of vehicles during flight; and, if necessary, terminate the flight of malfunctioning vehicles (NASA 1988; NRC 2000). The method used for flight termination depends on the vehicle, the stage of flight, and other circumstances of the failure. In all cases, propulsion is terminated. In addition, the vehicle may be destroyed to disperse propellants before surface impact, or it may be kept intact to minimize the dispersion of solid debris. Flight termination can also be initiated automatically by a break-wire or lanyard pull on the vehicle if there is a premature stage separation (FAA 1999). This section discusses requirements for flight termination, tracking, and telemetry and examines reliability. Impact limit lines (ILL) define the areas to be protected. ILLs are drawn around populated areas to protect them from falling debris. Flight rules specify the minimum distance from each land mass to which falling debris may approach. Two impact limit lines are drawn around the Kennedy Space Center and they are shown in figure 1 (NASA 1988).

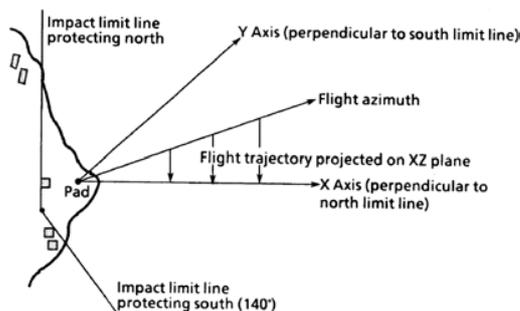


Figure 1 Impact limit line

The northern line points directly north. The southern line points southeast at an azimuth of 140°. The two lines intersect at a point southwest of the launch area. The vector lying on the horizontal plane that is perpendicular to the northern line with its origin at the pad is called the X-axis. The similar line perpendicular

to the southern line is called the Y-axis. The vertical line pointing up from the pad is called the Z-axis. The planes formed by these lines are called the XY plane or tangent plane and the XZ and YZ planes or vertical planes.

Destruct lines shown in figure 2 provide the criteria for terminating flight (NASA 1988). In general, a vehicle violating a destruct line is subject to termination by the Range Safety Officer. During ascent, the impact point of debris will be well forward of the direction of the Shuttle's last motion. Any deviation outside this limit indicates that the Shuttle is behaving in an abnormal, though not necessarily dangerous fashion. A normal performance envelope will include three times the standard deviation on either side of the nominal trajectory.

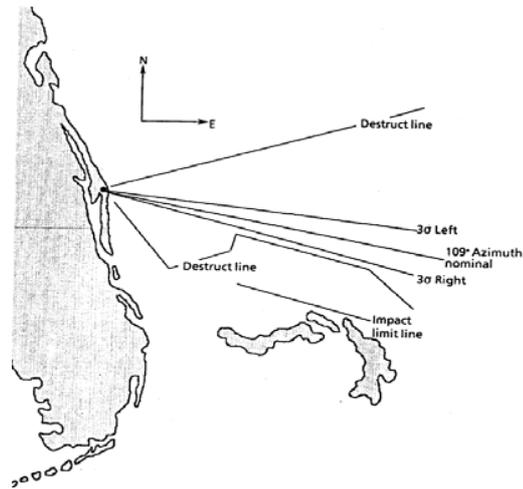


Figure 2 Destruct line

The destruct limit lines are constructed both on the vertical plane and on the impact lines in the horizontal plane. To construct a destruct line and determine debris scattering, atmospheric drag, effect of local winds, aerodynamics of debris pieces, system delays, Coriolis force and Δv imparted to the pieces of explosion are considered (Baskett and Pace 1995). The wind profile is based on the dispersed monthly wind and is assumed to be in the direction of the most critical impact limit line. Measuring from the pad, the drag impact range for each piece is placed along a line in the downrange direction. From each of these points, a semi-circle is drawn plus 90 percent of the worst possible wind. The object of this calculation is to determine how close to the impact line the farthest scattered piece of debris will land. If the result is either

beyond or before the impact line, the calculation is redone using a different direction for velocity until a velocity is found that will place the piece precisely on the line. This velocity vector becomes the slope of the tangent to the destruct line at this particular point on the range. When all the points have been processed, a curve fit is performed using the slopes to connect points from arc to arc of equal arc length (in figure 3) (NASA 1988).

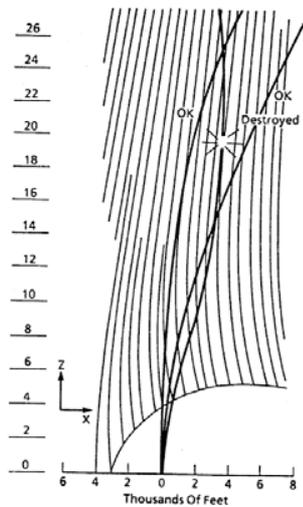


Figure 3 Vertical plane destruct line

When any of these points fall on the destruct line, the decision to destroy shuttle must be made so as to protect critical areas. Drag corrected (Figure 4) vehicle velocity and dispersion techniques are used to construct chevron destruct lines. Along with drag information, the maximum impact dispersion area is calculated for each piece of debris.

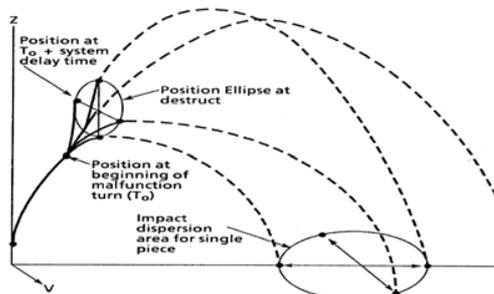


Figure 4 Drag correction

The chevron lines (Figure 5) indicate horizontal destruct lines. An impact point down range, called the vacuum impact point (VAC IP) is

calculated by assuming that all thrusting stops at the point of explosion and the vehicle flies ballistically as though there were no atmosphere. Several deviant velocity directions are chosen based on maximum turning rate and the vehicle is assumed destroyed several seconds later at each of the resulting points.

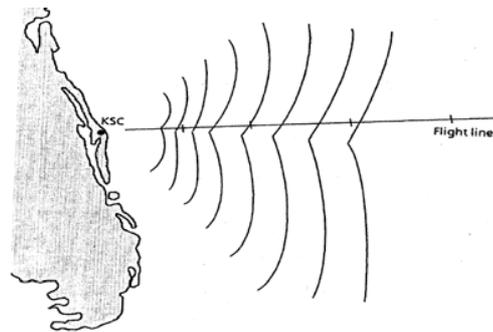


Figure 5 Chevron lines

For each debris piece, the calculated impact points for the various deviant velocity directions form an impact dispersion area on the ground. Chevron lines are produced for a series of velocities. As the flight progresses, the chevron lines move downrange. The chevron lines are constructed with a 3σ dispersion envelope factored. This allows a “slow” vehicle to stay ahead of chevron lines. It must be reemphasized that destruction action is taken only if the Range Safety Officer observes a violation on both vertical and horizontal deviation. The destruct limit lines, chevron lines, down range safety simulations, and nominal trajectory simulations are computed using various physical models and disseminated through the web using Java 3D API. The Java 3D model allows the user to simulate an infinite number of solutions of debris dispersion as well as flight trajectories. If the Range Safety Officer decides to abort the mission, the model simulates the debris dispersion; so that a catastrophe may be avoided. The following section outlines the details of the Java 3D Model which provides visualization rendering coupled with a physical model.

JAVA 3D – MODEL

The scene graph consists of superstructure components, a *VirtualUniverse* object and a *Locale* object, and a set of branch graphs (Brown and Petersen 1999; Davidson; and Selman 2002). Each branch graph is a subgraph that is rooted by a *BranchGroup* node that is attached to the

If the rocket orbits in a perfect trajectory and corresponding α reaches 1.0, the satellite orbit is activated in the second universe (Selman 2002). The scene graph for the planetary and satellite orbit model with the necessary branch groups are shown in figure 7.

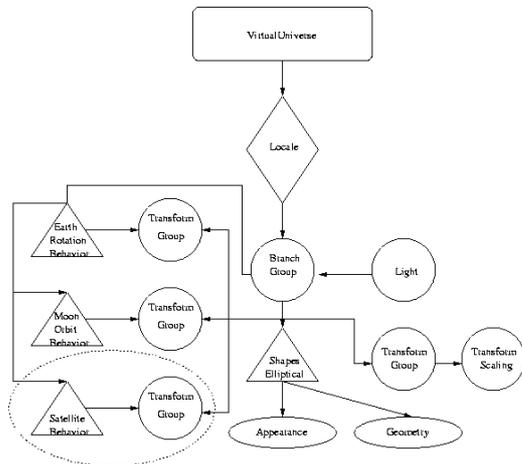


Figure 7 Scene graph for planetary and satellite orbit

The elliptical orbits are added as Shape 3D to root branch group. There are three behaviors namely *Earth rotation behavior*, *Moon orbit behavior* and *Satellite orbit behavior*. The satellite behavior will be initiated after completing the launch in an expected trajectory from the first universe and it is shown as a dotted ellipse in figure 8. If the trajectory is not an anticipated trajectory, satellite orbit behavior can not be started. The earth will have a zero radius so that it can occupy the first center position. The radius to the moon is added by an offset to the radius. The rotation and orbit behavior will operate from the center point. The satellite and moon have differential orbital speeds. The elliptical orbit is determined in the XZ plane and light directions are simulated in the direction of the orbit. The earth rotation is based on Y-axis and the rotation interpolator is used to achieve a particular speed.

Both universes are attached to a 3D canvas in an applet. The bounding sphere for orbit or rotator is assigned very large, so that the interpolator will always be active. The scaling for each branch group varies accordingly. The trajectories are constructed by *Bezier* curves (Lengyel 2004). The origin, end point and another two points which represent between the

start and end points are called control points. The curves are constructed as a sequence of cubic segments, rather than linear ones. The entire curve is contained in the quadrilateral whose corners are the four given points (their convex hull). These curves are very efficient to construct, since a simple recursion process means that the basic arithmetic operation needed to build the points along one is just division by two. Seven trajectories are constructed and dotted lines represent destruct limit lines. The bottom texture of the box has the appearance of Kennedy Space Center with launch pad locations. The screen shots of launch and orbit model are shown in figure 8 and 9.

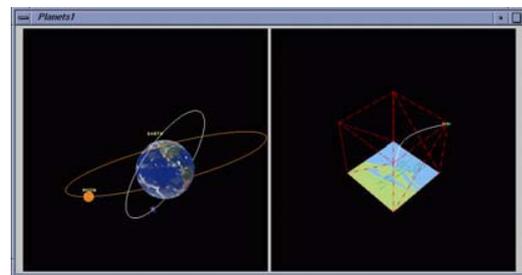


Figure 8 Screen shots of launch and orbit

In figure 8, the orange elliptical path represents the moon orbit and the white ellipse shows the satellite orbit. The debris dispersion is displayed in figure 9 and all debris particles fall into the elliptical impact zone (orange cluster of particles).

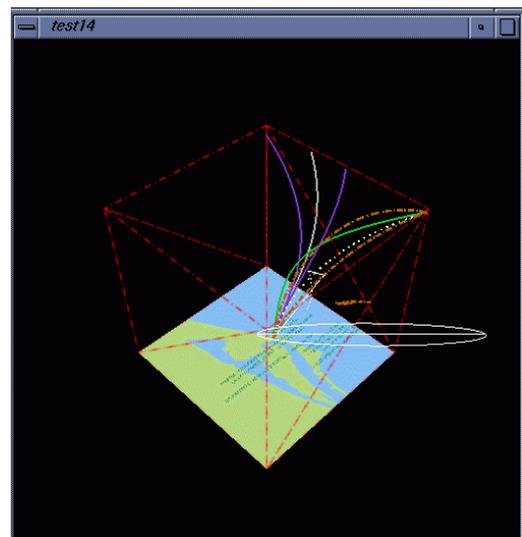


Figure 9 Debris dispersion model

The three dimensional visualization provides content to understand scattering effect and impact zones. Based on debris impact, human health risk assessment and expected casualty can be derived. These models will be further extended to different options of orbits and debris dispersion models.

CONCLUSIONS

The Java 3D model shows a simple orbital and debris dispersion model. The entire model will be used by Range Safety Officers as a simulation tool. Trajectories are constructed using *Bezier* curves and cubic splines. The behaviors are customized to suit our dispersion and orbital dynamics. Future research will focus on an updated dispersion model combined with wind profile (Rawinsonde data) using Java 3D. In this model, the user has to make a decision to destroy Shuttle when it violates flight launch rules. This will be automated and case based reasoning or instance based reasoning will be adopted to make a better decision support system based on launch commit criteria. Java 3D helps to deploy models on the web with suitable plug-ins to distribute among NASA centers.

The validation of models is based on telemetry and radar data to monitor orbital dynamics and any debris dispersion. Outside the Shuttle program, it can be used for missile tracking or any trajectory analysis. The visualizations are self-explanatory and can be interpreted easily by user. This visualization tool can be analyzed collaboratively using web. The applications of this tool can be used for studying nuclear fission and fusion, bomb blast dispersion models.

ACKNOWLEDGEMENT

We would like to acknowledge useful comments, suggestions and discussions from Dawn McIntosh and Stephen Rich. The authors also wish to thank Dr. Michael Shafto and Dr. Butler Hines for their financial support through the CICT program at NASA Ames Research Center.

REFERENCES

Bardina, J. and T. Rajkumar. 2003. "Intelligent launch and range operations Virtual test bed (ILRO-VTB)" In *Proceedings of SPIE Enabling technologies for simulation science*, vol. 5091 141-148 Florida, USA .

Baskett, R and J. Pace. 1995. *ARAC Dispersion modeling for January-March 1995 Vandenberg AFB Launches* Lawrence Livermore National Laboratory. USA.

Brown, K. and D. Petersen. 1999. *Ready-to-Run Java 3D*. Wiley Computer Publishing. USA.

Davidson, A. *Java Games Programming Techniques*. <http://fivedots.coe.psu.ac.th/~ad/jg/>.

FAA. 1999. *45th Space wing/Patrick air force base launch site safety assessment*. Prepared under contract no. DTFA01-95-D-03009. Licensing and Safety division, Washington D.C. USA.

Jensen, J; G. Townsend; J. Kork; and D. Kraft. 1962. *Design guide to Orbital Flight*. McGraw Hill Co. USA.

Lengyel, E. 2004. *Mathematics for 3D game programming and computer graphics*. Charles River Media Inc. Second Edition. USA.

NASA. 1988. *Range safety 1102*. Mission Directorate. Flight analysis branch. USA.

NRC. 2000. *Streamlining space launch range safety*. National Academy Press. Washington D.C. USA.

Selman, D. 2002. *Java 3D Programming*. Manning Publications Co. USA.

AUTHOR BIOGRAPHIES



Rajkumar Thirumalainambi, Ph.D. is a Computer Scientist at Science Application International Corporation (SAIC), working for NASA Ames Research Center, Moffett Field, California, USA. His research interests are neural networks, fuzzy logic, genetic algorithms, computer simulation and modeling. He is involved in building a virtual test bed for real time simulation of space vehicles using artificial intelligence techniques.



Jorge Bardina, Ph.D. works at NASA Ames Research Center in the Computational Sciences Division. He is leading the development of the Intelligent Launch & Range Operations, a part of NASA's Intelligent Systems program in collaboration with Kennedy Space Center and the University of Central Florida. His areas of interests are intelligent systems, modeling and simulation of collaborative team environments involving web-based, parallel architectures and engineering technology models.

MODELING AND SIMULATION OF SHUTTLE LAUNCH AND RANGE OPERATIONS

Jorge Bardina, Ph.D.
NASA Ames Research center
Mail Stop 269-2, Moffett Field
California, USA 94035
Jorge.E.Bardina@nasa.gov

Rajkumar Thirumalainambi, Ph.D.
SAIC @ NASA Ames Research center
Mail Stop 269-2, Moffett Field
California, USA 94035
rajkumar@mail.arc.nasa.gov

KEYWORDS

Shuttle Launch and Range operations, Weather Monitoring, Gas Dispersion, Debris Dispersion, Orbital Dynamics, Web-based Model

ABSTRACT

The simulation and modeling test bed is based on a mockup of a space flight operations control suitable for experiment physical, procedural, software, hardware and psychological aspects of space flight operations. The test bed consists of a weather expert system to advice on the effect of weather in launch operations. It also simulates toxic gas dispersion, impact of human health risk, and debris dispersion within 3D visualization. Since all modeling and simulation is based on the internet, it could reduce the cost of operations of launch and range safety by conducting extensive research before a particular launch. Each model has an independent decision-making module to derive the best decision for launch and range operations.

INTRODUCTION

The launch and range operations component of the test bed is a full-scale virtual dynamic operations control center for simulation of launch, range and safety systems. It will have the capability of simulating a variety of missions, vehicles, flight anomalies, and globally dispersed range operation via the internet, allowing new information and human factors technologies to be tested. Developing models for analyzing ground operations, trajectory analysis, and flight operations in an integrated system-of-systems fashion will give in-depth understanding of the process for mission planners, safety analysts and ground controllers. Intelligent Launch and Range Operations Virtual Test Bed (ILRO-VTB) (Bardina and Rajkumar 2003) is the tool and the process to integrate information technology for analyzing end-to-end shuttle and other vehicles in space launch simulations with special features

of virtual reality, scientific visualization, and command and control (Bardina and Rajkumar 2003). Initially four areas are focused in test bed viz., (i) Weather modeling (Rajkumar and Bardina 2003) (ii) Orbital dynamics and telemetry (Jensen et. al 1962) (iii) Range safety (gas and debris dispersion model) (iv) Decision modeling. For the Space Shuttle vehicles, weather forecasts are provided by the U.S. Air Force Range Weather operations facility at Cape Canaveral beginning at Launch minus 3 days in coordination with the National Weather Service Space Flight Meteorology Group (SMG) at Johnson Space Center in Houston. These include weather trends and their possible effects on launch day. A formal prelaunch weather briefing is held on launch minus one day which is a specific weather briefing for all areas of launch operations. The basic weather launch commit criteria focus on ambient temperature, wind speed, precipitation, lightning, type of clouds and cloud characteristics. In this paper, a decision support system for weather is constructed based on the real time data, which has been collected from different agencies (Rajkumar and Bardina 2003). During the launch, burning of the rocket engines in the first few seconds prior to and immediately following vehicle launches results in the formation of a large cloud of hot, buoyant exhaust products near the ground level which subsequently rises and entrains ambient air until the temperature and density of the cloud reach an approximate equilibrium with ambient conditions. The rocket engines also leave an exhaust trail from normal launches which extend throughout the troposphere and beyond. The toxic gas dispersion model calculates peak concentration and deposition downwind from normal launches. The required meteorological inputs for the dispersion model are vertical profiles of wind direction (Rawinsonde), wind speed, air temperature, pressure and dew point or relative humidity between the earth's surface and 3000 m. The toxic dispersion model is based on a

Gaussian approach which is a practical diffusion modeling tool (Boyd 1985 and Beychok 1995).

During launch of a rocket under prevailing weather conditions, commanders at Cape Canaveral Air Force Station evaluate the possibility of wind blown toxic emissions reaching civilian and military personnel near the area (FAA 1999). The Air Force uses a model called “Launch Area Toxic Risk Assessment (LATRA)” which is based on Monte Carlo simulation with a limited amount of data for toxic response functions to humans (Hudson et. al. 1999; Bennett and McDonald 1999). The main focus of toxic chemical species is hydrochloric acid (HCL), nitric acid (HNO₃), and nitrogen dioxide (NO₂) which are non-carcinogenic chemicals as per United States Environmental Protection Agency (USEPA). Without specific incidence data (e.g. mortality, acute illness etc) on humans or animals, it is difficult to endorse a particular response model to predict incidences. In this paper, we show a hazard quotient model, which is a ratio of estimated exposure concentration (EEC) to a no observed effect limit (NOEL) or other reference toxicity value (RTV).

If there is a malfunction during launches, the launch will be aborted or destroyed. Before such a condition arises; Range Safety Officers simulate debris impact zone and its effects for nearby residents around launch pad. The destruct limit lines, chevron lines, downward range safety simulations, and nominal trajectory simulations are computed using various physical models (FAA 1999). In the following sections, the intelligent launch and range virtual test bed architecture is provided in detail and underlying models are also explained.

LAUNCH AND RANGE OPERATIONS TEST BED ARCHITECTURE

The intelligent launch and range simulation test bed uses the latest information technology to bring a real time simulation to the web. The test bed consists of four dedicated servers which cater the complex simulation, modeling, data acquisition, processing and storage. Tomcat servers serve as web servers and Java Servlets, Applets act as the front-end graphical user interface (Bigus and Bigus 2001; Watson 1997). There are legacy codes (written in FORTRAN) which are running as backend processes supplying input data to other models. Java

applications acquire remote data for weather models. The decision model is based on a backward chaining expert system where rules are derived from flight launch rules. The overall architecture of the intelligent launch and range test bed is shown in figure 1. The ‘ILR01’ server is an independent web server, which processes various weather factors and automatically updates launch status for ‘GO/NO-GO’ scenarios (Rajkumar and Bardina 2003). In the back end, data acquisition, and processing of data is performed periodically by running a *cron* daemon. The ‘ILR02’ server provides an analysis of different types of orbits for different types of rockets. The orbital dynamics is provided in three dimensions, so that telemetry data can be captured which will provide enhanced knowledge of flight status (NASA 1988).

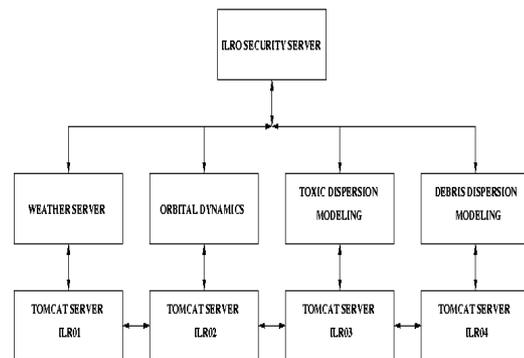


Figure 1 Intelligent launch and range test bed architecture

The toxic dispersion modeling coupled with geographical information system is served by ‘ILR03’ server. The dispersion models used are based on Gaussian dispersion model concepts (Boyd 1985). The dosage and concentration formulas are defined in rectangular coordinates (NRC 1998 and 2000). The x-axis is directed along the axis of the mean wind direction and y-axis is directed perpendicular to the mean wind direction. Normally the origin of the coordinate system is placed at the launch pad. The ‘*openmap*’ is a Java Beans based toolkit for building applications and applets needing geographic information. *Openmap* allows multiple layer integration to represent information about population density, gas dispersion, risk contours, etc.

The ‘ILR04’ supports the debris dispersion model, when there is a need for aborting a mission due to malfunction during launch. In the

present model, we have considered gravitational effect, air resistance, and particle/ground friction during settling of the particles. All particles are projected to disperse in an elliptical form where the particles hit the ground. If the particle has not reached the ground, there is a rotational effect for each particle in the air. In the following section each model is explained in detail and how these complex models interact with each other during simulation.

COMPONENTS OF THE MODELS

There are four major components involved in the simulation of launch and range safety systems to derive a 'GO/NO-GO' situation and they are : (i) Weather expert system (WES) (Rajkumar and Bardina 2003) (ii) Toxic gas dispersion model (Boyd 1985) (iii) Human health risk assessment model (Yassi 1998) (iv) Debris dispersion model and orbital dynamics (Lengyel 2004).

Weather Expert System

The weather expert system is launched by dedicated server as mentioned earlier and in the following sections the user interface, inference engine and knowledge base are discussed (Watson 1997).

User Interface

Java Servlet technology is adopted for accepting the user inputs and further analysis. In figure 2,



Figure 2 Graphical user interface of weather web server

there are 16 different buttons. The first four button rows deal with the US and North American continental weather system. The fifth button row provides information on global weather system including tropical cyclones. The

“Launch Decision” button activates the expert system and provides the expert decision for the shuttle launch. Except for the launch decision button, all other buttons invoke corresponding servlets. The servlets get the data or images from various sources across the US. The US weather button provides a 7 day weather forecast for a given zip code in the continental US. It provides a national weather service radar image and satellite image with daily weather forecasting. Apart from these images, specific weather details like humidity, wind speed, barometric pressure, heat index, and dew point are updated in hourly intervals. The U.S. Cloud classification obtained from the U.S. Naval postgraduate school at Monterey, California, lightning data from the National Lightning Detection Network, surface temperature, and wind speed from the National Weather Service are updated at 30 minute interval. Rawinsonde data are updated every day from the 45th Air Force wing located at Cape Canaveral, Florida. The sea state analysis is provided to the user to understand the booster rocket recovery. Weather criteria for NOAA in Spain and North Africa monitors an emergency landing at the TransOceanic Abort Landing Sites (TALS). The downloaded data are processed for Florida state and Cape Canaveral and form an input to the expert system. When the user presses the “Launch Decision” an expert system inference engine checks the values against the weather rules and it provides the Shuttle launch decision by GO or NO-GO.

In figure 3, the expert decisions for the Shuttle launch are shown below the button groups. The green value contributes to the GO situation whereas red value contributes to the NO-GO situation.



Figure 3 Weather expert system output

If GO should occur, every value in the lower frame should be green. The present expert system provides the decision for a generic Shuttle launch. For specific Shuttle launches, more stringent rules have to be added to the knowledge base.

Inference Engine

The inference engine looks at the goal variable of the expert system. The inference engine adopts a backward chaining mechanism because it only processes rules that are relevant to the questions and goals. It simply traverses the rule base trying to prove that clauses are true in a systematic manner. The rule is triggered, if all antecedent clauses are set to be true. The clause conditions are derived for each vehicle type.

Knowledge Base

The knowledge bases can be represented by production rules. These rules consist of a condition or premise followed by an action or conclusion (IF Condition .. THEN Action). Most of the rules for weather expert systems are derived from weather contingency rules developed over several years by NASA. Depending upon the type of launch vehicle, rocket propellant and payload, the weather rules change. The knowledge base consists of rules for GO and NO-GO situations. Depending on the prevailing weather conditions the expert system advises the end user. The details of Rawinsonde and other weather parameters form inputs for the toxic gas dispersion model.

Toxic Gas Dispersion Model

The required inputs for the gas dispersion model are vertical profiles of wind direction, wind speed, air temperature, atmospheric pressure and dew point or relative humidity between the earth's surface and 3000 m. This information is obtained during launch support activities from Rawinsonde measurements routinely measured at scheduled times throughout the pre-launch count down and after launch has occurred. The wind system is a series of 30 m towers located throughout Kennedy Space Center and one 152 m meteorological tower instrumented to measure wind direction, wind speed, turbulence and air temperature. Based on the inputs, the toxic gas dispersion model computes the dimensions of the ground cloud as a function of the height, distribution of vehicle exhaust products within

the cloud as a function of height, and position in space of the rising ground cloud as a function of time after launch until the internal cloud temperature equals the ambient air temperature (Boyd 1985; and Beychok 1995). The dosage concentration at an interval of 1 km from downwind of the launch pad is computed. For a normal launch, the assumption is made that all engines and the pad deluge system operate normally. In the case of a launch failure (single engine burn on pad), one solid engine does not ignite and the vehicle remains on the launch pad. In case of failure to lift off, an on pad explosion will cause scattering of solid rocket propellant. The fuel expenditure rates for normal launches are obtained by averaging fuel expenditure rates for the engines over the approximate period from lift off until the vehicle is about 3000 m above the surface. The fuel expenditure rates for the single engine burn are an average for the normal firing period of the engine. The exhaust cloud constituents are HCL, CO₂, and CO. The input to the toxic dispersion model is shown in figure 4.

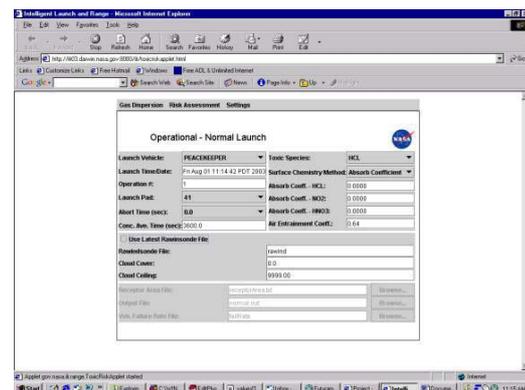


Figure 4 Inputs for toxic gas dispersion model

In figure 4, the user can select type of rockets, launch time and date, launch pad (39 A or B), concentration period (i.e., 1 hour) interval, cloud cover, Rawinsonde data, chemical species (HCL, NO₂, and HNO₃), surface chemistry and necessary coefficients. Once the user has provided all details, the backend of the server using FORTRAN code, computes chemical concentration at the ground level for a particular species. The chemical dosage is converted into contours and it is displayed in figure 5 via *openmap* interface. The concentration is expressed in parts per million (ppm) and five levels of contours are computed from minimum to maximum concentration. The contour simulation is performed by an applet. The

concentration computed at the ground is available for humans to inhale. The flight launch rules have an allowable limit of ground concentration for specific chemicals. If the simulated concentration exceeds the allowable limit, the mission will be kept on hold. The chemical concentration is monitored by real time sampling of ambient air after launch. The real time data and simulated data are compared and act as a surrogate data for other launches.

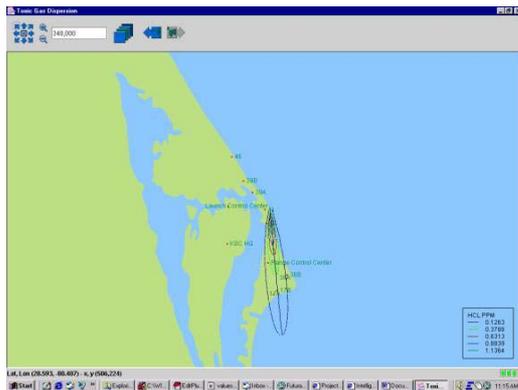


Figure 5 Chemical gas dispersion contour (HCL)

Human Health Risk Assessment Model

A binomial distribution is used to simulate the variance (uncertainty) associated with the predicted number of people affected. The potential for combined effects of exposure to more than one compound is estimated by developing joint probabilities of effect from the individual toxicants probabilities of effect. The LATRA model estimates for HCL, NO₂, and HNO₃ risk assessment. The available toxicological data for humans on the specified rocket emission toxicants are currently limited. The exposure response functions in the LATRA model are currently based on 1 hr time weighted average concentrations and ceiling values. Until more toxicological data is available, the hazard quotient model would be the most appropriate, which is developed by USEPA. The hazard quotient is the ratio of an observed or predicted exposure to an allowable exposure. The allowable exposure limit is set at a lower value by selecting an uncertainty factor that is sufficient to protect sensitive individuals. When the ratio of estimated exposure concentration (EEC) to the reference toxicity value (RTV) value is less than 1, effects are considered unlikely. When the quotient is greater than 1, some effects might occur in some individuals. As the value of EEC/RTV increases, both the severity and

incidence of effect are likely to increase, but the ratio is not used to predict incidence or severity. An additional advantage of the hazard quotient model is that it allows an estimation of the number of people at risk of additive effects from simultaneous exposure to two or more substances that is not possible in a traditional risk assessment. The risk to the exposed population is calculated by multiplying the individual risk and the number of exposed population (this should take into consideration age, other susceptibility factors, population activities etc.). In our model, the risk value is computed for a given latitude and longitude in a specified region of interest. The risk contour is generated based on the risk values and five levels of contour are plotted. The values are expressed in terms of 1 in million. The risk values are compared against the acceptable risk values and GO and NO-GO status is decided for the launch.

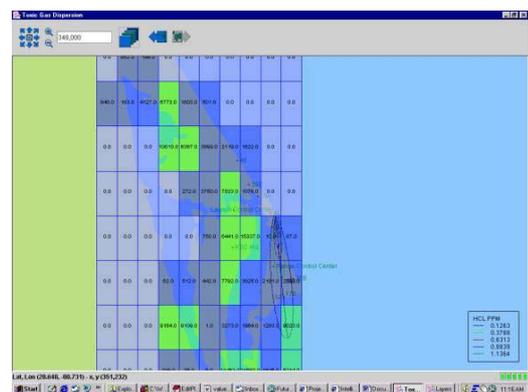


Figure 6 Population grid over Cape Canaveral

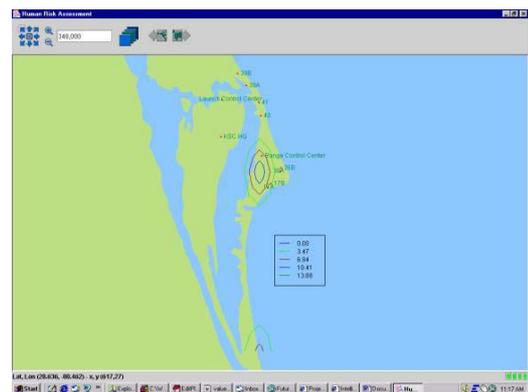


Figure 7 Human health risk contour

The figure 6 shows a population grid and it is added as a layer in *openmap*. The user can define any number of layers and it can be added dynamically. The population grid displays a selected region of interest and divides into 10 x

10 grids of equal intervals. The centroid of each square is computed by adding all population in the grid. The computed chemical concentration and population is translated into risk values based on the hazard quotient model. The risk contour is shown in figure 7. Presently two dimensional contours are plotted and a zoom feature is added via *openmap* interfaces.

Debris Dispersion Model

Range safety personnel evaluate various scenarios of failure during a launch. If there is a malfunction in separation of rockets, or any failure Range Safety Officers decide to terminate the mission. During termination, flight safety personnel will see that there is a minimum impact of debris scattering near inland. There are various flight rules before destructing the mission. In the present debris dispersion model, gravitational effect is implemented with air resistance. Wind effects are not considered. The debris dispersion model is developed in Java 3D with orbital dynamics. Presently all models interact with four web servers by issuing http requests. Since models are web based, it is easy to access from different corners of the world.

CONCLUSIONS

The complex launch and range safety models are served by four dedicated web servers. Furthermore we are enhancing support for ground operations by adding dedicated web servers. The test bed simulation combines physical model and decision models written in different languages to a unique test bed to visualize complex operations. Further research is planned to develop intelligent agents for each simulation and a decision support system to fully automate space launch initiatives. Since we use Java, it is compatible with any platform and operating system. Each class file in Java can be converted to an applet, Servlet, an application or a library in jar file. The virtual test bed technology enables an entire suite of applications and models for launch and range safety operations.

This project mainly focused a decision support system in a virtual environment using physical models and integrated in a single collaborative environment. The simulation of complex system of any type can be reconstructed using test bed. This test bed can be used as a generic test bed to

integrate various subsystems of a complex system.

ACKNOWLEDGEMENT

We would like to acknowledge useful comments, suggestions and discussions from Dawn McIntosh and Stephen Rich. The authors also wish to thank Dr. Michael Shafto and Dr. Butler Hines for their financial support through the CICT program at NASA Ames Research Center.

REFERENCES

- Bardina, J. and T. Rajkumar. 2003. "Intelligent launch and range operations Virtual test bed (ILRO-VTB)" In *Proceedings of SPIE Enabling technologies for simulation science*, vol. 5091 141-148 Florida, USA .
- Bennett, R. R. and McDonald, A. J. 1999. "Local environmental and toxicity issues for rocket launching and testing". *JANNAF 28th propellant development and characterization subcommittee and 17th safety and environmental protection subcommittee joint meeting*. 153-173., USA.
- Beychok, R. M. 1995. *Fundamentals of stack gas dispersion*. USA
- Bigus, J. and Bigus, J. 2001. *Constructing Intelligent agents using Java*. John Wiley and Inc. New York. USA.
- Boyd, B.F. 1985. "Operational use of REEDM". Weather Squadron 2nd Patrick Air Force Base, Florida. USA.
- FAA. 1999. *45th Space wing/Patrick air force base launch site safety assessment*. Prepared under contract no. DTFA01-95-D-03009. Licensing and Safety division, Washington D.C. USA.
- Hudson, J.M; A.M. See; and L. L. Philipson. 1999. "Launch Area Toxic Risk Analysis (LATRA): Risk management computer Program". *JANNAF 28th propellant development and characterization subcommittee and 17th safety and environmental protection subcommittee joint meeting*, 183-191., USA.
- Jensen, J; G. Townsend; J. Kork; and D. Kraft. 1962. *Design guide to Orbital Flight*. McGraw Hill Co. USA.
- Lengyel, E. 2004. *Mathematics for 3D game programming and computer graphics*. Charles River Media Inc. Second Edition. USA.
- NASA. 1988. *Range safety 1102*. Mission Directorate. Flight analysis branch. USA.
- NRC. 1998. *Assessment of Exposure-Response functions for Rocket-Emission Toxicants*. National Academy Press. Washington D.C. USA.
- NRC. 2000. *Streamlining Space Launch Range safety*. National Academy Press. Washington D.C. USA.
- Rajkumar, T. and J. Bardina. 2003. "Web-based Weather Expert System (WES) for Space Shuttle

Launch". In Proc. *IEEE SMC 2003*, Washington D.C. USA.

Watson, M. 1997. *Intelligent Java applications*. Morgan Kaufmann publishers. San Francisco. USA.

Yassi, A. 1998. *Basic Environmental Health*. World Health Organization.

AUTHOR BIOGRAPHIES



Jorge Bardina, Ph.D. works at NASA Ames Research Center in the Computational Sciences Division. He is leading the development of the Intelligent Launch & Range Operations, a part of NASA's Intelligent Systems program in collaboration with Kennedy Space Center and the University of Central Florida. His areas of interests are intelligent systems, modeling and simulation of collaborative team environments involving web-based, parallel architectures and engineering technology models.



Rajkumar Thirumalainambi, Ph.D. is a Computer Scientist at Science Application International Corporation (SAIC), working for NASA Ames Research Center, Moffett Field, California, USA. His research interests are neural networks, fuzzy logic, genetic algorithms, computer simulation and modeling. He is involved in building a virtual test bed for real time simulation of space vehicles using artificial intelligence techniques.

Supporting Multi-Level Models in Systems Biology by Visual Methods

S. Biermann, A.M. Uhrmacher, H. Schumann
Department of Computer Science
University of Rostock

Abstract

Multi-level models describe a system on different organization levels explicating the structure of a system, in terms of its components and the interaction between those. In Systems Biology, multi-level models result in a hierarchical structure, whose different layers might comprise thousands of model components. To make full use of the advantage of multi-level models, i.e. the explicit representation of the model's structure, visualization techniques are required that support overview and detail inspections in an interactive manner. Based on a model, which describes the tryptophan synthase as a multi-level model in JAMES, we show how different information visualization techniques, i.e. overview and detail techniques, Rings, interactive foldings of sub-structures and sinks for information hiding, can be combined to support the analysis of even highly unbalanced model structures.

1 Introduction

Bioinformatics is concerned with conceptualizing biology in terms of molecules (in the sense of physical-chemistry) and then applying informatics techniques (derived from disciplines such as applied mathematics, computer science, and statistics) to understand and organize the information associated with these molecules (Luscombe et al. 2001). Systems Biology can be interpreted as a sub-branch of Bioinformatics which is aimed at improving our insights into the dynamics of cellular systems. In this context modeling and simulation methods are crucial (Wolkenhauer et al. 2003).

During all phases of modeling and simulation, classical visualization techniques are employed (Nocke et al. 2003), i.e. during designing the model, experimenting with the model, and analyzing the results of the simulation. As do the different phases of modeling and simulation in general, the different visualization techniques serve an easier understanding of the system and its behavior (Slavik et al. 2003).

Among the different phases, the modeling process is considered crucial in gaining an insight into the system. Its effect is considered the larger, the less knowledge about the system is available, as is also argued in explorative modeling approaches (Davis 2000).

To support the modeling of complex systems, many formalisms, languages and tools allow to hierarchically compose models. They typically integrate the different traditional views in modeling systems, i.e. as functional models, as networks of interactions, and as hierarchical composition of models. Thereby, composition and interaction determines the overall structure of a model (Zeigler 1996). The more complex the structure of a model becomes, in terms of numbers, heterogeneity of components and interaction patterns, the more new visualization techniques are required that address the problems of these "large scale models". Under the term "Information Visualization" visualization methods have been developed to provide a "compact graphical presentation and user interface for rapidly manipulating large numbers of items ($10^2 - 10^6$), Effective information visualizations enable users to make discoveries, decisions or explanations about patterns (correlations, clusters, gaps, outliers, . . .), groups of items or individual items." (Shneiderman 2001).

The paper focuses on supporting the development and analysis of multi-level models based on applying and adapting advanced visualization techniques. The paper is organized as follows. First the concept of multi-level models and its role in Systems Biology is explained. Afterward a short overview on applying visualization techniques for modeling purposes in Systems Biology is presented. Thereafter, the multi-level model of the tryptophan synthase in JAMES is shortly described. In the following sections some of the employed visualization techniques are explained in more detail.

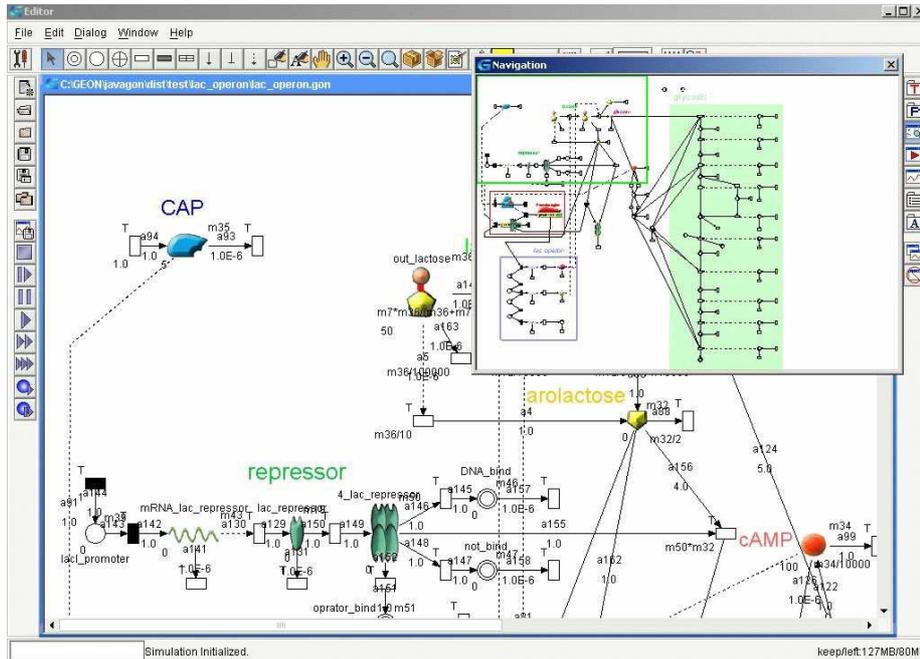


Figure 1: The interface of Genomic Object Net (Nagasaki et al. 2003)

2 Multi-Level Models in Systems Biology

When the term simulation is used in the context of systems biology most often it refers to continuous systems modeling and simulation (de Jong 2000). A series of simulation tools for continuous systems modeling and simulation in general and systems biology applications in particular exist, e.g. GEPASI (Mendes 1993), PRoMoT/DIVA (Ginkel et al. 2003). Continuous models reflect nicely what is measured in cellular biology. Small samples of cell cultures are analyzed by extracting the DNA, enzymes or metabolites and quantifying the concentration of the respective species over time. This type of model emphasizes a continuous, deterministic, macro perception on cellular systems.

However, other approaches interpret cellular systems as being composed of vast amounts of entities, each of which has a state and an individual behavior pattern. Their activities are triggered by discrete events, like the arrival and release of interacting species, or by the time flow, like the time required for intra-molecular rearrangements. This perception suggests using other modeling approaches (Strohman 2000) like a discrete event modeling approach, which has been applied to the dynamics of cellular systems already more than 20 years ago, e.g. by Bernard Zeigler (Zeigler 1981). One of the classical approaches in Systems Biology, which has been developed by Gillespie (Gillespie 1976), is based on discrete event simulation as well, even though it is best known for its stochastic modeling and simulation. It has led

to several extensions and refinements during the 90s. Among them are some that turn from the macro perspective of the analyzed system to a micro perspective where individual entities are described.

Individual-based models consist of multiple homogeneous entities, which do not interact directly but via the macro level. The macro-level model contains information about the number of individuals and other information about groups of individuals. The macro level is not restrained to simply aggregate the information of the micro level. It might have variables and a behavior of its own. In the later case not only upward but also downward causation of biological systems can easily be modeled (Campbell 1974). Individual-based models can naturally be employed to describe phenomena in Systems Biology, e.g. on the macro-level the cytoplasm or bulk solution keeps track of the concentrations and changes of concentrations and the micro-level comprises individual species like DNA, enzymes, metabolites etc, with their individual states and behavior pattern. If more than two levels of organizations are considered whose entities are not restricted to indirect interaction via the macro model, we arrive at the more general concept of multi-level models (Uhrmacher and Swartout 2003).

3 Visualization Techniques in Systems Biology

In systems biology visualization techniques are used throughout all phases of modeling and simula-

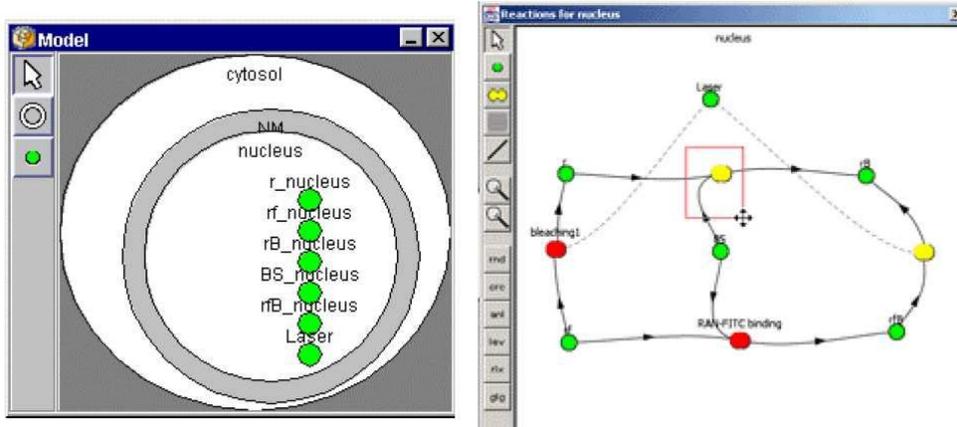


Figure 2: Model design with Virtual Cell (Schaff et al. 1997)

tion, i.e. to support the model design, the monitoring during simulation and the analysis of the achieved results (Allen et al. 2003). In the following we will concentrate on the first phase: designing or, to be more specific, representing the model and its structure.

Some modeling formalism lend themselves directly for a translation into graphical presentations. In this category formalism for discrete event systems modeling, e.g. Petri Nets or State Charts, and continuous systems modeling, e.g. Block Diagrams or Bond Graphs, belong equally. They are used by simulation systems to support the modeler in designing new models and are often provided as an alternative to non-graphical modeling languages. It is important to note that simulation systems support different types of users. Getting acquainted with an abstract modeling formalism holds little appeal to many users. If simulation systems are aimed at specific application areas, e.g. network simulation, or manufacturing simulation, the visual model can be based on metaphors and components libraries specific to this application area. Formal modeling formalisms and languages, on the one hand and pre-defined components, on the other hand, obviously address the need of different types of users.

To accommodate both types of users, two different avenues in supporting modeling via graphical interfaces are followed in System Biology.

- One is to enrich a general graphical modeling formalism with symbols from the application area. E.g. GENOMIC OBJECT NET (Nagasaki et al. 2003) combines the graphical formal representation of Hybrid Petri Nets with icons, that represent e.g. DNA, mRNA, or processes like transcription, to visualize the semantics of certain places and transitions (figure 1). These approaches inherit from the adopted general modeling formalism, that they are expressive, se-

mantically unambiguous, extendable, and supported by simulation engines.

- The second approach is to base the graphical modeling on metaphors and notations commonly used in the application area and thus, to obviate the need for defining the model in differential equations or other formal abstract languages. E.g. interaction diagrams are widely used in Biology and Chemistry. However, they lack the required expressiveness, semantic unambiguity, and consequently direct support by simulation engines. Their derivatives, like the Kohen diagram, have inherited these problems (Kitano 2003). To be executable by simulation engines, they have to be extended and refined. In VIRTUAL CELL (Schaff et al. 1997), a user defines a model of a cell by defining compartments and substrates, relating substrates to compartments (figure 2, left hand side), and interrelating substrates, the former is based on an interaction diagram variant (figure 2, right hand side).

One problem all modeling and simulation systems share: to display “large scale models”, so that users can explore details without losing orientation in the model.

For this purpose GENOMIC OBJECT NET already provides a technique from information visualization referred to as overview and detail. As can be seen in Figure 1, the user interface is divided into two windows. Whereas the larger window allows the user to explore his range of interest, the smaller window serves as a coarse-grained overview of the entire model. Thus, also a larger model can locally be explored in detail without losing the global orientation. However there are many more information visualization techniques, that can be used to reduce complexity and dimension of model visualizations in systems biology.

Generally, when faced with complex models, the visualization seeking mantra introduced by Shneiderman: overview first, zoom and filter, then details-on-demand, should be employed to provide different views with different granularity (Shneiderman 2001).

4 The Challenge of Multi-Level Model Visualization

Multi-level models integrate the different traditional views in modeling systems, i.e. as functional models, as networks of interactions, and as hierarchical composition of models.

At the lowest layer we find functional models of individuals. To define the interaction between models, models are equipped with ports. In modeling and simulation the distinction between system and its environment is crucial, so model components can be grouped to form more complex models, so called composite or coupled models that interact with their environment via their input and output ports. Most modeling formalism assume a strong composition, i.e. one model component belongs only to one coupled model. To support a successive hierarchical construction, the property of being closed under composition is crucial (Zeigler 1996).

Thus, as structuring elements we have the network perspective, who interacts with whom, and the composition. Both we also find in continuous macro models. A multi-level model contains sub-models that describe single enzymes, and sub-models that describe enzyme populations. So unlike hierarchically structured continuous models which typically comprises a medium number of heterogeneously structured sub-models, multi-level models are faced with representing different populations. Each of its sub-models might comprise 1000s of homogeneously structured individuals. The overall structure of the model might become rather unbalanced, as at the higher level of organization a couple of heterogeneously structured models (same as in the continuous realm) interact, however, one of which might contain several 1000s sub-models.

Thus, one perspective on the model will hardly suffice to visualize the model structure in a compact manner and to enable users to manipulate rapidly the model structure. Therefore, different perspectives shall be supported. At least the functional, the network, and the composition perspective should be distinguished.

5 Modeling the Tryptophan Synthase in James

Based on the multi-level model of the tryptophan synthase we will illustrate the visualization tech-

niques and their use in supporting multi-level models in Systems Biology. The visualization techniques are implemented as part of a Graphical User Interface which is currently under development for the simulation system JAMES (A Java based Agent Modeling Environment for Simulation) (Uhrmacher et al. 2000).

JAMES (A Java based Agent Modeling Environment for Simulation) (Uhrmacher et al. 2000) has been developed for simulating multi-agent systems. Systems are interpreted as communities of interacting autonomous entities, with the ability to adapt their behavior, interaction, and composition patterns. JAMES is based on the formalism DYNDEVs (Uhrmacher 2001) which adds reflection to DEVs (Zeigler et al. 2000) capturing the notion of self aware and self manipulating entities. The model design supports a hierarchical, compositional construction of models. It distinguishes between atomic and coupled models. Atomic models are equipped with input and output ports by which they communicate with their environment. Their behavior is defined by transition functions, an output function, and a time advance function which determines how long a state persists “per se”. DEVs models can be interpreted as time triggered automata and thus graphically represented as STATECHARTS, for a detailed discussion see e.g. (Schulz et al. 2000). A coupled model is described by a set of component models, which may be atomic or coupled, and by the couplings that exists among the components and between the components and its own input and output ports. Thus, the structure of coupled models forms a compound directed graph (see section 6).

The application example, i.e., the tryptophan synthase, catalyzes the final two reactions of the biosynthesis of tryptophan. In bacteria it exists as a nearly linear $(\alpha\beta)_2$ complex. Each α subsystem catalyses the cleavage of indole 3-glycerol phosphate (IGP) to produce indole and D-glyceraldehyde-3-phosphate, and each β subsystem produces tryptophan from L-serine and the channeled indole. The α and the β subsystem are connected by a largely hydrophobic tunnel. The α subsystem transforms IGP to indole and D-glyceraldehyde 3-phosphate (GAP). The first is forwarded to the β subsystem whereas the latter is released into the bulk solution. Its functionality is hampered by glycerol 3-phosphate (G3P).

In JAMES, the entire tryptophan synthase model is described as a multi-level model (Degenring et al. pear). The macro level contains models that describe the state and dynamics of the different populations of the bulk solution. The macro models responsible for the indole, the serine, the IGP, and the G3P interact with the “micro model” responsible for the synthase. The former keep track of the amounts of substrates, products and enzymes and defines the behavior at the level of concentrations and collision probabil-

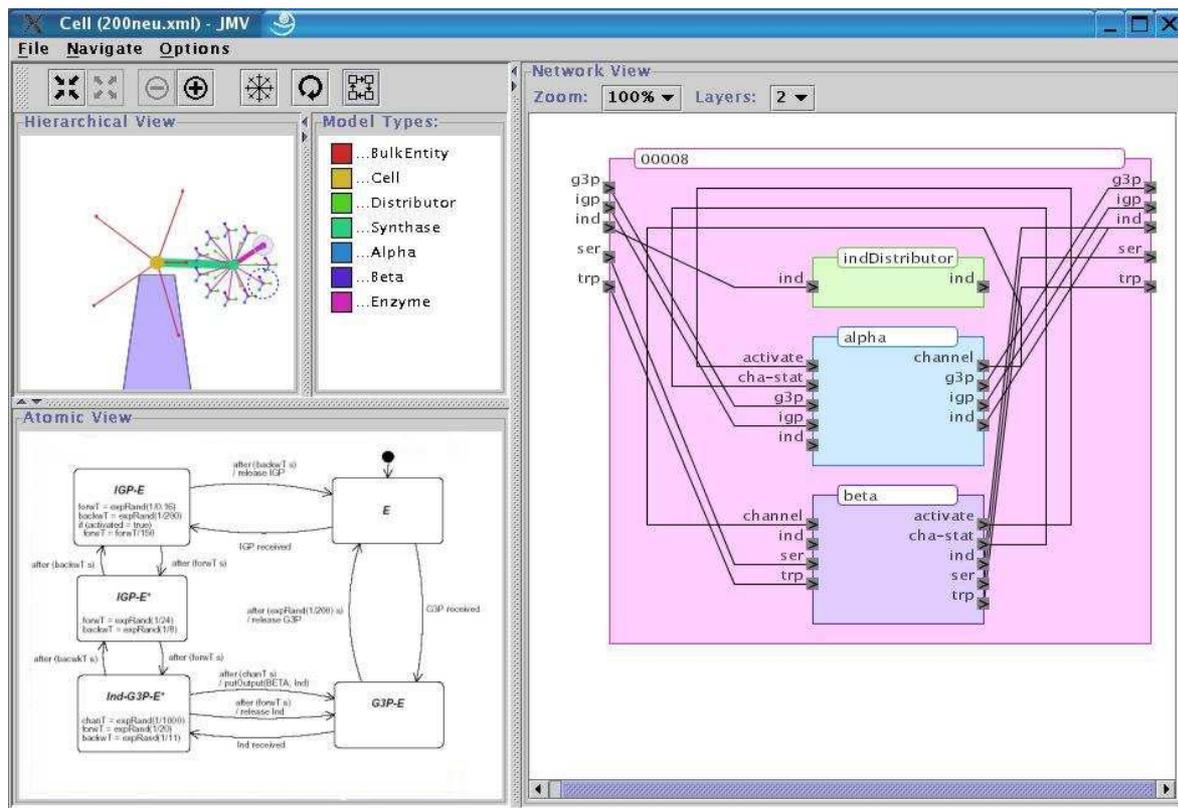


Figure 3: Screen shot showing the three different perspectives

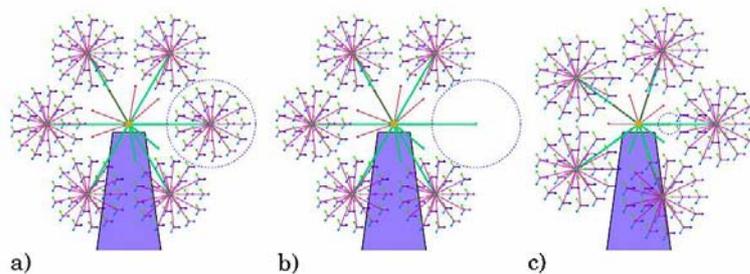


Figure 4: Interactive Folding.

ity. The micro model synthase contains thousands of models each of which describes a single enzyme, in upper left corner. As we are interested in the role, the channel plays in the tryptophan synthase, we define the enzyme model to consist of two different sub-units, i.e. alpha and beta, which communicate via the channel, see figure 3 on the right hand side.

The behavior of each sub-unit is modeled as discrete transitions from one state to the other. State changes might be triggered by the arrival of metabolites or by the flow of time, see figure 3 in the lower, left corner. The overall model structure is highly unbalanced, one of the children has more than 200 children, see figure 3, in the upper left corner. In the following we will discuss some of the realized fea-

tures.

6 Employed Visualization Techniques

The visualization has to be divided into different windows referring to the different perspectives of the model. The three different perspectives (see section 4) are combined to visualize models developed with JAMES: the atomic view, the network view and the hierarchical view (figure 3). This reflects the traditional distinction between functional, network, and hierarchical modeling and reduces the complexity and dimensions of each individual visualization. Thereby, the hierarchical structure of the model acts as the mental map, as it guides the user up and down

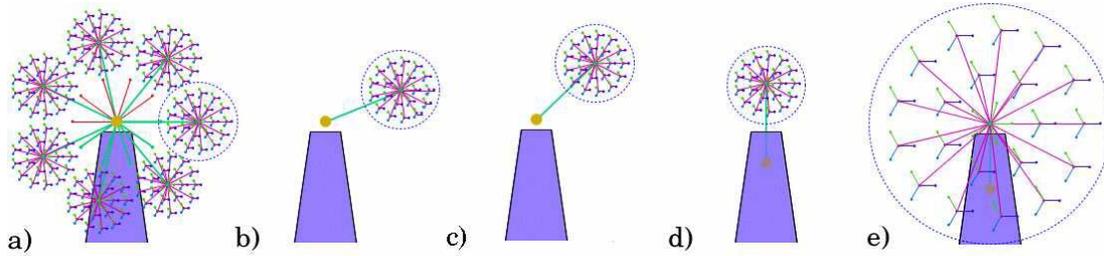


Figure 6: Zooming in based on sinks.

the different organizational levels. Only the compositional structure is shown and the user can interactively select an area of interest. The selected area is shown in a more detailed second window: the network view which shows the interaction structure of the selected model. If an atomic model is selected, its internal structure will be displayed in the functional view.

Since a model in JAMES can comprise a plethora of models at different compositional layers, we decided to use a radial techniques to present the hierarchical structure. Those techniques allow to display the entire hierarchical structure even with huge numbers of nodes. Among those techniques, the visualization technique RINGS (Ringed Interactive-Navigation Graph System) (Teoh and Ma 2002) exploits more efficiently the limited screen space than other radial techniques (figure 5). In addition, its ringed circular layout provides good support for the

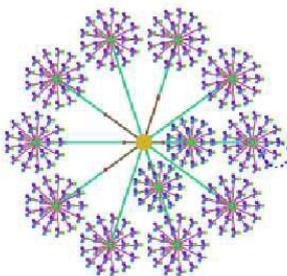


Figure 5: RINGS: Ringed Interactive-Navigation Graph System

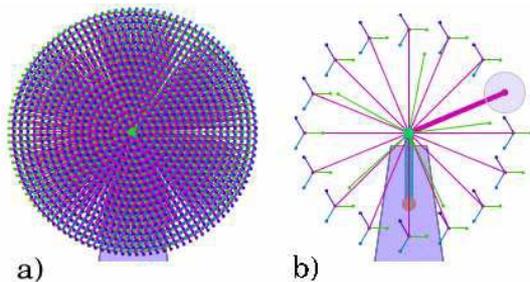


Figure 7: The functioning of a children sink

mental map of the user. In RINGS a parent node is placed in the center of all its child nodes. The child nodes are placed in concentric rings around the parent. Thereby nodes with many children are placed on outer rings, those with fewer ones on inner rings. For a full description of the layout algorithm of RINGS, see (Teoh and Ma 2002).

Nodes with many children occupy a lot of space. To more easily explore the structure of the model, appropriate information hiding techniques have to be integrated. Sub-structures can be interactively folded, which leaves more space for the remaining sub-trees. A folded sub-tree is shown as a blue circle within the window (Figure 4).

Another visualization concept are sinks where an explicit container collects the currently not visible entities. Two different types of sinks have been implemented.

- To zoom in, the sub-tree of interest can be selected. Thereby, all other sub-trees fade away into the sink, the root of the hierarchy moves visually into the sink while the root of the selected sub-tree moves gradually into the center. Now the subtree occupies the entire window. The sink is transparent and shows the predecessors of the sub-tree (Figure 6). The process is of course reversible.
- Using sinks, the number of displayed children can be reduced, as well. Here, the sink is moved interactively over the children that shall not be displayed (Figure 7). Different types of sinks can be used for “removing” children of different types.

The above techniques have been used in the context of exploring the hierarchical structure of the model and are features of the window HIERARCHY VIEW.

The window NETWORK VIEW utilizes the technique developed by Sander (Sander 1996). His approach is based on compound structured graphs. Whereas it clearly emphasizes the interaction structure between entities, the composition structure still remains visible. This facilitates the orientation of the user. However, as the composition hierarchy is

shown in a separate window and the system might comprise different composition layers with a plethora of components, it is important to be able to hide information on demand. Therefore, the depth of shown composition levels can be determined interactively.

If in the NETWORK VIEW an atomic model is selected, it is shown in the third window, the ATOMIC VIEW. Here the common Statecharts by Harel (Harel 1987) are used, to represent the functional level of the model. What is currently not visualized is the ability of a model to access its own structure, i.e. to change the composition, interaction, and behavior pattern of model.

7 Conclusion

The current work has concentrated on visualizing the structure of an existing model. The visualization of a multi-level model has to consider different perspectives, the atomic view, the network or interaction perspective and the hierarchical or compositional perspective. Whereas the atomic view could be realized by classical visualization techniques, for the network and hierarchical perspective more advanced visualization techniques have been employed due to the high number of models and interactions. The focus of the work has been on exploring the hierarchical structure of a model. A special problem for applying existing visualization techniques has been the typical, rather unbalanced structure of multi-level models in Systems Biology. To support the user in the analysis of, and orientation in the model structure, the presentation technique RINGS has been equipped with interaction techniques like folding, and installing sinks. The work presented is only a first step. Specific interaction features for the network perspective has still to be added, same as reflecting interaction facilities across the different views.

The current visualization is based on existing models. The next step will be to analyze the developed concepts and to adapt them to help fostering new multi-level models in Systems Biology. The visualization of the model structure has still to be connected to the running simulation, to animate the development of the structure. This is not trivial, due to the complexity of the models and due to the possibly changing model structures in JAMES. Thus, components and interactions might appear and disappear, and components might even move through the model structure.

Multi-level models promise a more flexible approach toward the understanding of cellular systems. However, they also provide new challenges for modeling, simulation, and visualization techniques, alike – which we have just started to address.

References

- Allen, N., C. Shaffer, N. Ramakrishnan, M. Vass, and L. Watson (2003, Dec.). Improving the development process for eukaryotic cell cycle models with a modeling support environment. *SIMULATION: Transactions of the Society for Modeling and Simulation*, 79(12), 674–688.
- Campbell, D. (1974). Downward causation in hierarchically organized biological systems in studies in the philosophy of biology. In F. Mayala and T. Dobzhansky (Eds.), *Studies in the Philosophy of Biology: Reductionism and Related Problems*, pp. 179–186. New York: Macmillan Press.
- Davis, P. (2000). Exploratory analysis enabled by multiresolution, multiperspective modeling. In *Proceedings of Winter Simulation'2003*, pp. 293–302. SCS.
- de Jong, H. (2000). Modeling and simulation of genetic regulatory systems: A literature review. Technical Report 4032, INRIA.
- Degenring, D., M. Röhl, and A. Uhrmacher (to appear). Discrete event, multi-level simulation of metabolite channeling. *BioSystems*.
- Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* 22, 403–434.
- Ginkel, M., K. A., T. Nutsch, R. Rehner, and E. Gilles (2003). Modular modeling of cellular systems with promot/diva. *Bioinformatics* 19(9), 1169–1176.
- Harel, D. (1987, June). Statecharts: A visual formalism for complex systems. *Science of Computer Programming* 8(3), 231–274.
- Kitano, H. (2003). A graphical notation for biochemical networks. *Biosilico* 1(5), 169–176.
- Luscombe, N., D. Greenbaum, and M. Gerstein (2001). What is bioinformatics? a proposed definition and overview of the field. *Methods Inf Med* 40, 346–358.
- Mendes, P. (1993). GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Computer Applications in the Biosciences* 9(5), 563–571.
- Nagasaki, M., A. Doi, H. Matsuno, and S. Miyano (2003). Genomic object net: a platform for modeling and simulating biopathways. *Applied Bioinformatics*.
- Nocke, T., U. Boehm, H. Schumann, and M. Flechsig (2003). Information visualization supporting modelling and evaluation tasks for

- climate models. In *Proceedings of Winter Simulation'2003*, New Orleans.
- Sander, G. (1996). Layout of compound directed graphs. Technical Report A/03/96, Universität des Saarlandes, FB 14 Informatik.
- Schaff, J., C. Fink, B. Slepchenko, J. Carson, and L. Loew (1997). A general computational framework for modeling cellular structure and function. *Biophys. J.*, 1135–1146.
- Schulz, S., T. Ewing, and J. Rozenblit (2000, March). Discrete event system specification (devs) and statechart equivalence for embedded systems modeling. In *Proceedings of the 2000 IEEE Conference and Workshop on Engineering of Computer-Based Systems*, Edinburgh, Scotland.
- Shneiderman, B. (2001). Supporting creativity with advanced information abundant user interfaces. In *Human-Centred Computing, On-Line Communities, and Virtual Environments*, London, UK. Springer.
- Slavik, P., M. Gayer, F. Hrdlicka, and O. Kubelka (2003). Problems of visualization of technological processes. In *Proceedings of Winter Simulation'2003*, New Orleans.
- Strohman, R. (2000). Complexity - organization becomes cause in the matter. *Nature Biotechnology* 18, 575–576.
- Teoh, S. and K.-L. Ma (2002). RINGS: A technique for visualizing large hierarchies. In *Graph Drawing*, London, pp. 268–275. Springer.
- Uhrmacher, A. (2001). Dynamic Structures in Modeling and Simulation - a Reflective Approach. *ACM Transactions on Modeling and Simulation* 11(2), 206–232.
- Uhrmacher, A. and W. Swartout (2003). Agent-oriented simulation. In M. Obaidat and G. Papadimitriou (Eds.), *Applied System Simulation*. Kluwer Academic Press.
- Uhrmacher, A. M., P. Tyschler, and D. Tyschler (2000). Modeling Mobile Agents. *Future Generation Computer System* 17, 107–118.
- Wolkenhauer, O., H. Kitano, and K.-H. Cho (2003). An introduction to systems biology. *IEEE Control Systems Magazine* 23(4), 38–48.
- Zeigler, B. (1996). A Framework for Modeling and Simulation. In D. Cloud and L. Rainey (Eds.), *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, Chapter 3. US Air Force.
- Zeigler, B., H. Praehofer, and T. Kim (2000). *Theory of Modeling and Simulation* (2nd ed.). London: Academic Press.
- Zeigler, B. P. (1981). Simplification in biochemical systems. In L. Segel (Ed.), *Mathematical Models in Molecular and Cellular Biology*. Cambridge University Press.

ACKNOWLEDGMENTS

The research on the simulation system JAMES is supported by the DFG (German Research Foundation).

AUTHOR BIOGRAPHIES

SUSANNE BIERMANN holds a MSc in Computer Science from the University of Rostock. She is currently participating at a postgraduate studies program at the Imedia Academy in Providence / USA. She is particularly interested in data visualization, information architecture, information technology, design and information visualization.

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 include the development of modeling and simulation methods, particularly for agent-oriented modeling and simulation, and the applications of these methods. Current projects centre on developing a component-based simulation framework to support the design and testing of software agents and on exploring the role of multi-level modeling and simulation in Systems Biology. She is co-editor-in-chief of the *Simulation - Transactions of the SCS*.

Heidrun Schumann is heading the Computer Graphics Research Group at the Institute of Computer Science at the University of Rostock. Her research profile covers Information Visualization and Visual Data Mining, Mobile Interfaces, and Rendering as well as Image Display. She was heading the cross-institutional research group “Multi-Visualization of Multimedia Information on Mobile Computer Systems,” supported by the German Research Society - DFG. Current research projects, supported by research institutions or industry focus e.g. on the development of scalable frameworks for information visualization, on the visualization of climate data and human health data as well as on problems of user defined image display and transmission.

3D VISUALIZATION AND ANIMATION OF METABOLIC NETWORKS

Ermir Qeli

Dept. of Math. & Computer Science,
University of Marburg
Hans-Meerwein-Str.
D-35032, Marburg, Germany
ermir@informatik.uni-marburg.de

Wolfgang Wiechert

Institute of Systems Engineering,
University of Siegen
Paul-Bonatz-Str. 9-11
D-57068, Siegen, Germany
wiechert@simtec.mb.uni-siegen.de

Bernd Freisleben

Dept. of Math. & Computer Science,
University of Marburg
Hans-Meerwein-Str.
D-35032, Marburg, Germany
freisleb@informatik.uni-marburg.de

KEYWORDS

Metabolic Networks, Metabolic Engineering,
Animation, Visualization, 3D, Java.

ABSTRACT

Metabolic networks represent an important research area in the systems biology field. Simulation is one of approaches used to understand their behaviour, and visualization and animation techniques help the human user to better understand the results of a simulation. In this paper, a novel approach for visualizing metabolic network simulation results is presented. In contrast to previous proposals, this approach makes it possible to animate the timely evolution of a metabolic network in three dimensions, allowing in this way a better interpretation of the results. The relevant issues regarding the proposed animation concepts and their implementation in Java are discussed.

INTRODUCTION

Metabolic engineering is the targeted modification of metabolic pathways by means of genetic manipulations in order to achieve specific goals (Bailey 1991, Stephanopoulos and Sinskey 1993). Simulation plays an important role in metabolic engineering. Different models of metabolic pathways are built to describe the metabolism of a cell. Simulators operating on these models generate results which are then compared to *in vitro* experimental data and to *in vivo* data if they exist, in order to select the model which describes these data best. However, the selection process does not only require reliable simulation results but also their correct interpretation.

Visualization is very helpful in evaluating such simulation results. Current visualization tools typically offer 2-dimensional plots which represent the trajectories of concentrations of metabolites and reaction flows and which can be interpreted easier than raw numbers. Nevertheless, it is difficult for the human user to evaluate the relatively large quantities of data for the different phenomena at the same time.

In this paper, we present a new approach which allows to dynamically animate the evolution of a metabolic network based on generated simulation data in three dimensions (3D). This work is a continuation of (Qeli et

al. 2003), extending the past approach with a new visualization method to animate the progression of metabolites and reactions in the three-dimensional space. The 3D view allows to avoid some of the difficulties encountered in 2D visualizations. Even the central metabolic pathways, as for example the pentose phosphate pathway, cannot be drawn in 2D without line intersections. A much more difficult problem occurs when the metabolic cofactors like ATP or NADH are involved. They are coupled to almost all central metabolic reaction steps thus inducing a strong network connection resulting in many line crossings. Finally, only a few tools visualize the metabolic regulation (inhibition and activation of reactions by other metabolites). If this is done for a large network, it becomes extremely difficult to keep an overview over the heavily entangled 2D network graph.

This paper is organized as follows. The next section gives a survey of the related work in this field. Then, our new approach to 3D visualization and animation of metabolic networks is presented. Experimental results will be shown in the following section. Finally, some conclusions are drawn and possible future work is discussed.

RELATED WORK

The related work about metabolic network modelling is mainly concerned with different modelling and simulation techniques to simulate the behaviour of a cell. Several commercial or academic tools and frameworks exist which allow to model and simulate metabolic networks. Typically, they consist of the following three components:

1. Design component
2. Simulation component
3. Visualization component

However, in many cases the first and the third component are omitted. This, of course, does not affect the quality of simulation, but the presence of these components makes the evaluation of simulation results much easier.

Gepasi (Mendes 1993, Mendes 1997) is one of the early simulation tools in metabolic engineering. However,

from the visualization point of view, it offers only common visualization methods in form of snapshots.

SBW (Hucka et al 2001a) is a framework intended to be extensible for running on different platforms to aid people in research in the systems biology field. SBML (Hucka et al. 2001b) is one of their major achievements, representing a standardized XML based format for modelling biochemical or metabolic networks, respectively. Several tools are included in this framework, *Jdesigner* being one of them. This tool allows the interactive design of metabolic networks.

Another tool called *Virtual Cell* (Loew and Schaff 2001) offers similar functionalities like SBW.

INSILICO Discovery (Mauch et al. 2001) is a commercial tool which offers a full fledged simulator and also allows the user friendly design of metabolic networks. The design is a mixture of manual interaction with different graph drawing techniques adopted for metabolic networks. Some of these techniques are described more precisely in (Karp and Paley 1994) and (Becker and Rojas 2001).

FluxAnalyzer (Klamt et al. 2003) is a MATLAB package which besides offering several functionalities for analyzing the structure of metabolic networks allows the visualization of so called interactive flux maps, which are a first step towards interactive visualization of metabolic networks.

MetVis (Qeli et al. 2003) offers a new approach for visualizing metabolic networks. It allows not only their interactive design, but also their animation in two dimensions according to results generated via simulation. The simulator used is MMT (Hurlebaus 2002; Haunschuld 2002) which generates simulation code based on pulse experiments.

Dwyer et al. (Dwyer et al. 2004) present a similar approach where experimental data is visualized directly in a metabolic network. This work is a continuation of (Brandes et al. 2003) where related metabolic networks are superimposed over each other to create a 2 1/2 dimensional view of metabolic networks.

MNV (Rojdestvenski and Cottam 2002; Rojdestvenski 2003) visualizes metabolic networks by means of VRML (Virtual Reality Markup Language) to allow a 3D view. However, it is somewhat difficult for a biologist to get accommodated with the visualization as the transformation from 2D to 3D is difficult to comprehend from a biological point of view.

OUR APPROACH

The approach we present in this paper is realized as part of *MetVis* (Qeli et al. 2003), but can also function as a separate application. Furthermore, *MetVis* is part of a framework which includes a simulator called MMT written in C++ (Hurlebaus 2002; Haunschuld et al. 2002). The exchange between *MetVis* and MMT is achieved by using XML documents for metabolic models and CSV (Character Separated Value) for

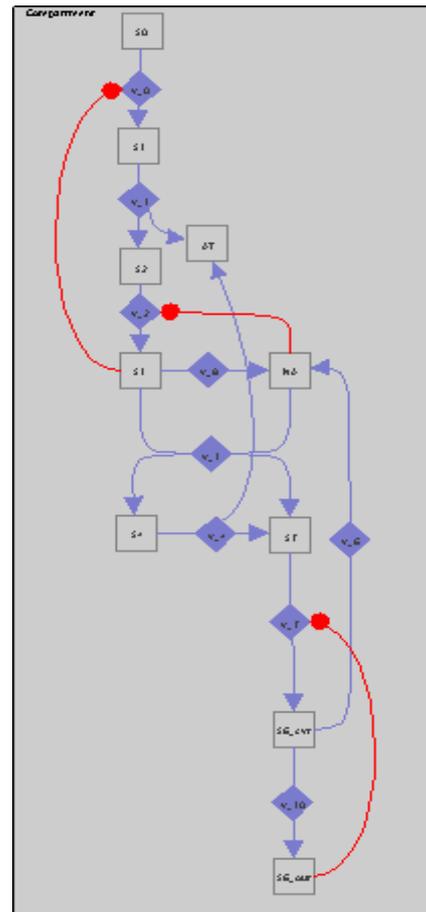


Figure 1: A Fictitious Metabolic Network

simulation data. From a graph theoretic point of view, a metabolic network is stored as a directed bipartite graph with two types of nodes: metabolites and reactions. The edges of the graph show which metabolites take part in a certain reaction. Another approach could be to model the metabolic network as a hypergraph.

Figure 1 shows a fictitious metabolic network designed in 2D, and figure 2 shows its corresponding XML fragment. Metabolites are represented by squares and reactions by rhombuses. The flows are represented by Bezier curves.

```
<!-- list of species -->
<specie name="AT2" compartment="X" initialAmount="1" fixed="1" />
<specie name="S0" compartment="X" initialAmount="1" from_data="S0_X" />
<specie name="S1" compartment="X" initialAmount="0.03" measure="S1_X" />
<specie name="S2" compartment="X" initialAmount="1.02" />
<specie name="S3" compartment="X" initialAmount="0.0826" />
<specie name="S4" compartment="X" initialAmount="0.0668" />
<specie name="S5" compartment="X" initialAmount="1.9" />
<specie name="S6_cyt" compartment="X" initialAmount="0.032" />
<specie name="S6_out" compartment="X" initialAmount="0.026" />
<!-- reaction v_1 -->
<reaction name="v_1" -->
  <listOfReactants -->
    <specieReference specie="S1" />
    <specieReference specie="AT" stoichiometry="2" />
  </listOfReactants -->
  <listOfProducts -->
    <specieReference specie="S2" />
  </listOfProducts -->
  <listOfKinetics -->
    <kineticLawReference kineticLaw="Kinetic1" -->
      <parameter symbol="ki" value="550" />
      <parameter symbol="ki" value="1" />
      <parameter symbol="pot" value="1" />
      <specieLink symbol="M1" specie="AT" />
      <specieLink symbol="M2" specie="S1" />
    </kineticLawReference -->
  </listOfKinetics -->
</reaction -->
```

Figure 2: XML Model of the Network of Figure 1

The XML format we use is a dialect of SBML (Hucka et al. 2001) which also supports model variants (Haunschuld et al. 2002). The graphical information related to the metabolic network is stored in another XML file. This allows to associate several graphical representations with the same metabolic network (Qeli et al. 2003).

The simulation data is obtained in form of a CSV file from MMT. In contrast to (Dwyer et al. 2004) where experimental data (the analogue of our simulation data) is stored directly in the graph data file, we have it separated to allow viewing the results of several simulations of the same metabolic network.

3D Visualization

2D visualization of metabolic networks represents a big step forward in the analysis of their functionality. Currently, it is the favourite approach in designing metabolic networks from the biological point of view. However, it has several drawbacks, one of them being the complex view that is created for large networks due to crossings that are created in two dimensional views. One way of eliminating crossings is to duplicate metabolite nodes, but there are no clear rules how often duplication should be done. Additional edges like the ones representing activation or inhibition relationships between a metabolite and a reaction complicate the situation further.

We have developed a new approach as part of MetVis which creates a 3D visualization based on a 2D visualization. The 3D view is intended to be used as a complementary part to the 2D view. The third dimension not only allows the elimination of crossings in 2D, but also allows us to visualize several metabolic networks in the same view for comparison purposes. Furthermore, the generated 3D views are similar to their 2D counterparts, without creating a totally different view which would confuse modelers in their work.

In the proposed approach, metabolites are represented by 3D cubes, and edges that in 2D were represented by Bezier curves with 4 control points are now represented by tubes that have the shape of a three dimensional Bezier curve and with a certain diameter. To eliminate crossings between edges, the two middle control points are displaced in different z-planes. Figure 3 presents the 3D visualization of the example shown in Figure 1, where the edges of the reaction $S1+Na \rightarrow S4+S5$ are in plane $z=0$ and those of $S4 \rightarrow S5+AT$ are displaced with a constant c . A separate z-plane is used for inhibitor and activator edges such that they do not visually affect the rest of the network, as shown in Figure 3.

3D Animation

To animate the 3D views, the raw data taken from a simulation is converted into relative percentages of the respective metabolite or flow, such that the values are in a certain range and are comparable to each other to

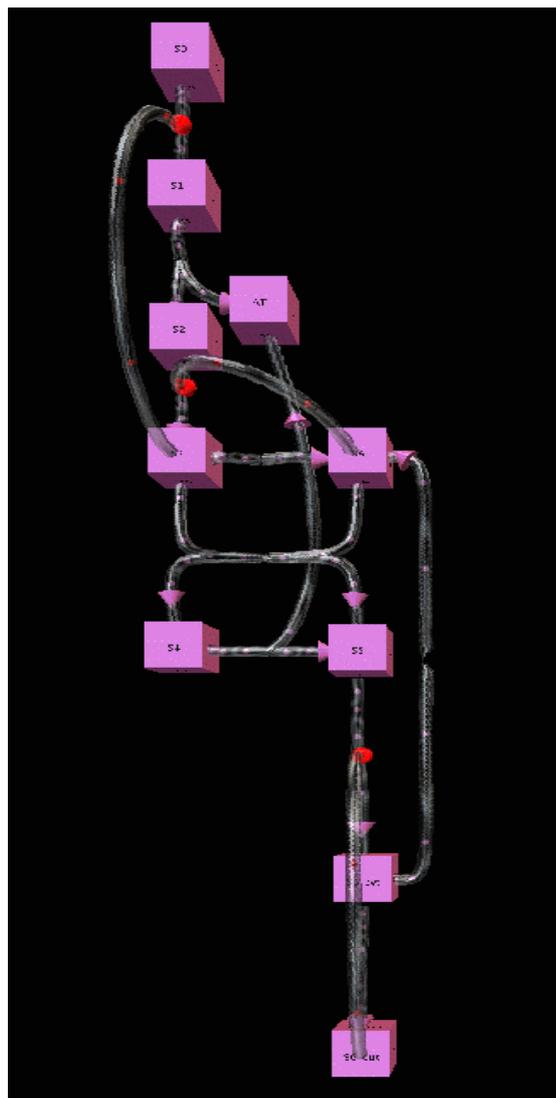


Figure 3: 3D Visualisation of the Model of Figure 1

avoid drawing false conclusions from their visualization. For each metabolite or flow X at time t , the percentage value of the respective metabolite/flow is calculated as follows:

$$Percentage(X,t) = \frac{Value(X,t) - Min(X)}{Max(X) - Min(X)} \quad (1)$$

where $Value(X,t)$ represents the value of metabolite X at time t and $Max(X)$ and $Min(X)$ represent the minimum and maximum value for each metabolite/flow during the entire simulation. Two types of objects are animated:

- For the cubes representing the metabolites, their z dimension varies according to the concentration of the corresponding metabolite calculated according to formula (1).
- Tubes represent flows for both reactions and inhibitions, and accordingly colored spheres moving within the tubes represent the material flow for the corresponding reaction. For the reaction flows, these spheres are colored with a nuance of blue, for inhibition

they are colored in red and for activation they are colored in green. The speed of flow of these spheres depends on the percentage speed of the respective reaction calculated according to formula (1).

Depending on the point of time, cubes will show how high the concentration of the metabolite is, and the reaction flow will be faster or slower.

Implementation

The approach has been implemented completely in the Java programming language. For 3D rendering, the idx3d library (Walser 2000) is used. This library is similar to the well know Java3D library, but offers additional functionality for our specific case and is in pure Java allowing easy inclusion in Java applications and applets without additional libraries.

The application consists of 20 Java classes. The I/O package is reused from MetVis. To allow effective user interaction and to achieve better performance, threads are used, which allow parallel processing of the rendering process. These threads are organized in a pool of threads, in order to keep the overhead of the administering thread low.

It is worth mentioning that for eliminating crossings, a simple but effective algorithm was implemented to assign different z-planes to the middle control points of bezier tubes. The nodes of the metabolic network (i.e. reactions and metabolites) are all left in plane $z=0$. Only the two inner control points of Bezier curves (i.e. edges) that connect metabolites with reactions are shifted to different z-planes. The edges are processed consecutively. In the beginning, the first edge is directly assigned to the first group of edges to be drawn in plane $z=0$. Then, the second edge is processed; if it intersects with the first edge, a new group is created with edges to be drawn in plane $z=c$, otherwise it is inserted in the first group. The third edge is processed in the same way, if it intersects any of the edges in the first group or second group (if this group exists), then a new group is created with edges to be drawn in plane $z=-c$, otherwise it is inserted in the first group where it does not intersect with other edges. The fourth edge would be drawn in the plane $z=2c$ if it intersects with at least one edge from every previous group, the fifth in plane $z=-2c$, and the algorithm proceeds in this way until all edges are processed.

EXPERIMENTAL RESULTS

Our project partners of the Institute of Biotechnology in the Research Center Jülich, Germany are the ones who create new models, simulate them, evaluate the simulation results and do experiments to see the correctness of their hypotheses. In figures 4 and 5 we see one of the models of a part of the metabolism of E.coli built in Jülich. Figure 4 shows a randomly selected step of the animation in 2D, and Figure 5 shows the same step in 3D. The advantages of 3D are

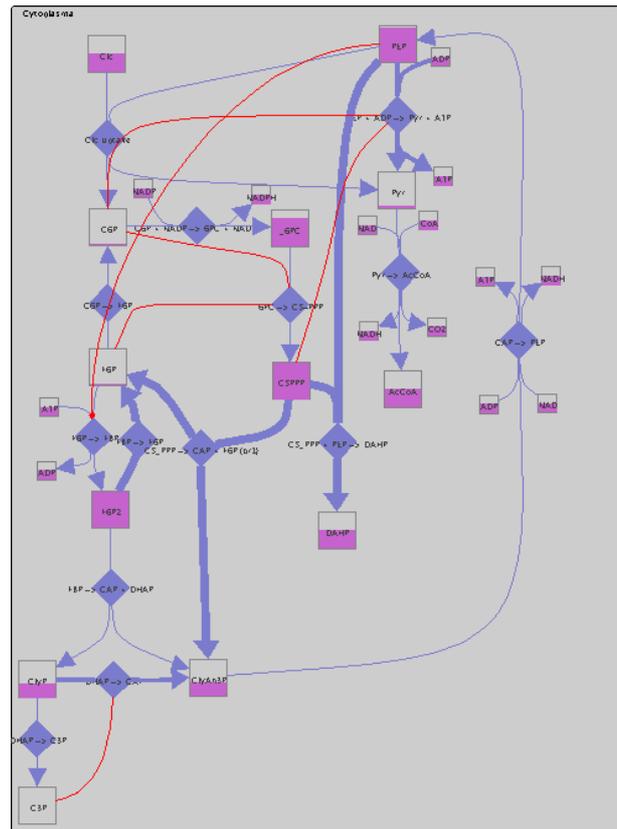


Figure 4: 23rd Step of a 2D Animation of a Partial Model of E.coli

not directly clear from the screenshots; a demo version may be obtained from the authors to view them in action. For example, whereas in the 2D animation one must be concentrated to view the changes in different flows, in 3D animation the speed of the movement of spheres makes it directly clear which part of the metabolic network is more active.

CONCLUSIONS

In this paper, we have presented a new approach to animate metabolic networks in three dimensions. In contrast to previous approaches, this approach allows the dynamic visualization of the evolution of a metabolic network model based on data generated via simulation. The approach could also be used to animate other kinds of networks where information about the flows in the network is available e.g. the visualization of regulatory networks in general.

There are several areas for future research. Currently, we are experimenting with different layout algorithms in two dimensions. The next step would be the adoption of these algorithms for the 3D space to increase the clarity of the visualization of metabolic networks but also remain consistent with the 2D representations biologists are used to. Highlighting specific pathways of a metabolic network and animating selected parts of a certain metabolic network is also a direction we are

PHILOSOPHY OF SIMULATION

VALIDATING SIMULATION MODELS

Klaus G. Troitzsch
Universität Koblenz–Landau
Postfach 201 602, D-56016 Koblenz-Germany
E-mail: kgt@uni-koblenz.de

KEYWORDS

Multi-agent model, mathematical model, stochastic model, simulation model, validation.

ABSTRACT

This paper discusses aspects of validating simulation models designed to describe, explain and predict real-world phenomena. It starts with a short review of arguments used in the *simsoc* mailing list discussion on theory, simulation and explanation a few months ago, deals with the use of quantitative and qualitative computational models to make quantitative and qualitative predictions or rather to draw conclusions from complex antecedents, and then discusses different types of explanation and prediction (and the relation between these two), It closes with an overview of topics in validity and validation from the point of view of the structuralist programme in the philosophy of science.

INTRODUCTION: THEORY, SIMULATION, EXPLANATION AND OBSERVATION

A few months ago, the *simsoc* mailing list experienced a longish discussion¹ which originated from Thomas Kron's question "about the relation of computer simulation and explanation, especially sociological explanation". More than fifty contributions to this discussion followed within less than three weeks, and contributors discussed the role of simulation in theory building (mostly, but not only) in the social, economic and management sciences — as well as the relation between observation on one hand and computer-assisted theory building (Hanne-man 1988) on the other. Scott Moss came back to his presidential address at the 1st conference of the European Social Simulation Association, Groningen, September 2003, in which he said "that if social simulation with agents is to be anything other than another in the long line of failed approaches to social science, it will be a positive departure only because it facilitates the dominance of observation over theory" and continued that the great successful scientists (outside the social sciences) built their generalisations around observation, developing new theoretical structures based on and validated by new evidence (quoted from his contribution to the *simsoc* mailing list as of November 14, 2003). Well in the line of this trait of thinking is the role of simulation or computational modeling which can be found in Gilbert and Troitzsch 1999

¹The discussion can be found in the November 2003 section of <http://www.jiscmail.ac.uk/archives/simsoc.html>, topics "simulation and explanation" and "theory and simulation".

which was recently extended by Alexis Drogoul (Drogoul *et al.* 2003: 5) and can be seen in Figure 1.

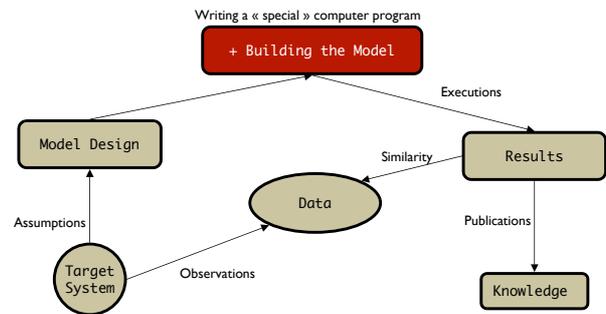


Figure 1: Drogoul's and his colleagues' interpretation of Gilbert's and Troitzsch's methodological proposition on the role of simulation

This diagram does not even contain the word 'simulation', but in its centre we find 'data' which are taken from observation and compared with results from simulation runs, for their similarity. Gilbert's and Troitzsch's original diagram describing "the logic of simulation as a method" (Gilbert and Troitzsch 1999: 16, 54, see also Troitzsch 1990: 2) is much the same: A model is built by abstraction from a target system, it is translated into a computer programme which can then be run and delivers results in the form of simulated data which can, and have to, be compared to data gathered from the same kind of target systems in the real world from which the model was abstracted.

Being aware that observation (as contrasted to just looking around in the world) presupposes at least some primitive form of theory (which tells us which entities and which of its properties to observe and which relations between them to register to find out whether there are some regularities), we should admit that our assumptions and our observation are not independent from each other (although Figure 1 insinuates this). And we should admit that in most cases computational (and other) models do not directly start from observation data but from a theory which in turn should build on, but often does not refer explicitly to observation data. Instead, we often start from a verbal theory which expresses our (or other authors') belief in how reality works, comparing simulation results with stylised facts instead of observation data.

A good example of this strategy is *Sugarscape* where the question "can you explain it?" is interpreted as "can you grow it?", and where "a given macrostructure [is] 'explained' by a given microspecification when the latter's generative sufficiency has been established." (Ep-

stein and Axtell 1996: 177)

At the other extreme, we might have microanalytical simulation which starts from a large collection of observational data on persons and households and the population as a whole. The model is initialised with empirical estimates of transition probabilities, age-specific birth and death rates and so on. Tens of thousands of software agents are created with data from real world people. And all this aims at predicting something like the age structure or kinship networks of this empirical population in the far future (see for instance Harding 1996).

In what follows we want to discuss the use of quantitative and qualitative computational models to make quantitative and qualitative predictions or rather to draw conclusions from complex antecedents and discuss different types of explanation and prediction (and the relation between these two) and close with an overview of topics in validity and validation.

QUALITATIVE AND QUANTITATIVE SIMULATION

Although most simulation uses quantitative procedures — doing calculations with numbers, often real valued, which make believe that the properties of the target system are quantitative, metric properties —, most of our mental models and verbal theories which are the predecessors of most of our simulation programmes do not talk about numbers and numerical values, but rather of properties which are categorical or, at best, ordinal. “However we claim that the use of numbers in this way is often simply a result of laziness — we often use numbers as a stand-in for qualitative aspects that we do not know how to program or have not the time to program.” (Edmonds and Hales 2003: 3)

Example: Gender desegregation among staffs of schools

The following example — which is taken from (Gilbert and Troitzsch 1999: 108–114) and earlier papers — tries to “explain” how the process of overcoming gender segregation in German schools went on in the 1950s and 1960s. The modeling process started from a large collection of empirical data showing the proportion of male and female teachers in all grammar schools in the federal state of Rhineland-Palatinate (approximately 150 in number) from 1950 to 1990 (see Figure 2, left graph). The model reproducing the empirical distribution of this proportion over time quite well was designed as parsimonious as possible, just assuming three hypotheses:

1. that all teachers leaving their jobs are replaced by men and women with equal overall probability (Article 2 line 2 of the German Basic Law),
2. that men stay in their jobs approximately twice as long as women (an empirical observation), and
3. that new women are assigned to an individual school with probability $P(W|\xi) = v(t) \exp(\kappa\xi)$ according to the percentage ξ of women among its

teachers (a theoretical assumption); κ is 0.5 in this simulation run, and $v(t)$ is such that at all times men and women have the same overall probability of replacing retired teachers, to comply with hypothesis 1.

The simulation is initialized with a gender distribution close to the empirical distribution of 1950. With $\kappa > 1$, gender segregation would continue and even become stronger.

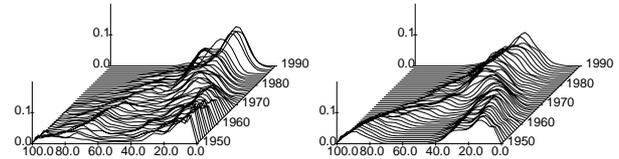


Figure 2: Distribution of percentages of women among teachers at 150 secondary schools in Rhineland-Palatinate from 1950 to 1990; left: empirical data, right: simulation

The simulation model reproduced the qualitative result that in the early 1970s the staff of all these 150 schools became mixed after twenty years of segregation where there were schools with high proportions of either male or female teachers but nearly no schools with between 40 and 60 per cent female teachers. And this reproduction / retrodiction was effected with the help of quantitative simulation, calculating probabilities of assigning teachers to schools. But did the model explain how and why this happened? Obviously not — since it is clear that the school authority, in fact officers in the ministry of education, did not cast dice or draw random numbers to select candidates for particular schools. Perhaps these officers saw to it that the overall proportion of men and women in school staffs was sufficiently equal to give women an equal chance, but even this has not been observed — instead we know that the process of desegregation of school staffs had entirely different origins: it was only the consequence of desegregation among girls and boys which in turn was due to the fact that most small towns could not afford separate schools for boys and girls (the percentage of girls in grammar schools rose steeply in the 1950s and 1960s). To summarise: a nice prediction (or at least retrodiction), but a poor explanation.

Example: Artificial eutrophication of a lake

Another example which is at the borderline between quantitative and qualitative simulation is the following. It was derived from a purely quantitative System Dynamics simulation in the tradition of Meadows and Forrester (Anderson 1973) which was used to quantitatively predict the consequences of bringing fertiliser into the soil in the neighbourhood of a lake and of actions taken to avoid these consequences by, for instance, harvesting algae or dredging the ground of the lake. This was, as it were, a simulation machine to predict the outcomes of real-world

experiments or perhaps to replace such experiments. Anderson's model was not designed to predict how farmers, fishers, tourist offices, local authorities around the lake would act when they realised that dead fish was swimming on the surface of the lake or when its water reeked of decay: this was only introduced in a revised model where local authorities — modelled as software agents — could decide which action to take when they were informed about the state of the lake, and where local farmers — also modelled as software agents — could decide whether it was more profitable for them to pay taxes for using artificial fertiliser on their fields and to grow more crop or to waive fertilising, not to pay fertiliser taxes and to be satisfied with lower yield (Möhring and Troitzsch 2001; see Figure 3).

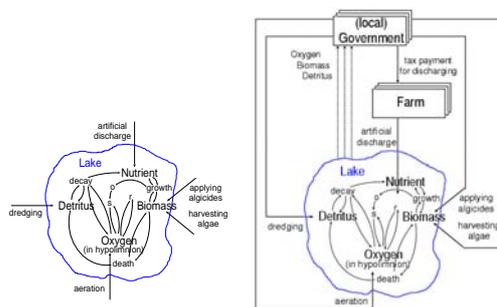


Figure 3: Two versions of Anderson's model of the eutrophication of a lake (Anderson 1973); left: a model of a lake subject to simulation experiments, right: a model of the lake and its (agri-) cultural environment

The difference between the two approaches is twofold:

- First, in the original approach Anderson took a formal model of the physical, chemical and biological processes running in a lake to simulate what could happen if these processes were disturbed by external influences imposed on the model by the simulating experimenter — whereas the extended model embeds the model of a lake into its social, political and economic environment and models influences external to the lake as internal to the model, thus taking into account that the lake and its socioeconomic environment interact and are interdependent.
- Second, the original approach starts from physical, chemical and biological theory providing the equations between the main variables describing the lake and generates quantitative simulation results (predictions) which might be compared to further observation data and help improve (fine tune) the theory of the biochemical processes occurring in a lake — whereas the extended model takes the behaviour of the lake for granted and adds a number of assumptions about the behaviour and actions of a number of human actors which are (or at least could have been) based on everyday observation

and discussions with stakeholders, and these assumptions are not aimed at generating quantitative predictions about the effect of the tax rate imposed on artificial fertilisers on the oxygen concentration in the lake, but only in qualitative predictions answering the question under which tax regime and under which additional measures taken by government and other neighbours of the lake its eutrophication could or could not be avoided.

Both approaches would serve as explanatory models: the restricted model would explain how and why under certain external or internal circumstances a lake would be eutrophicated and suffocate and what was the physical and biochemical mechanism behind the processes leading to total consumption of oxygen at the ground of the lake — and it would at the same time recommend countermeasures and allow for an estimation for their potential success; and the extended model would explain under which conditions such countermeasures would be taken by the population living around the lake and what incentives one part of the population would have to promise another part of the population to take the necessary countermeasures.

EXPLANATION AND PREDICTION

There is a long discussion about the question whether explanation and prediction are equivalent, or, to put it in other words, whether a theory which predicts empirical observations correctly at the same time explains what it predicts. Grünbaum (1962) pleaded for the equivalence while Scriven (1969) pleaded that both were “non-symmetrical”. If we consider prediction and explanation equivalent then our first example above would have explained the gender desegregation in German schools observed in the second half of the 20th century (although this was only retrodiction, but in principle, the three assumptions could have been stated in 1950), but this explanation is of the same quality as the explanation Mesopotamian priests could give 2,500 years ago for their (mostly correct) predictions of solar eclipses. In both cases, some scepticism is in order: from our research into the history of school staffs we know that desegregation had different causes than those stated in the assumptions, and the Mesopotamian theories of planetary movements were superseded 400 years ago by new theories which are substantially more valid.

The controversy between Grünbaum and Scriven, however, was different: Scriven had argued the other way round: “Satisfactory explanation of the past is possible even when prediction of the future is impossible.” (Scriven 1969: 117; Grünbaum 1962: 126) while we argued above that even when prediction of the future is possible with the help of a theory, this does not mean that this theory satisfactorily explains what happened (another theory could yield the same prediction and deliver a better explanation).

Without going into the details of this old controversy we should instead discuss what explanation and prediction could mean in the context of (social) simulation. Ep-

stein and Axtell argued that explanation of a phenomenon is achieved once the phenomenon could be reconstructed or generated (“grown”). From this point of view, the development of the distribution of percentages of female teachers in German grammar schools is explained by the three assumptions mentioned above, since this time-dependent frequency distribution as a macrostructure could be reconstructed quite well from the microstructure defined in the three assumptions. Of course, this reconstruction is by no means quantitatively precise: the two graphs are similar, but not identical (perhaps due to some simplifications in the assumptions, perhaps due to the fact that the random number generator in the simulation run which generated the time-dependent frequency distribution was not perfect, or for any other reasons) — and, of course, Sugarscape explanations are of the same, non-quantitative type.

What simulation models like these are designed to predict is only how a target system might behave in the future qualitatively; what we want to know is whether any macrostructures might be observed and what these macrostructures might look like, given that on a micro level some specific rules are applied or some specific laws hold. This is what we should call a qualitative prediction which at best would tell us that a small number of categorical outcomes can be expected with their respective probabilities. But this is not the type of prediction as the objective of simulation “which most people think of when they consider simulation as a scientific technique” (Axelrod 1997: 24) — “most people think of” attempts at simulating planetary formation (Casti 1996: 14) instead of “simulating the movement of workers or armies”. But if we use prediction in a non-quantitative sense, predictions delivered by simulations might still be useful “for the discovery of new relationships and principles” which Axelrod finds “at least as important as proof or prediction”. They might answer questions like “Which kinds of macro behaviour can be expected from a given micro structure under arbitrarily given parameter combinations and initial conditions?” The definition of this micro structure will typically be derived from observations on the micro level, and the simulated macro structures will typically be compared to macro structures in the target systems (which perhaps have not even been discovered). And a simulation model which generates a macro structure which resembles real-world macro structures from simulated micro structures which resemble micro structures observable in the real world might be accepted as a provisional explanation of real-world macro structures.

In a second step we might apply simulation to proceed to a second stage of qualitative prediction, where we are not interested in the general behaviour of a certain *class* of target systems, but in the future behaviour of a particular *instance* of this class of target systems — say, the future market shares of a number of competing products in a market, trying to answer the question whether most trademarks will survive with reasonable market shares or whether most of them will survive only in small niches whereas one product will gain an

overwhelming share of the whole market; this would still be a qualitative answer: we might not be interested in which trademark will be the winner, and we might not be interested in how many per cent of the market it will win (this would be only the third use of simulation, namely to predict quantitatively and numerically, as in microanalytical simulation and, perhaps, also in the simulation of climatic changes where we would not be content with the outcome that mean temperatures will rise but wanted to know when, where and how fast this process would have effects on nature and society). Or, to return to the example of the lake, its eutrophication and the countermeasures taken by its neighbours, we would

- first apply simulation to the very general question whether an artificial society “living” around an artificial lake which functions much like an empirical lake could ever learn to avoid eutrophication (something like a tragedy-of-the-commons simulation),
- then apply simulation to an empirical setting (describing and modelling an existing lake and its surroundings) to find out whether in this specific setting the existing lake can be rescued, and
- eventually to apply simulation to the question which political measures have to be taken to make the lake neighbours organise their economy in a way that the best possible use is made of the lake — and obviously this would be a discursive model in which stakeholders should be involved to negotiate and find out what “best possible use” actually means for them.

And to involve stakeholders in the development of a simulation model like this it will be necessary to validate the model (which could be done in the first two steps described just above) — otherwise stakeholders would not believe it was worthwhile to work with the simulation model.

TYPES OF VALIDITY

With Zeigler we should distinguish between three types of validity:

- replicative validity: the model matches data *already acquired* from the real system (retrodiction),
- predictive validity: the model matches data *before* data are acquired from the real system,
- structural validity: the model “not only reproduces the observed real system behaviour, but truly reflects the way in which the real system operates to produce this behaviour.” (Zeigler 1985: 5)

Zeigler here addresses three different stages of model validation (and development). Social science simulation does not seem to have followed this path in all cases: Since often data are very poor in the social sciences, early models, too, tried to be structurally valid and did

not bother much about replicative or predictive validity. “Data already acquired from the real system” were not available in a form that could be matched to the bulk of data simulation models had generated. There are several reasons for this difference between natural and social sciences: Data collection is a very expensive task in the latter, and in most cases it is even impossible to generate long time series for individual or group behaviour — individual attitudes, e.g., may be changed by the very measurement process, and groups may have changed in their composition before they were able to generate a time series which would have been long enough to allow for parameter estimation. On the other hand, the different kinds of influences non-living things exact upon each other are very much limited in their number, such that a structurally valid model can much more easily be found for the target systems natural sciences deal with than for social systems.

When talking about structural validity, a digression on structuralism might be in order: Structuralism as defined by Sneed (1979) and Balzer *et al.* (1987) sees both simulation models and observations as models of a theory which in turn — for them — is a mathematical structure consisting of (among others) three sets of such models. And these models — full models, potential models, and partial potential models — are defined as lists of terms and functions and (in the case of full models) invariants. Observations in this structuralist programme in the philosophy of science are intended applications of a theory, they are a subset of the set of its partial potential models in a sense that we can talk about them in terms which are non-theoretical with respect to a theory **T** in question (“T-non-theoretical terms”, for short). Elsewhere it was shown that a simulation model “*of a theory*” is “analogous to a structuralist reconstruction of this theory”, and that such reconstructions can easily be translated into simulation models and vice versa (Troitzsch 1994), provided the simulation language is object-oriented and functional (in other simulation languages the translation might be less straightforward). Simulation models would then be translated into full models in so far as they contain both T-non-theoretical terms (those we can use for talking about the target system irrespective of whether the theory is validated or not) and its T-theoretical terms — those which are only introduced by the theory, “in the sense that their meaning depends on **T**”, (Balzer *et al.* 1987: 40) — and, thirdly, the axioms or invariants the theory postulates — whereas observations (or rather: intended applications, to keep to the terminology of structuralism) are only partial potential models listing just the terms which are non-theoretical with respect to this theory. Thus, simulation is “richer” than observation.

Validation of simulation models is thus the same (or at least analogous) to validation of theories. In the sense of structuralism, we can interpret validation as the attempt at finding whether there exist intended applications of a theory (observations to which the theory refers) which belong to the content of the theory — which means that it should be possible to make an observation (in T-

non-theoretical terms) which complies with the axioms of the theory (which in turn might be expressed in T-theoretical terms, but then these must be linked to T-non-theoretical terms).

What does this mean for agent-based simulations in the range defined in the introduction? Sugarscape agents and plants correspond to T-theoretical terms, and the rules which the agents obey correspond to the axioms of this theory. But is there any empirical claim of the theory behind Sugarscape? If this theory predicts that — with a given parameterisation and initialisation — macrostructures emerge from the microstructures programmed into “its” models, and the emerging macrostructures sufficiently resemble observable macrostructures, we could admit that this observable macrostructure together with its microstructure (provided it resembles the model’s microstructure) is an intended application of the theory behind Sugarscape and that it complies with its axioms.

In the case of the empirical examples sketched above, the case is even simpler. Our model of a lake and its socioeconomic environment was based on observation, but it would still contain a number of terms which can only be used within a theory of, say, ecological consciousness: There would be some link between the state of the lake (its smell or colour) and the state of ecological consciousness of a particular person living near the lake (something like “the worse the water smells, the more am I willing to protect the lake from sewage”) and the action this person takes, and we could only observe the direct link between the observable smell of the lake and the observable actions taken, so the two “internal” links (as functions with their numerical coefficients, or as fuzzy rules with their membership functions) would remain theoretical with respect to such a theory — but the computer programme used for this simulation would still be a full model of this theory, because it would contain a function or rule representing this link, and that part of the simulation output which could be compared to empirical observational data would be the partial potential model of the theory. Stakeholders, however, might find that the T-theoretical links between the observable state of the lake and the observable actions on one hand and the T-theoretical state of ecological consciousness comply with what they think how ecological consciousness (if ever such a thing exists) works. And this could be the special value simulation could have in participatory modelling approaches (cf. the last few paragraphs of El hadouaj *et al.* 2001).

CONCLUSION

We dealt with the question about the relation of computer simulation and explanation, especially sociological explanation. and came to the conclusion that computer simulation programmes can be seen as models of theories from the point of view of the structuralist programme in the philosophy of science. This means that computer simulation should always have an empirical claim the same way as any theory should have an empirical content. Empirical claims of computer simulations come in different

forms — from quantitative predictions of future measurements down (or up?) to qualitative descriptions of possible scenarios. Both can be used to validate (the theory behind) the simulation model.

“Good validation of social simulation requires prediction” (Moss 2001: 9), but a good prediction is not always a sufficient indicator for validity. And “descriptiveness” is also a good indicator “for the validity of ... models” (Moss 2001: 10): When we model social processes in a participatory context, then agreement of the participating stakeholders on the validity of the model can be a reasonable indicator for the validity of the model.

References

- Anderson, J. M. (1973) The eutrophication of lakes. In D. Meadows and D. Meadows (eds), *Toward Global Equilibrium*, pp. 171–140. Wright Allen, Cambridge MA.
- Axelrod, R. (1997) Advancing the art of simulation in the social sciences. In R. Conte, R. Hegselmann and P. Terna (eds), *Simulating Social Phenomena*, pp. 21–40. Springer-Verlag, Berlin.
- Balzer, W., Moulines, C. U. and Sneed, J. D. (1987) *An Architectonic for Science. The Structuralist Program*, Synthese Library, Vol. 186. Reidel, Dordrecht.
- Casti, J. L. (1996) *Would-Be Worlds. How Simulation Is Changing the Frontiers of Science*. Wiley, New York.
- Drogoul, A., Vanbergue, D. and Meurisse, T. (2003) Simulation orientée agent: où sont les agents? In *Actes des Journées de Rochebrune, Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels*. Megève, France. janvier 2003, <http://www-poleia.lip6.fr/~drogoul/papers/Drogoul.rochebrune03.pdf>, last access 28 February 2004.
- Edmonds, B. and Hales, D. (2003) When and why does haggling occur? Some suggestions from a qualitative but computational simulation of negotiation. In *First conference of the European Social Simulation Association*. Groningen. September 2003, <http://www.uni-koblenz.de/~kgt/ESSA/ESSA1/essa-edmonds.pdf>, last access 28 February 2004.
- El hadouaj, S., Drogoul, A. and Espié, S. (2001) How to combine reactivity and anticipation: The case of conflicts resolution in a simulated road traffic. In S. Moss and P. Davidson (eds), *Multi-Agent-Based Simulation*, pp. 82–96. Springer, Berlin.
- Epstein, J. M. and Axtell, R. (1996) *Growing Artificial Societies – Social Science from the Bottom Up*. MIT Press, Cambridge, MA.
- Gilbert, N. and Troitzsch, K. G. (1999) *Simulation for the Social Scientist*. Open University Press, London.
- Grünbaum, A. (1962) Temporally-asymmetric principles, parity between explanation and prediction, and mechanism and teleology. *Philosophy of Science*, 29: 162–170.
- Hanneman, R. A. (1988) *Computer-Assisted Theory Building. Modeling Dynamic Social Systems*. Sage, Newbury Park, CA.
- Harding, A. (ed.) (1996) *Microsimulation and Public Policy*, Contributions to Economic Analysis, Vol. 232. Elsevier North Holland, Amsterdam.
- Möhring, M. and Troitzsch, K. G. (2001) Lake anderson revisited. *Journal of Artificial Societies and Social Simulation*, 4/3/1. <http://www.soc.surrey.ac.uk/JASSS/4/3/1.html>.
- Moss, S. (2001) Editorial introduction: Messy systems — the target for multi agent based simulation. In S. Moss and P. Davidson (eds), *Multi-Agent-Based Simulation*, pp. 1–14. Springer, Berlin.
- Scriven, M. (1969) Explanation and prediction as non-symmetrical. Explanation and prediction in evolutionary theory. In L. I. Krimmerman (ed.), *The Nature and Scope of Social Science. A Critical Anthology*, pp. 117–125. Appleton-Century-Crofts, New York. (First published in 1959 in *Science* 130: 477–482).
- Sneed, J. D. (1979) *The Logical Structure of Mathematical Physics*. Synthese Library 35. 2nd edn. Reidel, Dordrecht.
- Troitzsch, K. G. (1990) *Modellbildung und Simulation in den Sozialwissenschaften*. Westdeutscher Verlag, Opladen.
- Troitzsch, K. G. (1994) Modelling, simulation, and structuralism. In M. Kuokkanen (ed.), *Structuralism and Idealization*, Poznań Studies in the Philosophy of the Sciences and the Humanities, pp. 157–175. Editions Rodopi, Amsterdam.
- Zeigler, B. P. (1985) *Theory of Modelling and Simulation*. Krieger, Malabar. (Reprint, first published in 1976, Wiley, New York).

THE CONTROVERSIAL STATUS OF SIMULATIONS

Günter Küppers and Johannes Lenhard
Institute for Science and Technology Research (IWT)
University of Bielefeld
PB 100 131
33501 Bielefeld, Germany
E-mail: johannes.lenhard@uni-bielefeld.de

KEYWORDS

Epistemology, modelling, imitation, simulation, validation.

ABSTRACT

It is claimed that simulations maintain an independent status in knowledge production – they combine traits both of experiment and theory. In particular, simulation procedures differ fundamentally from numerical calculations in the strict sense, rather they can be seen as imitations of complex dynamics by a suitable generative mechanism. To argue for this claim results from a case study of climate research are employed. The simulation models of the general circulation of the atmosphere are based on theoretical models (PDEs), but they cannot be derived from the theory. The case study will show that simulation models have to take certain liberties. Namely, the performance on the computer is more important than the model's derivation and its accuracy of calculation.

INTRODUCTION

Computer simulations are employed in rapidly increasing fields of science and practice. In many cases, there is no other method of investigation at hand, be it that analytical methods break down because of the non-linear behaviour of complex systems, or that experimental methods are too risky or simply too expensive. To mention just some examples out of the broad spectrum where computer simulation are applied: How does a plasma behave under the influence of a magnetic field, how is a car body deformed during an accident, i.e. in a virtual crash test? Or how will the global climate change over the next fifty years? These are only three examples out of a large list of research questions tackled by simulation methods.

Simulations form an important part in the analysis of complex dynamics. This suggests the question: what is the nature of the knowledge produced by simulations? In particular, what can be said about the validity and certainty of simulation-based knowledge? Can questions of that kind be considered in a general manner at all? How can simulations be characterized in epistemic respect?

These questions belong to what can be called the philosophy of simulation. In the literature of the philosophy of science, one can find a controversial discussion. Are simulations merely numerical calculations of mathematical models, supplemented perhaps by more or less instructive visualizations? Or, to the contrary, do simulations assume an original position in the production of new knowledge?

Some philosophical accounts try to give a definition of computer simulations by considering the specific capacities of simulation models. E.g., Paul Humphreys derives his definition from the fact that traditional mathematical methods fail:

Working Definition. A computer simulation is any computer-implemented method for exploring the properties of mathematical models where analytic methods are unavailable. (Humphreys 1991, 500f.)

Various other accounts focus on further properties of simulation models, like their specific affinities to dynamic models (cf. Hartmann 1996; Brennan in McLeod 1968), or the visualisation of simulation results (cf. Rohrlich 1991, or Hughes 1999).

The dominating opinion, considered by Winsberg (2003) to be the “common view”, holds that simulation methods open up new possibilities to calculation. In philosophical respect, however, they would be rather uninteresting. The success story is impressive, but essentially simulations deliver the solutions of mathematical equations by the *brute force* of the electronic computer. For instance, M. Stöckler takes this view. He writes:

“Computer simulations enable tremendous progress on a pragmatic level, as a matter of degree in terms of speed and quantity, but not a revolution in the principles of methodology.” (Stöckler 2000, 356)

In the following, we should like to defend the opposite view. We shall argue that simulations constitute a new kind of synthesis between experimental and theoretical approaches.

SECOND ORDER MODELLING

We intend to show that computer simulation rely on procedures that differ fundamentally from numerical calculation in the strict sense, that is, they do not simply replace variables and parameters by numerical values in mathematical formulas. They do not solve complex systems of equations. Rather, simulations are numerical *imitations* of the unknown solution of differential equations, or the imitation of complex dynamics by a suitable generative mechanism.

The Monte Carlo method may serve as an illustrative example. This method goes back to the joint effort of Stanislaw Ulam and John von Neumann in Los Alamos, where they cooperated during the Manhattan project. Monte Carlo may count as the first simulation method (cf. Richtmyer and von Neumann 1946). Imagine that one intends to determine the volume of a certain body via Monte Carlo. One can embed the body into a cube of which the volume is known. The idea is to replace the (unknown) primitive by a ratio that can be determined “empirically”, or quasi-empirically, by iterating computer runs. The computer determines a point out of the cube at random, if this point belongs to the body, the trial is said to be successful. By re-iterating this random choice one can determine the unknown volume as the ratio of successful trials out of a great number of trials. To put it in other words: the integration is imitated by a generative mechanism.

In the following, we intend to foster this thesis of simulation as imitation, and we shall argue that simulation models cannot be derived directly from theoretical models like systems of differential equations. To find a convincing argument requires to avoid the easy way, and to choose for example a field where the theoretical approach via partial differential equations (PDEs) seems to determine the simulation model. Let us choose climate research as a case study, and have a closer look at the so-called general circulation models. Even here, simulations can be characterized as imitations of complex processes by generative mechanisms. Admittedly, these mechanism are guided by theory, but not determined by theory. At the end, we shall argue that the performance of the model on the computer is more important than its theoretical derivation and its accuracy of calculation.

Imitation of a Solution: the Dynamics of the Atmosphere

In 1955, Norman Phillips, working at Princeton’s Institute for Advanced Studies, succeeded with the so-called *first experiment* in simulating the dynamics of the atmosphere, i.e. in reproducing the patterns of wind and pressure of the entire atmosphere in a computer model. (Phillips 1956. For more details of the experiment, cf. Lewis 1998, for a broader history of ideas of the modelling of the general circulation of the atmosphere, cf. Lorenz 1967.) This development of a simulation

model of the general circulation of the atmosphere was celebrated as a major breakthrough. It surprised the experts, because it was generally accepted that a theoretical modelling approach would hardly be possible – the complexity of the processes that determine atmospheric circulation was judged insurmountable for an approach via a simple model.

After a little while, the simulation-based approach of such general circulation models (GCMs) became the silver bullet of climate research. In 1960 already, the *Geophysical Fluid Dynamics Laboratory* (GFDL) was founded in Princeton to follow up on this approach. This was the first institution with the official task to simulate in climate research. (The National Center for Atmospheric Research (NCAR) in Boulder, Colorado was founded in the same year.) Today, GCMs dominate climate research, being developed further at a handful of research institutions worldwide.

This first attempt to build a simulation model of the entire atmosphere was considered an “experiment”. This underlines how uncertain the success of this project was. At the same time, the conception of “experiment” expresses an important aspect for the methodology: in simulations, scientists use their models like an experimental set-up. Hence, the results of simulations acquire a quasi-empirical character.

The simulation model of the “first experiment” worked with a very coarse spatial discretization of the atmosphere. In vertical direction, it exhibited only two layers, and horizontally the grid cells covered more than 200.000 km². In the experiment, the initial state was an atmosphere at rest, with no differences in temperature and no flow. Phillips had to introduce the physical laws that govern the dynamics of the atmosphere. He used only six basic equations (PDEs) which since then are called the “primitive equations”. They are generally conceived of as the physical basis of climatology. These equations express well-known laws of hydrodynamics – the surprising thing was that only six PDEs were sufficient to reproduce the complex behaviour, and Phillips had the skill and luck of making an adequate choice.

The physical basis had to be adapted to the grid. The construction of a discrete model is a typical task of simulation modelling. The global and continuous equations of hydrodynamics had to be reformulated in order to calculate the time evolution of the relevant variables – pressure, temperature, wind speed – at the grid nodes step by step.

In the second step of the experiment, the dynamics was started, i.e. the radiation of the sun and the rotation of the earth were added. The atmosphere settled down in a so-called *steady state* that corresponded to stable flow patterns.

The tantalizing question was whether the model was able to reproduce the global flow patterns of the real atmosphere well-known from observations. For instance, one criterion was the complex pattern of the so-called surface westerlies, winds blowing continuously north the equator. The result was positive – everyone was impressed by the degree of correspondence. As we mentioned before, the experts were sceptical about the possibility of a global (and not far too complicated) model, but the empirical success was convincing. The decisive criterion for success was the adequate imitation of the phenomena, i.e. the flow patterns, not the derivation from theoretical principles.

The continuous primitive equations of the atmosphere were not solved in the strict sense during Phillips' experiment, rather, the phenomena of the atmosphere were imitated by the generative mechanism of the discrete difference equations. The success of the imitation was judged by its correspondence to the flow patterns observed. Insofar, the validation of simulation results relies on a quasi-empirical strategy.

There is a certain condition that simulations have to fulfil, a condition that you will certainly be well-acquainted with, namely stability. The generative mechanism chosen that is to imitate a certain dynamics, must "run" on the computer. It must not become instable because of discretization errors and truncation errors building up. Numerical instabilities are a severe and fundamental problem of simulation modelling.

The simulations of climate were caught up in this problem as well. In the course of the story of climate simulations, we will hit on a remarkable fact: The orientation to simulation as the numerical calculation of a solution led climate research to a dead end. Further progress became possible only after the insight had been accepted that in simulation modelling something more was allowed and required than strict dependency from the given system of equations to be solved. In particular, the discrete model cannot be deduced from the theoretical model (the primitive equations)! To justify this claim will be the strongest argument for our viewpoint that simulations are imitations and can be described as modelling of second order.

Arakawa's trick

Phillips simulation experiment was a tremendous success, but it exhibited also an important failure of the simulation model: the dynamics of the atmosphere were stable only for a few weeks. After about four weeks, the internal energy blew up, and the system "exploded" – the stable flow patterns dissolved into chaos. „After 26 days, the field ... became very irregular owing to large truncation errors, and is therefore not shown." (Phillips 1956, 145)

Nevertheless, the experiment was seen in general as a success. The possibility of simulating the atmospheric

circulation was not doubted. Instead, it was acknowledged as a challenge for further research to achieve a stable model. Questions of stability were important for climate research, as it was interested in long-time predictions. And they were of equal importance for the simulation method in general, because it is a rather general and typical task of replacing the "natural" dynamics of a system of PDEs by the "artificial" dynamics of a discrete system in a stable way. Phillips was well aware of the superior importance of stability issues and he suggested the truncation errors as the cause of instability.

Years of intensive and highly competitive research followed. The solution of the problem was assumed to consist in adequate smoothing procedures to cancel out the errors before they could blow up. This strategy was obviously oriented at the ideal of calculating as correctly as possible. Instabilities were seen as resulting from errors, inaccurate deviations of the discrete model from the true solution of the continuous system.

The decisive breakthrough, however, was achieved by a different approach, one pursued by Akio Arakawa, a mathematically highly gifted meteorologist who was developing a GCM at the University of California in Los Angeles (UCLA). For him, imitating the dynamics was of prime importance, and less the precise calculation of a solution.

At heart, Arakawa had realised that one could set aside a strict solution of the basic equations. One even *should* do so! If the time development of the simulation was reproducing the patterns of the atmosphere in a sufficiently adequate manner, and if the simulation was stable, then it was not obliged to be a solution of the basic equations – not even in the limit! To put the whole thing in a nutshell, the point of Arakawa's approach was: imitation of the phenomena beats solution of the equations.

Of course, this does not imply that one can simulate a given dynamics by completely arbitrary mechanisms. Far from that. And Arakawa very wisely adhered to the given equations. BUT he applied what later was to be known as "Arakawa's computational trick". The basic equations define a generative mechanism whose development over time is formally described by the Jacobi operator. Arakawa replaced the Jacobian by another operator he himself had constructed, later called Arakawa-Jacobian. The construction of the Arakawa operator is full of highly sophisticated mathematical arguments. The details do not matter in our context here (cf. Arakawa 1966, and the reconstruction in Arakawa 2000.) The pivotal fact is that the Arakawa operator permitted a stable long-time integration because he avoided the non-linear instability Phillips had had to face. Arakawa was able to prove this property mathematically.

To guarantee the stability of the simulation procedure, Arakawa had to introduce further assumptions, partly contradicting experience and the physical theory. He had to assume that the kinetic energy in the atmosphere would be preserved. This is definitely not the case in reality, where part of this energy is transformed into heat by friction. Moreover, dissipation is presumably an important factor for the stability of the real atmosphere. So we can summarize that Arakawa, in assuming the preservation of kinetic energy, “artificially” limited the blow-up of instabilities. In the real atmosphere, friction is responsible for that effect.

Incidentally, John von Neumann had employed a very similar strategy. He introduced an “artificial viscosity” to bring about a realistic behaviour while simulating the propagation of “shock waves” (cf. Winsberg 2003).

Utilizing conservation assumptions that obviously went against theory and experience was taken up very sceptical by the community. As Arakawa remembered, the tenor was: “Why require conservation while nature does not conserve?” (Arakawa 2000, 16) While most researchers were convinced that the promising path was to find a solution of the primitive equations as accurate as possible, Arakawa had made an additional modelling step. This step was not derived from the theoretical basis, and was justified only by the results of simulation runs that showed quasi-empirically that the Arakawa operator led to a successful imitation. The success of his approach was eventually generally accepted, and his initially controversial approach, is today conceived of as a “computational trick” (cf. Weart 2001).

It is an illustrative fact that Arakawa's approach proved to be superior in the course of a simulation experiment. At first, in the 1960ies, Arakawa had a couple of scientific rivals that tried to develop different smoothing procedures to avoid the problem of non-linear instabilities. Again, an experiment brought about a decision. In 1978, Jule Charney conducted a simulation experiment that consisted in the competition of different GCMs. They ran in parallel, starting with the same initial conditions. Three GCMs took part: Leith/Lawrence Livermore Lab; Smagorinsky/ GFDL Princeton und Arakawa/ UCLA. The first had implemented smoothing procedures, while the UCLA-model was based on the Arakawa-operator. Phillips describes the outcome:

“,...three general circulation models (...) were used for parallel integrations of several weeks to determine the growth of small initial errors. Only Arakawa's model had the aperiodic behavior typical of the real atmosphere in extratropical latitudes, and his results were therefore used as a guide to predictability of the real atmosphere. This aperiodic behavior was possible because Arakawa's numerical system did not require the significant smoothing required by the other models, and it realistically represented the nonlinear transport of

kinetic energy and vorticity in wave number space.“ (Phillips 2000, xxix)

This strengthens our argument: The traditional view interpreted simulation as numerical calculation of the solution of a mathematically or physically defined system. In the face of instabilities, this view had come to a dead end. The desired stability could be obtained only by smoothing procedures. These, in turn, resulted in unrealistic long-term behaviour. The competing approach of Arakawa attained a more “realistic” picture of the dynamics of the atmosphere, because it was based on counter-intuitive, artificial and physically unmotivated assumptions. In other words: he utilised the partial autonomy of simulation modelling as a modelling of second order.

CONCLUSION

Let us take stock of the case study. We have seen that there is no relation that would permit to derive a simulation model from a theoretical, or from a mathematical model. Simulation modelling can – and must – take certain liberties. In short: performance beats theoretical accuracy.

Because of this partial autonomy of the simulations from the theoretical basis, simulations are not merely numerical calculations. Rather, they are models of second order in the sense of iterated model construction. This result does not seem to be restricted to simulation models that stem from a system of continuous PDEs. They constituted the extreme case where simulation as calculation seemed to be most plausible. Other simulation procedures employ generative mechanisms in a similar way to imitate a certain dynamic.

Therefore, the adequacy of a simulation model cannot be theoretically deduced, nor derived from general principles. Simulation results have to be judged by experience. The quasi-empirical approach, that permits to tune the models in face of theoretical model experiments, is a necessary methodological condition for simulations of that kind. Theoretical alternatives can be compared empirically. Hence, it is justified to speak of computer simulations as of experimenting with theories. This provides simulations an independent status in knowledge production, they combine traits both of experiment and theory. (For similar theses concerning the autonomous status of simulations cf. Rohrlich 1991, Humphreys 1991, Galison 1996.)

This seems to be a result that forms (at least) an important part of an epistemic characterization of simulations, and of the knowledge they provide. What does this imply for our initial question concerning the validity of simulation results?

There, however, one evil seems to have been replaced by another. Simulation methods can achieve a good performance without strict theoretical justification. But

then the results have to be validated by experience. Does this not fundamentally weaken the potential for predictions?

We maintain that no easy answer is possible. One may observe interesting new strategies of validation, e.g. in climate research. There the models are validated by comparing their predictions in retrospect with known data. One has to trust, however, in the stability of the dynamics to do so. In general, this is a challenge for further research: to make a survey of the pragmatic solutions that can be found in science to cope with the problem of validating this new kind of simulation knowledge.

REFERENCES

- Arakawa, A. 1966. "Computational Design for Long-Term Numerical Integration of the Equations of Fluid Motion: Two-Dimensional Incompressible Flow. Part I." *Journal of Computational Physics* 1, 119-143.
- Arakawa, A. 2000. "A Personal Perspective on the Early Years of General Circulation Modeling at UCLA." In *General Circulation Model Development*, D. A. Randall (ed.). Academic Press, San Diego, 1-66.
- Brennan, R. D. 1968. « Simulation is Wha-a-t? Part II." In *Simulation, The Dynamic Modeling of Ideas and Systems with Computers*, J. McLeod (ed.). McGraw-Hill, New York, 5-12.
- Galison, P. 1996. "Computer Simulations and the Trading Zone". In *The Disunity of Science: Boundaries, Contexts, and Power*, P. Galison and D. J. Stump (eds.). Stanford University Press, Stanford, California, 118-157.
- Hartmann, S. 1996. "The World as a Process. Simulations in the Natural and Social Sciences." In *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, R. Hegselmann (ed.). Kluwer, Dordrecht, 77-100.
- Hughes, R. I. G. 1999. "The Ising Model, Computer Simulation, and Universal Physics." In *Models as Mediators*, M. S. Morgan and M. Morrison (eds.). Cambridge University Press, Cambridge, 97-145.
- Humphreys, P. 1991. "Computer Simulations". In *PSA 1990*, Vol. 2, F. Fine and R. Wessels (eds.). Philosophy of Science Association, East Lansing, 497-506.
- Lewis, J. M. 1998. "Clarifying the Dynamics of the General Circulation: Phillips's 1956 Experiment." *Bulletin of the American Meteorological Society* 79, No. 1, 39-60.
- Lorenz, E. 1967. "The Nature of the Theory of the General circulation of the Atmosphere". Technical Paper No. 218, World Meteorological Organization WMO, Geneva, 115-161.
- Phillips, N. 1956. "The General circulation of the Atmosphere: a Numerical Experiment. *Quarterly Journal of the Royal Meteorological Society* 82, No. 352, 123-164.
- Phillips, N. 2000. "Foreword". In *General Circulation Model Development*, D. A. Randall (ed.). Academic Press, San Diego, xxvii-xxix.
- Rohrlich, F. 1991. „Computer Simulation in the Physical Sciences In *PSA 1990*, Vol. 2, F. Fine and R. Wessels (eds.). Philosophy of Science Association, East Lansing, 507-518.
- Stöckler, M. 2000. "On Modeling and Simulations as Instruments for the Study of Complex Systems". In *Science at Century's End*, M. Carrier, G. J. Massey and L. Ruetsche (eds.). University of Pittsburgh Press, Pittsburgh, 355-373.
- Weart, S. 2001. "Arakawa's Computation Trick", American Institute of Physics.
www.aip.org/history/climate/arakawa.htm.
- Winsberg, E. 2003. "Simulated Experiments: Methodology for a Virtual World." *Philosophy of Science* 70, 105-125.

AUTHOR BIOGRAPHIES

GÜNTER KÜPPERS was educated in physics and did his dissertation in theoretical physics on problems of pattern formation in hydrodynamic convection flows 1969, University of Munich. Habilitation in Science Studies 1991, University of Vienna. 1966 – 1974 research fellow at the Max-Planck-Institute for Plasma-Physics, Garching. From 1974 – 1994 Director of the University Centre for Science Studies. Since 1994 member of the board of directors of the Institute for Science and Technology Studies (IWT) at the Bielefeld University. His fields of specialisation are: systems-theory and theory of self-organisation, dynamics of social systems, innovation networks and knowledge society.

JOHANNES LENHARD was educated in mathematics and philosophy at the universities of Heidelberg and Frankfurt. 1993–1998 assistant professor at the University of Frankfurt, in 1998, he did his dissertation in probability theory (population genetics). After an intermezzo as mathematical consultant he changed to the University of Bielefeld. Currently he is conducting a research project on the method of computer simulations (together with Günter Küppers) that is funded by the VW-Stiftung. His fields of interest include: philosophy of science, science studies, epistemology of mathematics.

THE MODELING APPROACH IN ECOSYSTEM RESEARCH AND MANAGEMENT

Michael Hauhs
BITÖK
Universität Bayreuth
95440 Bayreuth, Germany
E-mail: michael.hauhs@bitoek.uni-bayreuth.de

Holger Lange
Skogforsk
Høgskoleveien 12
1430 Ås, Norway
E-mail: holger.lange@skogforsk.no

KEYWORDS

Modeling approach, interactive simulation, ecosystem management

ABSTRACT

The process of model building in the environmental sciences and when dealing with ecosystems is discussed. Two types of modeling approaches need to be distinguished: An algorithmic one, which has been used traditionally in physics, meteorology, and other branches where biological degrees of freedom are either absent or neglectable; and an interactive one, which is a new framework in computer science and seems to be most suitable in cases where organisms (including humans) as agents in ecosystems are to be taken into account. The first modeling approach is exemplified by state models in dynamic systems theory and expresses the correspondence imposed by laws of nature between inferential entailment in a formal system and causal entailment in natural systems. Modeling is to be separated from simulation. In the first case simulation is a less restrictive type of modeling in which the description of non-interactive behaviour is the purpose and no constraints on the correspondence to internal states are imposed. The second (new) modeling approach is exemplified by interactive simulation models. It is able to express the correspondence in behaviour imposed by engineering standards (or cultural norms in general) between documentation, training and application in interactive choice situations such as games or ecosystem management. It generalises the notion of simulation for interactive problems. In an idealised situation the strictest correspondence between behaviour in a natural and a virtual system is expressed as bisimulation. The principles for model building are shortly demonstrated with examples.

INTRODUCTION

The term “model” has a wide range of notions in science. After restricting the focus to environmental sciences, ecosystem research or ecology conflicting meanings remain (Stehr 2001). The widespread and confusing applications of the term “modeling” is similar to that of “force”. However, in the case of “force” some very specific meanings have been identified and given

formal definitions. It can even be said that the whole edifice of physics rests upon four notions of force and the models derived from the state concept. To what extent is this success story of modeling in physics transferable into sciences dealing with Life? Can model approaches dealing with living entities be formalised and to what extent has that already been achieved? Since modern computing has become cheaply available the term simulation has been used in the same or similar meaning as the term model. What is the difference between modeling and simulation for ecosystem management and its implications?

TWO TYPES OF MODEL REASONING

The algorithmic approach

We start from and base the subsequent argument upon the approach proposed by Rosen (Rosen 1991). However, later it will be necessary to extend the meaning of the term modeling approach to include a second case. Rosen distinguishes between material (natural) and formal realms and two forms of entailment. In the material world events are connected by causal entailment, whereas in the formal world they are connected through inferential (logic) entailment. (Figure 1). The concept of laws of nature is thought to connect the two and to guarantee a deep congruence between these two forms of entailment operating upon the actual configurations (states) in the natural and in the abstract realms, respectively. Here we will restrict this correspondence to an algebraic congruence, i.e. the equational forms of laws of nature. In addition to the correspondence imposed by laws of nature the encoding of observations represents a critical link. It is not part of the formal system and requires a way of intersubjective agreeing upon an objective content in observations. It is here where much of the “art of modeling” in the sense of Robert Rosen lies.

A diagram as in Fig. 1 could only be drawn after an inside/outside (or object/subject) distinction has been made, by an observer (modeler) exterior to the natural system to be observed, seeking a way of capturing systems in an efficient way through abstraction. This exo-observer compares two object systems and often acts as a filter - by either neglecting “unwanted”

behavior or actively preparing the system into defined initial states from which subsequent observations (of behaviour) can be repeated. This is the modeling concept that underlies dynamic system theory and the (Newtonian) natural sciences today (Rosen 1991). It connects these two worlds: the direction from the material to the formal is termed “encoding” (abstracting) whereas the opposite direction is termed “decoding” or realisation. In practical terms, the encoding step is the model building process, providing a system of equations that, when complemented with appropriate static system descriptors (e.g., boundaries) build the formal system. The decoding step is solving these equations for the appropriate system – this step produces model data which are then compared to the observed behavior for a given time period.

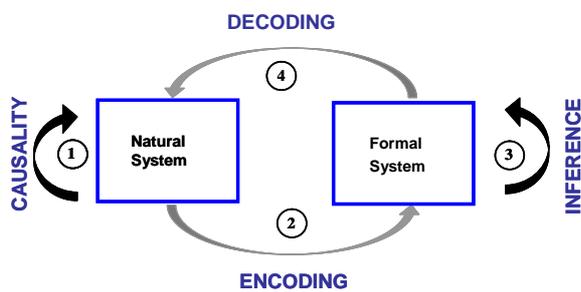


Figure 1: Modeling relation between real and abstract systems (Rosen 1991). This is the modeling approach that is used most widely in natural sciences and sometimes taken even as synonymously with them. The black arrows (1,3) indicate those parts of the cycle that are connected by and corresponding to laws of nature (Rosen 1991). The grey arrows (2,4) indicate the “open parts” in this loop (unentailed in Rosen’s terminology).

The nontrivial part of this loop is the encoding. The decisions about changes to be made in the model after observing deviations between model data and actual behavior in step 4 (Fig. 1) are also non-trivial.

The flow between natural and formal system indicated by the arrows is one of data and equations or mathematical abstractions. The latter have *lawlike character*, and the procedure has to be considered as repetitive. Lawlike means that its validity has temporal and spatial extension and stands behind observations being repeatable in space and time. The premise is that the procedure described by Fig. 1 eventually converges, i.e. that the formal system is a proper representation of the real world. It is the fundamental claim of natural sciences that nature is describable by natural laws, emphasizing the crucial role of reproducibility in experiments (Mittelstrass 1995). This is expressed by the notion of Natural Law in the words of Rosen.

The interactive approach

We propose a second type of modeling approach, suitable for interactive systems and their corresponding simulation models as given in Fig. 2. Here, no exo-observer is present, and the flow between the natural system and its virtual counterpart is represented by the observers themselves and their persistent memory when physically switching from one context to the other. Any participating agent in such a series of repeated transitions will be termed endo-observer. However, this diagram is to be drawn only after the system/environment boundary has been delineated for each of them (localization of the system that carries such persistent memory is the precondition for endo-observers). After each endo-observer has been localised one can pose the problem of how their collective memory can be intersubjectively documented.

The participatory agents run through a training cycle with an interactive simulation as part of the virtual system. Thereby the interactive relationship between environmental system and agent behavior is established. This leads to a pattern of actions and reactions which are then applied to the natural system after the endoobserver has (physically) moved back to the other side of the diagram. Usually, there is a mismatch between the trained pattern and the actual interactive relationship of the practitioner with the natural system. Thus, in a second step, the practical experiences are built into a changed description as virtual system. This latter step is the nontrivial one, as there is no obvious way to transform expert knowledge into formal descriptions (source code) in interactive systems. It is also not trivial to classify interactive systems correctly and distinguish them from non-interactive complex signals (see discussion).

Whether this loop converges or not is not a matter of natural laws - it is an active process and it has normative aspects: among all systems and behaviors the trained expert utilizes the most reproducible or stable ones, since these are the only ones he exerts control on. Convergence would imply that the differences in behaviour between virtual and natural system disappear - in sharp contrast to the situation of Fig. 1. In interactive systems such a convergence allows to assign an overall purpose (service) to the behaviour that can be interactively brought to a specific and repeatable outcome by experts. Typically the utilization interest rests upon the ability of experts to (interactively) judge and extend the service available from such a system. The existence of a set of expected reactions for the interactive phenomenon provides the basis for the normative background of any local decision (but this is not a functional one and requires evaluation competence of the expert).

We have argued elsewhere that the ability to interactively extend a service available from ecosystems

in an unlimited manner by experts, provides a definition of sustainable use of ecosystems (Hauhs et al. 2003). Criteria to test sustainability cannot be abstracted from the documented state of expertise and expert competence in valuation. For example, sustainability in forestry must be judged by changes in the forest ecosystem and the forester.

Interactive behaviour and the services that can be expressed this way has been studied in an abstract way in computer science. The notion of interactive computation has been derived from many situations where internal states are hidden and cannot even be reconstructed and the interest (specification) of the user is solely on the interactive service, i.e. the way the hidden system responds to inputs and outputs. For these problems the term “Bisimulation” has been introduced (Milner 1989) and extends the notion of observational equivalence into the temporal dimension for interactive systems. The states of a system are termed bisimilar, if an external observer is unable to distinguish between them by their input – output behaviour. These states can be substituted with each other with no consequences for the observer.

We are interested in the question whether this notion of bisimulation can also be used to express an underlying link in an (extended) modeling approach, analogous to the role of laws of nature in Fig. 1. Is it possible that the interactive behaviour displayed in the virtual realm of a simulation model becomes bisimilar with the interactive behaviour displayed in a natural (real) system? Apparently in some areas such as chess playing and pilot training the answer seems to be yes.

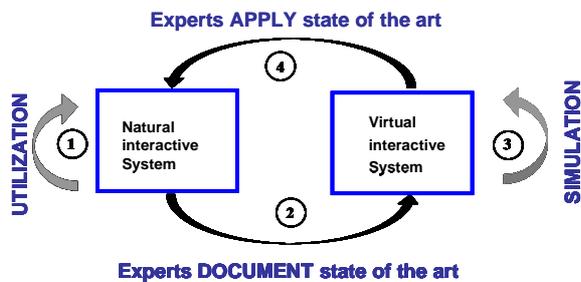


Figure 2: Modeling relation between real and virtual interactive systems. This approach is proposed for models that are capable of simulating interactive behaviour. The grey and black arrows have been exchanged relative to Fig. 1. Here the experts themselves physically move between the virtual and natural systems. Hence it is their consistent behaviour between training, application and documentation that links the cultural norms (standards) in natural and virtual systems. The grey arrows (1,3) indicate the open parts in this loop. It is here where the interaction takes place. These parts are by definition outside the range of algorithmic functions.

In such technical applications, the ultimate goal is that the expert’s abilities are equally valid both in the natural and the virtual system, and that the expert is no longer drawing a distinction between the two. The foremost example in this direction may be flight simulators.

In the following section we will inspect the numbered steps within Fig. 1 and 2 more closely. Examples for the two types of modeling approaches will be given when we try to integrate the two cases into an approach that generalises the one proposed by R. Rosen.

STEPS IN MODEL BUILDING AND COMMUTATIVE DIAGRAMS

The algorithmic approach

The graphical illustration of the modeling approach in Fig. 1 was chosen by Robert Rosen to intentionally resemble a commutative diagram of algebra, where the objects are the natural and the formal system and the encoding/decoding are the morphisms. The reasoning and logical inference of model building for systems amenable to an algorithmic representation is outlined in Fig. 3.

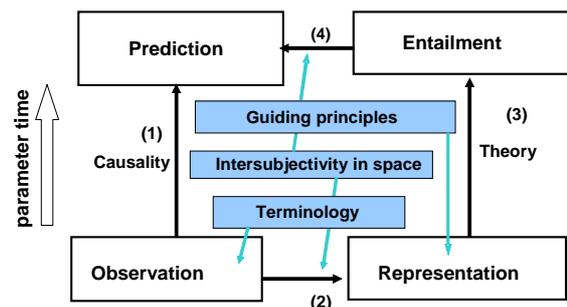


Figure 3: The process of model building for state systems

At the starting point we have observations (as made by an exo-observer) from a given system at a given time (period). The natural system develops over time from an initial state to a future state (path 1) in a causal way. This temporal evolution is accessible through local observations of the system for (at least) one additional point in time. This is indicated by the box “prediction”.

The crucial point here is that time is solely a Newtonian parametric one; the system is considered to be invariant with respect to time shifts. This opens the possibility to reproduce experiments by reinitialisation. The usually presupposed additional spatial translation invariance is less significant here, although we are for both algorithmic as well as interactive systems mostly interested in aspects not unique to the system.

In the context of these systems, only those observations are scientifically valid that are to a high extent observer-independent, communicable, not relying on individual

perspectives: they must be *intersubjective*. Abstraction e.g. from the individual competencies in handling measurement devices (say) is mandatory. We are aware that this is an idealization but stress the point that for interactive systems, this is an impossible requirement.

The formal representation of the observations (path 2 in Fig. 3) has to use a terminology agreed upon in the scientific community; it will typically involve objects, forces, and spatial configurations. When observations become encoded into state variables one has successfully abstracted from the histories. Each branch of natural sciences has its own standard set of referents.

On the right side, one seeks a minimal model (“representation”), from which inferential entailment is most comprehensive. From a fixed set of axioms (“laws”) and constraints the exo-observer hopes to cover a maximum of reproducible observations.

The inferential entailment (path 3) is the proper model building (the encoding procedure from Fig. 1). Out of the almost infinite set of choices among models, the scientist selects according to guiding principles. These comprise symmetries of the investigated system, associated conservation laws, restrictions of forces according to the observed variables, imposed initial and boundary conditions, parsimony considerations, and ultimately also aesthetic principles (“this theory is too beautiful to be wrong”). For well-established theories, these constraints reduce the remaining model space drastically.

The algebraic model allows to infer (simulate) the temporal development, starting with observational data, and a later comparison with the same set of observables as measured again from the real system (path 4). This predictive power of the model is most pronounced if there are counterintuitive phenomena in the formal system which are later confirmed by observations. The reference for a counterintuitive phenomenon must at least be partially be grounded on observation.

One example for the latter is sudden instabilities or phase transitions in well-controlled lab-scale systems (like the onset of ferromagnetism), another one the detection of celestial bodies like Neptune from a careful perturbation theory analysis.

The illustration of Fig. 3 implies a scientific program, which iteratively runs through the loop to enhance the model. This program may be considered as successful once the mismatch between observed and predicted phenomena disappears in practical terms. When (and only if) this is achieved, the diagram of Fig. 3 *commutes*: the sequence (2) → (3) → (4) leads to the same result as the direct temporal development (1):

$$(2) \rightarrow (3) \rightarrow (4) \Leftrightarrow (1) \quad (1)$$

However, the commutativity requires predictability, which is a property of the investigated natural system rather than the procedure described here.

Robert Rosen speaks about simulation when the correspondence of internal structures between natural and abstract systems is sacrificed. A simulation mimics the (non-interactive) behavior of a system, without necessarily having any synonymy of elements and their causal entailments in the physical system and the symbols and inferential entailments in the simulation. The drawback of simulations is that due to this lack of synonymy, one cannot learn much about the system under study by examining the internals of the simulation. Simulations try to reproduce (non-interactive) function rather than structure.

The interactive approach

For interactive modeling we need to extend Rosen’s notions of model approaches and of simulations.

The commutative diagram derived in Fig. 3 is given another interpretation in figure 4. The criterion whether commutativity was achieved in Fig. 3 is the congruence between predicted and observed system states in the upper left box. For interactive systems and an endo-observer participating in virtual and real interactions this criterion cannot be applied, as the base reference (“truth”, the observations in Fig. 3) is non-existing. Here the notion of bisimilar behaviour replaces that of congruent states (observable configurations). The test itself must therefore be extended in time as it may relate to a whole series of interactive decisions.

That is why we put (expert) memories in the lower left box of Fig. 4 instead of observations. Clearly memory (even if we refer to some vague standards of expert knowledge) is a much less well-defined concept than observation. Let us assume for the moment that this difference reflects just the historical situation and we know that in earlier days of modern science observations were regarded as highly dubious, too.

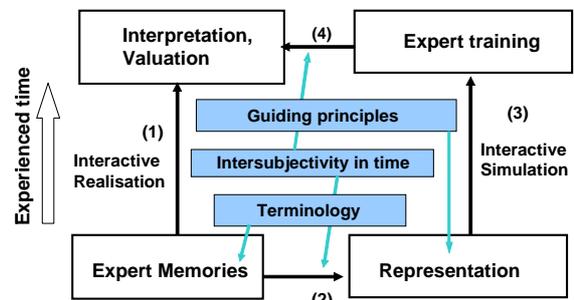


Figure 4: The process of building a simulation model for interactive systems

We will regard the iterative movements of experts through the cycle as the test by which their memories become intersubjectively documented. The encoding

(2) and decoding (4) steps, which were the problematic ones in Fig. 3, become trivial in this case. The experts themselves physically move between the natural and virtual systems and their memories are assumed (by definition) as persistent. However, the entailed steps reflecting the action of laws of nature before, steps (1) and (3) in Fig. 3, are now the open and unentailed sequences of interactive decisions. By definition no algorithmic model is able to encompass the complete sequence as it unfolds (Wegner and Goldin 1999), though they clearly can cover each single step within it (exemplified by a chess computer).

Simulation in algorithmic and interactive problems

In simulation models for algorithmic systems, the main emphasis is devoted to mimicking the non-interactive behaviour of the dynamic system. A typical example from environmental modeling are neural net simulations. Such models do not yield an understanding of internal mechanisms, e.g. of water flow through ecosystems. They have been useful, however, in characterizing the relation between input and output data. (Küppers and Lenhard 2004) argue in a parallel paper that even in this restricted context the heuristic component in the practical building of a simulation model cannot be completely reduced to scientific understanding

This situation changes when interaction is taken into account. For these problems a formalization of behaviour has been proposed, as it is no longer possible to derive it from a closed algebraic system of equations (Goldin et al. 2003). Interactive systems are in a fundamental sense open and thus different from algorithmic computing where the environment is excluded between input and output events. Once an output has been achieved by an algorithm, its internal state must be reset before the next input may occur, hence such algorithmic systems cannot have persistent memory. A Turing machine that is not reset to a predefined initial condition and thus may contain persistent internal states between output and subsequent input events is termed “persistent TM” and has been proposed as a formal model of interactive computing (Goldin 2000).

Persistent states of such a machine are typically inaccessible for an external observer and the only criterion for comparing states is observational equivalence: Bisimulation. The theoretical results in computer science based on this concept suggest that an approach starting from behaviour is the appropriate one for interactive systems, whereas starting from internal states is appropriate for state systems. Formally, the theories derived for interactive systems are coalgebraic duals to the algebras that express the equations of laws of nature.

Undoubtedly, dynamic system theory so far is the dominating approach. Its theoretical framework is fully worked out, whereas for the interactive approach we basically have a number of promising starting points, which are, however, interesting epistemologically. In the last section we will discuss how both approaches could be accommodated into a joint scheme.

MODELING IN SCIENCE AND TECHNOLOGY

We consider the ideal situation that commutativity has been achieved in both Fig. 3 and 4. Robert Rosen has discussed why the first modeling approach is often regarded as synonym with natural science itself, though neither he nor we share that opinion. However, we will use this strong label here to make the difference clear between the two cycles.

Commutativity implies a correspondence between arrows 1 and 3 in Fig. 3 and 4. That is why these arrows (indicated by numbers and colour) are shown in parallel in Fig. 5. In the lower case, we are dealing with exo-observers and non-interactive dynamics. The area indicated as the realm of laws of nature (“Science”) can be regarded as given systems; modeling is inference on it. The open part of this modeling approach (grey arrows) is where ideas and creativity comes in: the parts dealing with encoding and decoding operations. The goal is to express laws of nature in their most simple form as equations.

Once an object has been thoroughly understood, science is able to reproduce the observations. Eventually, science may turn into technology: when further repetitions no longer have the purpose of demonstrating understanding, but rather achieve a desired functionality. The label “science” should thus be read as an indication that it started from a scientific objective for which this modeling approach is typical.

Commutativity in Fig. 4 implies a correspondence between the experts’ response in a virtual training situation and their response to a real/natural system. Here it is the participatory and open parts of the cycle that are shown in parallel. They are linked by cultural norms. Through these norms, overall functions (services) are defined for the expert: getting food, timber, etc. from an ecosystem, being able to win a chess game, flying from A to B are a few examples. This provides the final causation typical for technology (Rosen 1991).

The building and documentation of the model is in this case a more formal procedure. For any documented and achieved service one can ask whether this response has already been covered by the simulation model. The purpose of the model is not to provide the most simple explanation, but the most comprehensive repertoire of all possible behaviours that had already been encountered in the managed natural system. If in our idealised case commutativity has been achieved, then

the model can be regarded as complete. A symptom of completeness is when experts are no longer able to distinguish between the interactive system and its model (“Am I playing a human or a chess program?”). This is the equivalent of the Turing test for interactive systems.

Documentation reduces here to the inclusion of additional potential behaviour in the model, in cases where the last cycle does not confirm the congruence relationship. Application reduces to the training of novices to everything that might happen in this system and judging his/her performance in the training environment.

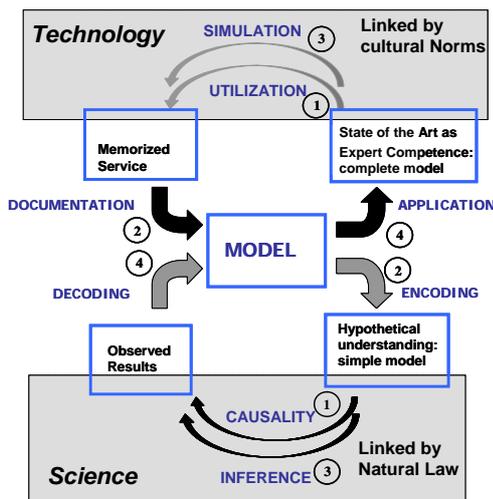


Figure 5: The two modeling approaches combined. In the upper cycle interactive simulation and utilization (e.g. of ecosystem services) is closely related in an idealized (sustainable) Technology. In the lower cycle causal structure in real systems and inferential structure in formal systems are merged by an idealised science. The lower cycle alone is Rosen’s modeling approach which is here extended into the upper half to cover also simulation of interactive behaviour.

When a complete model exists, i.e. the most comprehensive competence is included, the interactive simulation proceeds with the same theoretical rigor as the most simple explanation in the scientific modeling approach.

Chess serves as an example. Complete understanding can be achieved in two ways. The first one is to find a winning strategy. In this case the interactive aspects can be shown as virtual, it will cease to be a game. This corresponds to the scientific modeling approach, where the problem is ultimately reduced to function. The second possibility is to derive a complete model of human heuristics in playing. This program would contain all constraints of human playing, it could be beaten, but no more by any single human, only collectively by all of them (i.e. by another program of the same type). This interactive model would be able to

simulate and master anything that can happen in human chess playing. A novice could be trained to any level by just playing this simulation. This second case shows the technology approach where the heuristics of good playing can never be substituted by scientific understanding. In the contrary who would doubt that the complete model represents some “understanding” of the human way to play the game heuristically? Both cases may still occur in real chess, but the second one seems currently a bit more likely.

EXAMPLES FROM ENVIRONMENTAL MODELING AND ECOSYSTEM RESEARCH

An example of the modeling approach of (non-interactive) environmental science is discussed in the paper by Küppers and Lenhard (this proceedings). That is why we will restrict the examples to cases where living systems are directly involved. We regard ecosystem management as a paradigmatic one. Humans have not fully emancipated from the necessity to utilize ecosystems and it is unlikely that they ever will. In addition science has not achieved an understanding of Life that would allow to reconstruct it by the modeling approach depicted in figure 3 in science from its building blocks. It is also unlikely that this will change in the future. Robert Rosen has argued that there are principle difficulties to be solved.

A political goal addressing this issue is expressed by the notion of sustainability with respect to ecosystem utilisation. One of the best investigated examples is forestry in Europe, where the term sustainability has been used for two hundred years. Under the given environmental conditions and given the range of species in Middle Europe, forestry involves a rotation period of typically a century. Many of the management decisions that are necessary in the course of such a rotation period can only be taken interactively. The current situation has to be judged competently and appropriate action be taken in response to it. We have argued elsewhere that this interactive aspect in silviculture is irreducible, and simulation models of forest growth have thus to incorporate these interactive decisions. (Hauhs et al. 2003).

DISCUSSION AND CONCLUSION

Two complementary approaches for modeling a natural system have been introduced. The first one is widely known and recognized as the prototype of scientific modeling. We suggest that a dual second modeling approach has become possible since the proliferation of interactive computing software, with flight simulation or silviculture as examples.

These two approaches have been sketched as two dual (idealised) cases in a generalised modeling scheme (Fig. 5). An unsolved question is what decides in a given context whether the upper or lower loop is more appropriate. The notion of interactivity is closely

connected with complexity. A system that behaves very regular (too simple) or entirely unpredictable (too random) is no candidate for an interactive relationship. Hence interactive systems must appear as complex to any participatory observer. Like explaining complexity, modeling interactivity is thus a notion that depends on both systems that become connected.

Human culture contains examples in which complex environmental signals were mistaken as indicative of an interactive relationship. One of the most prominent example is the annual flood heights of the Nile. Even by today's standards this has remained among the most complex signals known to hydrologists and its nature is still discussed. This single factor has been responsible for the Egyptian economy through its long history. The annual yield could be estimated by the Nile height and the Nilometer was used to set the tax level for the year.

If a complex signal is erroneously interpreted as resulting from interaction it may block a proper understanding of its causes. However, at the other extreme is the complementary situation when one only tries to reduce an interactive situation to a functional one, but this is impossible in principle. This attitude could be exemplified by somebody focussing on finding a winning strategy in chess instead of learning how to play properly. Environmental research over the last two decades can be placed into this realm (Hauhs and Lange 2004). In the case of climate modeling the scientific option is probably the only feasible one and has been taken very far (Küppers and Lenhard 2004). Ecosystem research as started in response to regional environmental changes such as acid rain is a much more problematic one. Here a plentitude of modeling approaches has been investigated and it is still open what will be the most appropriate approach. In any case, a better understanding about the nature of and the choice in the modeling approaches available will help to clarify the situation.

REFERENCES

- Goldin, D., Smolka, S.A., Attie, P. and Sonderegger, A. (2003): Turing Machines, Transition Systems, and Interaction. UConn BECAT/CSE Technical Report TR-03-2. 31 pp.
- Goldin, D.Q. (2000): Persistent Turing Machines as a Model of Interactive Computation. In: Schewe, K.-D. and Thalheim, B. (Eds), Foundations of Information and Knowledge Systems - Lecture Notes in Computer Science. Springer, 116-135.
- Hauhs, M., Knauft, F.-J. and Lange, H. (2003): Algorithmic and interactive approaches to stand growth modelling. In: Amaro, A. and Reed, D. (Eds), Modelling Forest Systems. CABI Publishing, Wallingford, UK, 51-62.
- Hauhs, M. and Lange, H. (2004): Modeling the complexity of environmental and biological

systems - Lessons of ecological modeling. In: Klonowski, P. (Ed), Nonlinear Dynamics in Environmental and Biological Sciences. PABST Science Publisher, Lengerich, 130-154.

- Küppers, G. and Lenhard, J. (2004): The Controversial Status of Simulations. ?, 5.
- Milner, R. (1989): Communication and concurrency. Prentice Hall, Oxford.
- Mittelstrass, J. (1995): Enzyklopädie Philosophie und Wissenschaftstheorie. Metzler, Stuttgart.
- Rosen, R. (1991): Life itself - a comprehensive inquire into the nature, origin, and fabrication of Life. Columbia University Press, New York.
- Stehr, N. (2001): Models as focusing tools: Linking nature and the social world. In: Storch, H.v. and Flöser, G. (Eds), Models in Environmental Research. Springer Verlag, Berlin,
- Wegner, P. and Goldin, D. (1999): Interaction as a Framework for Modeling. In: Chen, P., Akoka, J., Kangassalo, H. and Thalheim, B. (Eds), Conceptual Modeling: Current Issues and Future. *Lecture Notes in Computer Science*. Springer, Heidelberg, 243-257.

AUTHOR BIOGRAPHIES

MICHAEL HAUHS was educated in forestry and soil science at the University of Göttingen. His dissertation (1985) was on water and ion budgets of a small forested headwater catchment in the Harz mountains. He worked for two years as a postdoc at the Norwegian Institute of Water Research (NIVA) in Oslo. Habilitation in Soil Science in Göttingen (1991). Since 1992 he holds the chair of Ecological Modeling at the Bayreuth Institute for Terrestrial Ecosystem Research (BITÖK, University of Bayreuth).

HOLGER LANGE was educated in physics and did his dissertation in theoretical physics on stability analysis of quantum field theoretic models 1992, University of Dortmund. 1992-2003 researcher at the Bayreuth Institute for Terrestrial Ecosystem Research (BITÖK), University of Bayreuth. Habilitation 1999 in Ecological Modeling, University of Bayreuth. Since 2003 senior researcher at the Norwegian Forest Research Institute (Skogforsk), Ås, Norway. His fields of specialisation are: ecological modeling, hydrology, systems theory, nonlinear dynamics and time series analysis.

**TECHNOLOGY
PROCESSES
AND
OPERATIONS
RESEARCH**

SIMULATION BASED JOB SHOP PRODUCTION ANALYSER

Wolfgang Kuehn
Dep. of Electrical, Information and Media Engineering
University of Wuppertal
D – 42119 Wuppertal
wkuehn@uni-wuppertal.de

Clemens Draschba
SIPOC GmbH
D - 28199 Bremen
draschba@sipoc.de

KEYWORDS

Job Shop simulation, rule based dispatch, automated model generation

ABSTRACT

With the Simulation based Job Shop Production Analyser a powerful analysis tool has been developed by use of modern Java and database technology. This straight forward approach offers the possibility to model the production flow of even complex job shop production systems efficiently. It provides a professional decision base for the daily operative production business. System integration, automated modelling from the database and very fast simulation runs are key features of this approach. As an industrial application for the Job Shop Production Analyser the simulation of a complex semiconductor production plant is shown.

JOB SHOP PRODUCTION

In job shop production the routing and scheduling of jobs is much more complex than in other production systems. The workflow of each production job is given by a job description which defines the required sequence of the necessary production steps. The decision when to schedule a job and which resource shall be used for each production step depends on the job requirements, the actual availability of the resources and other circumstances.

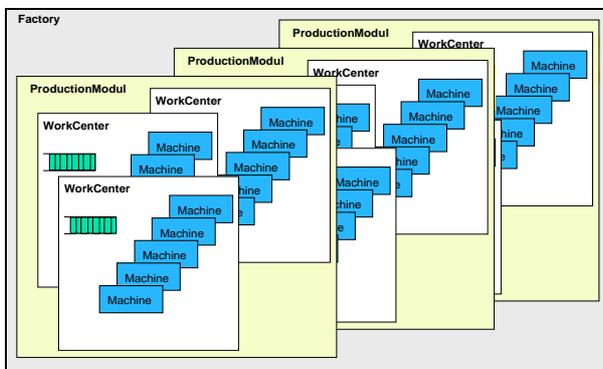


Fig. 1: Job Shop Production

For operative planning and control a rule based dispatch system (RBD) can be applied to calculate the sequence of the jobs for each resource respectively machine. The

production sequence created by such a rule based production planning and control system depends on defined priorities, applied rules and the actual state of the shop.

SIMULATION BASED ANALYSE

The Simulation Based Job Shop Production Analyser offers the possibility of a flexible and efficient approach to model and simulate large job shop production facilities very fast. Therefore the routing and scheduling of the jobs through the production is modelled in detail. The Simulation Based Job Shop Production Analyser offers:

- automatic model generation
- integrated simulation concept
- powerful separated/independent simulation kernel
- integrated database interface
- fast and powerful simulation runs
- fully integrated in a client server environment
- open concept for user specific extensions

The Simulation Based Job Shop Production Analyser is based on modern Java and database technology and can be used for:

- research on future production concepts
- operative production planning and control,

The system can be integrated in existing computer networks for production planning and control. Therefore interfaces for input data, for running and controlling the system and for result output are necessary.

Data Import from PPS – and ERP-Systems

The required data import from any production planning system can be performed directly via database interface or XML interface. The data are processed by means of definable data filter for the simulation. Three categories of input data are required:

Resource data: parameter of production resources, work centres, machines etc.

Operating data: production data, workflow definition, individual process definitions, calendar assignment etc.

Job data: job lists, job dates, priorities etc.

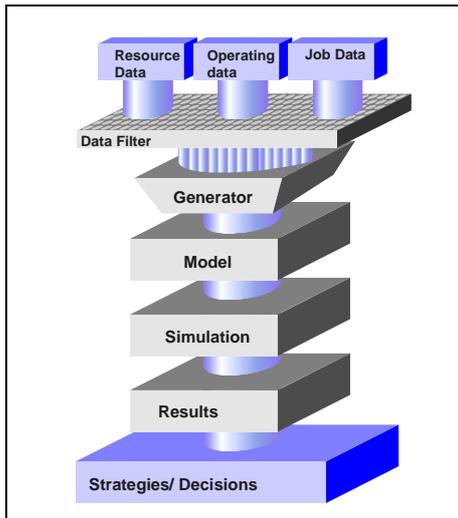


Fig. 2: Production analyse with automatic model generating

The resource data are static data, which are related to the plant structure and production equipment. These data normally are fixed and will change only if the plant structure or the equipment is modified. Operating data are data related to the processes, and provide principle definitions for operating processes. These data are changed, when processes are modified. Job data provide the information of actual jobs. These data are always changed in the daily business. The data described above can be used for an automated model generation.

Automated Model Generation

The automated generation of simulation models is performed with input data from a database. This automatic model generation allows a rapid and flexible modelling approach also of large production areas. Problems from manual data input and data inconsistencies are minimized by this approach. Further a great advantage of this procedure is that the maintenance effort for the simulation is reduced considerably, since the model in each case is directly generated from the current data base automatically again. So the state of the data from the database and that of the simulation model matches always.

If necessary due to customers needs some specific areas may be modelled more in detail. This normally should be performed by use of more specific building blocks. Further for the check of very specific production details it is also possible to modify an automatically generated model manually. However for this option some programming knowledge is required.

SOFTWARE ARCHITECTURE

The first prototype of the Simulation Based Job Shop Production Analyser was realized by use of the simulation tool emPlant (Tecnomatix). For the professional version an advanced Java structure has been chosen. This approach has a lot of advantages.

Native Java Structure

The software architecture of the second generation Simulation Based Job Shop Production Analyser is based on native Java and very flexible. The object oriented features of Java are directly used for the simulation model structure. Towards a very modular and flexible software approach there is a clear separation between the simulation task, the result analysis and the user interface. The simulation model is running as an independent thread. The simulation task is connected by an JDBC interface to a database. The model is created from the content of the database and the required results are written into the database. The user interface interacts with the simulation through defined interfaces. The result analysis can be performed by interaction with the simulation task directly or with data from the database.

In order to speed up the simulation process and the data base interaction the main simulation task is performed independent from the user interface. E.g. on a LINUX server it is even possible to run more than one simulation task parallel and in a second step it would also be possible to simulate different job shop productions at the same time and to interact between these in a distributed simulation network.

Integration in a Client Server Environment

The user interaction for preparing simulation experiments and the analysis of simulation results can be located on client computers somewhere in the network. The model building and simulation process can be started and controlled by interface technology on base of RMI, CORBA or XML-RPC. This offers a fully integration of the simulator into advanced software architectures, e.g. in an application server environment. Such architecture allows an advanced integration of simulation with production planning and control systems.

Due to the advantage that the user interface on the client computers is separated from the main simulation task it can be designed independently to the customers needs. For the operative business the production planner and controller shall be able to operate the software without special knowledge of simulation or programming technology.

Java Simulation Library

The applied Java based Simulation Library is not limited to a certain simulation level like many other simulation tools. The architecture of the library uses an object oriented approach. It is modular and open for additional extensions. The library offers features similar to the flexibility of a simulation language, over the level of general building blocks, such as workstations and assembly stations, up to the efficiency of simulators with highly specialised building blocks for a special purpose.

Job Shop specific Classes

For the Simulation Based Job Shop Production Analyser some specific classes have been developed on top of the general simulation classes. These are special classes for modelling the job shop such as *Shop*, *Department*, *WorkCenters*, *ShopMachines* and *Scheduler*. With these it is possible to build hierarchical models of the job shop facility. Some more special model classes may be added due to specific customer requirements. Further customer specific rules for the routing and scheduling can be integrated into the production control module.

The *Shop* module represents the upper level of a job shop production model. A *Shop* may contain several *Departments* which represent the organisational structure of the facility.

The *Department* module is used to model production areas. A *Department* may contain several *WorkCenters* which are assigned to a certain processing tasks. A *Department* does not provide processing features itself; it serves as organisational unit to group production facilities.

The *WorkCenter* is the module which offers certain processing tasks for the jobs. In a *WorkCenter* there may be several machines located which fulfil the required processing tasks. A *WorkCenter* contains an input and an output buffer. All material flow is sent into the input buffer. For the jobs in the input buffer it has to be decided, which lot shall be produced next on a particular machine respectively resource. This production control decision is done by the *Scheduler* using rule based dispatch algorithms.

The *ShopMachines* are the material flow modules to perform an operation. In a job shop model machines are always located in a *WorkCenter*.

The *Scheduler* is a very flexible production control module which allows to model rule based production planning and control algorithms. A scheduler can be used on different model levels. On the upper model level the *Scheduler* is used to schedule jobs into the system, on level of a *WorkCenter* it is used to decide which job shall be processed next.

New Job Shop Models

New job shop models are generated completely from the database. This is a very powerful and flexible feature towards efficient modelling. Through this approach models can be even defined in third party software products for production planning and control. Further, if necessary the job shop class library can be extended by additional customer required classes in order to fit to the needs of the respective project.

The Simulation Based Job Shop Production Analyser is open for modelling and simulation of any job shop production. However the system has been developed due to the need of complex semiconductor production systems. In the following an industrial application of the

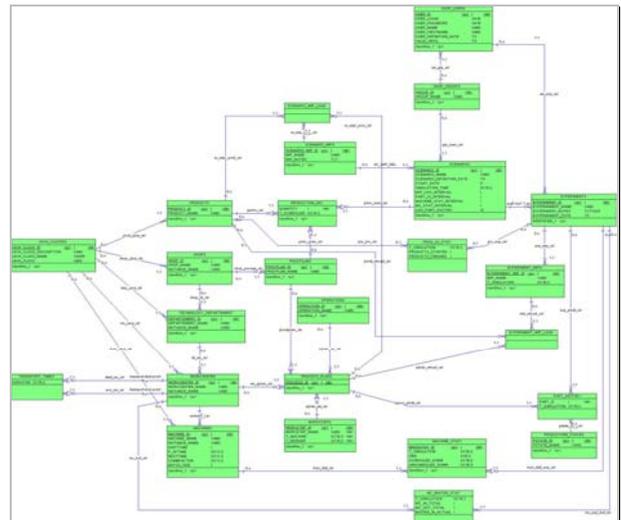


Fig. 3: Database model

Simulation Based Job Shop Production Analyser in the field of semiconductor production shall be shown.

INDUSTRIAL APPLICATION: SEMICONDUCTOR PRODUCTION

The production of semiconductors is a typical job shop production and it is characterized by very extensive processes with a wide span of processing times, even partly with very long machining times, which cannot be interrupted. Due to the complicated processes and the clean room production environment the investment for the plants and even the operation of the shop is very expensive. The necessary work schedules for the processing of the wafers consist of more than 350 single steps. A product may be routed in bows to the same resource up to 20 times again. Therefore the planning and control of the manufacturing chain is extremely complex and requires a very powerful software tool. The modelled production consists of several production areas with more than 100 work centers, more than in total 350 machines as well as more than 500 work schedules for more than 1000 different products. A special characteristic of this particular semiconductor production is the relatively high amount of urgent development lots, which require a high priority and a very fast throughput time, in order to fit the demands of the developers for customer definable wafers.

For this reason the simulation of the evaluated semiconductor production requires a considerable expenditure due to the described complexity of the production. With the aid of automatic model generation from a database hierarchical models can be generated conveniently always from actual production data. In this concept the actual simulator kernel is encapsulated and may run as a separate task on a simulation server. The model creation shall be operated by employees in the field of the production control without particular simulation and programming know-how by use of a convenient user interface. The manual modelling of large size companies is very time consuming and

inflexible. Therefore an automatic model generation offers a considerable advantage. The production data necessary for the simulation are kept in a database, converted according to the requirements and used then for the automatic model generation.

Rule Based Production Control

In a complex semiconductor production with a high number of different products, the production planner and / or operator must decide frequently, how lots and tasks shall be routed and scheduled through the production facility in order to fulfil different production goals at a maximum. The sequence of the lots for the production chain can be generated by use of production strategies according to a rule based job scheduling strategy (RBD = Rule Based Dispatch). A particular production strategy can be built up from several priority classes, filtering and sorting rules, that are defined as default rules in order to improve local production areas or to optimize the global production.

The production planning system used in this semiconductor production offers more than 50 different operation rules for the production control, such as:

EDD, DD (Earliest Due Date)	Next job to process is the one with the earliest due date among the waiting jobs
CR (Critical Ratio)	Next job to process is the one with the least time (time to due date divided by total remaining processing time) among the waiting jobs
SST (Shortest Setup Time)	Next job to process is the one with the shortest setup time
LS, MS (Least Slack, Minimum Slack)	Next job to process is the one with the least time [time to due date minus total remaining processing time] among the waiting jobs
CP (Critical Path)	The Job on the critical path is processed first
Rush (Emergency)	Jobs with special demand to be processed very urgently
...	...

In the daily business the semiconductor manufacturer employs a very few of these rules to the daily operation of the production only. And even these rules are used very conservatively since the production is very complex and has a relatively long reaction time. For the production planner and production controller in the plant it is very difficult to judge the influence of particular rules respectively parameter settings for the entire system. However the goal to reduce e.g. the flow factor in the production requires considerable modifications of the production control. Here simulation offers a powerful tool to test various strategies and modifications without risk for the current production.

USE OF THE SIMULATION BASED JOB SHOP PRODUCTION ANALYSER

The use of simulation technology has the goal to improve the production system by optimizing a combination of production strategies under consideration of the respective status of the job shop facility. For this purpose the Simulation Based Job Shop Production Analyser has been applied. The model generated for the analyses consists mainly of production modules, work centres and machines. Each production module consists of work centres, which itself may contain several job shop machines. With the Simulation Based Job Shop Production Analyser the effect of various parameters to the production process chain can be examined in a dynamic model, such as:

- effect of additional tasks
- changes of scheduling sequence in the production system
- change of the rule based production control
- changes in the workflow sequence
- changes the lot size
- effect of special and hot lots
- changes in the plant structure

Prototype based on emPlant

The first modelling of the semiconductor plant has been performed with the prototype version of the Simulation Based Job Shop Production Analyser, which was developed on base of the simulation tool emPlant (Tecnomatix) and an Oracle Database.



Fig. 4: Hierarchical simulation model

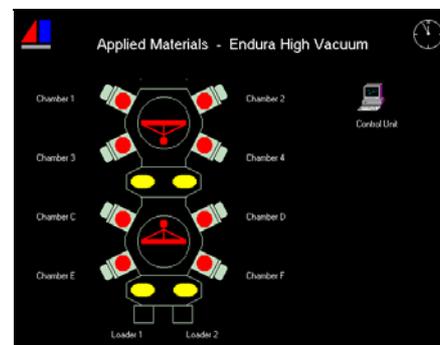


Fig 5: Detailed simulation of the Custer Tools Endura High Vacuum chamber. Meanwhile the simulation tool has been changed in order to improve simulation speed, robustness and the

integration possibilities. The new advanced approach is based on a very efficient Java simulation library.

Improvement by Next Generation Analyser

The second generation Simulation Based Job Shop Production Analyser is based on an advanced Java simulation library. It offers a faster optimized simulator kernel and powerful model classes approving a most extremely efficient modelling and simulation approach.

The library follows an object oriented design. It offers general simulation objects, material flow objects, information flow objects, decision making objects and applied material flow objects. For the internal information flow the features of Java are used directly to implement information flow for the simulation model very efficiently.

With the prototype on base of emPlant and Oracle the model building took some minutes. With the new Java based version the time generating the complete model building process of the plant described above takes now less than 10 seconds. Further due to the very clear structure the simulation speed has been increased by a factor of 10^2 to 10^3 . The database interactions are much more flexible through the use of JDBC.

Operative Planning and Control

The Simulation Based Job Shop Production Analyser can be integrated into the production planning software used for the daily business. The stuff running the plant can apply the simulation results for the operative production planning and control without the requirement of specific simulation and programming knowledge.

The input of data, the change of control strategies etc. as well as the graphical presentation of the results can be adapted by use of modern software tools in a user-friendly way to the respective requirements. The model can be generated automatically from the database. Interaction of the user is not necessary, even not reasonable. The actual simulation process runs in a separate task in the background, this may be even possible through the network somewhere else on a separate simulation server. This integrated approach offers a very powerful and user friendly concept which can be adapted to the respective application very flexibly due to the requirements.

Simulation Results

Results of the simulation can be processed efficiently by use of the database. In freely definable statistic intervals parameters can be gathered and analysed, such as:

- finished lots
- maximum and mean utilisation of input buffers for selected or all resources
- utilization from resources, production areas, work centres, machines

- number, location and processing status of all lots in the system
- WIP and entire flow factor
- standby, scheduled down and unscheduled down of resources
- overall equipment efficiency (OEE)
- additional customer specific measures

Further various measures can be gathered for a product or a product group over a freely definable time period, such as:

- flow factor
- number of finished lots
- number of lots in the production system
- throughput times
- adherence to schedules
- additional customer specific measures

Analyses of the Semiconductor Production

The analyses of the complex semiconductor production described above has been performed with various data sets for specific scenarios. For the customer it was very helpful to get some knowledge about the influence of special lots and hot lots in this particular production. A certain percentage of lots with a high or very high priority have the effect that the large volume main business with about 90% of all jobs will be disturbed very much by 3 to 8 % of urgent jobs. However for the research and development department at least a part of the urgent jobs is necessary. Some of these may be unnecessary if the entire system is running more efficient and the flow factor is reduced anyway.

By use of simulation it can be considered, how this necessary part of very urgent development lots can be routed and scheduled through the production without disturbing the large volume business to much. There is no general rule to improve. However simulation can show for any actual situation, based on the actual state of the system, different possibilities to handle the high priority lots without too much disturbance. The operator is able to check for the best possibility and to run the production system due to the actual situation with a minimum of disturbance.

SUMMARY

For the modelling of rule based job shop production systems a modern simulation concept has been developed. This concept is based on modern Java technology and an automatic model generation with production data from a database applied directly before performing each simulation run. The concept of the Simulation based Job Shop Production Analyser is based on a strict object oriented Java simulation kernel which is designed in a modular approach with open interfaces. In order to offer a maximum on flexibility and robustness the concept allows a fully automated generation of the actual simulation model by use of production data from the database. The Simulation

Based Job Shop Production Analyser allows running simulation models on various hardware platforms and operation systems.

The Job Shop Production Analyser has been successfully applied to model the simulation of a complex semiconductor production plant. Due to its structure the Job Shop Production Analyser can be integrated into existing production planning and control systems. It offers the possibility to model the production flow of even complex job shop production systems efficiently in detail and in order to provide a professional decision base for the daily operative production business.

The comparison between the prototype realised on base of the simulation tool emPlant and the second generation based on Java technology shows a strong improvement. Speed is increased considerably by a factor of 10^2 to 10^3 . The database interactions for input data and for writing results to the database have been improved and are much more flexible. Further the analyser modules can be fully integrated into other production planning and control systems. With the powerful simulation and integration features an advanced analysis tool with great flexibility has been achieved.

REFERENCES

- Balas, E., Vazacopoulos, A., 1998. "Guided Local Search with Shifting Bottleneck for Job Shop Scheduling", *Management Science* 44, 262-275.
- Brucker, P., Neyer, J., 1998. "Tabu-search for the multi-mode job-shop problem", *OR Spektrum* 20, 21-28.
- Burke, E. and R. Kilgore. 2000. "Silk@, JAVA AND OBJECT-ORIENTED simulation". Proceedings of the 2000 Winter Computer Simulation Conference. SCS International, Ghent, Belgium.
- Dauzere-Peres S., Paulli, J., 1997. "An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search", *Annals of Operations Research* 70, 281-306.
- Dauzere-Peres S., Roux J., Lasserre J.B., 1998: "Multi-resource shop scheduling with resource flexibility", *European Journal of Operational Research* 107, 289-305.
- Dudek-Dyduch, E., 2000. "Learning based algorithms in scheduling", *Journal for Intelligent Manufacturing*, (11): 135-143
- Haasw, K. and Kimms, A., 2000. "Lot sizing and scheduling with sequence dependent set-up costs and times and efficient rescheduling opportunities", *International Journal of Production Economics*, 66: 159-169,
- Holthaus, O. and Rajendran, C., 2000. "Efficient job shop dispatching rules: further developments", *Production Planning and Control*, 11 (2): 171-178,
- Jansen K., Mastrolilli M., Solis-Oba R., 1999. "Approximation Algorithms for Flexible Job Shop Problems", *Proceedings of Latin American Theoretical Informatics (LATIN'2000)*, LNCS 1776, pp. 68-77.

Kacem, I., Hammadi, S., Borne, P., 2002. "Pareto-optimality Approach for Flexible Job-shop Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic." *Journal of Mathematics and Computers in Simulation*, Elsevier.

Kacem, I., Hammadi, S., Borne, P., 2002. "Lower bounds for evaluating schedule performances in flexible job shops. Performance Metrics for Intelligent Systems Workshop, PerMIS'02, Gaithersburg, MD, USA.

Kacem, I., Hammadi, S., Borne, P., 2002. Approach by Localization and Multi-objective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems. *IEEE Transactions on Systems, Man, and Cybernetics. Part C*, 2002, Vol 32. N1, pp 1-13.

Kühn, W., 2002. JAVA-Sim - An advanced Discrete Event Simulation Library, SCSC 2002, Summer Computer Simulation Conference, San Diego

Ottjes, J.A. Veeke. H.P.M., 2000. "Production Scheduling of Complex Jobs with Simulation Proceedings of the Business and Industry Simulation Symposium" (ASTC 2000). Washington

Shafaei, R. and Brunn, P., 1999. "Workshop Scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of the scheduling rules in a dynamic and stochastic environment", *International Journal of Production Research*, 37 (18): 4105-4117

Singer, M., 2000. "Forecasting Policies for scheduling a stochastic due date job shop", *International Journal of Production Research*, 38 (15): 3623-3637

Subramaniam, V. et al., 2000. "Machine Selection Rules in a Dynamic Job Shop". *International Journal of Advanced Manufacturing Technology*, 16: 902-908,

BIOGRAPHY

WOLFGANG KUEHN studied mechanical engineering at the University of Brunswick, Germany. Afterwards he worked two years with Blaupunkt. At the University of Bremen he got 1991 his PHD in production engineering and 1997 his habilitation in the area of simulation of production systems. From 1993 to 1995 he worked as Associated Professor at the Asian Institute of Technology in Bangkok. 1996 he founded the SIPOC Simulation based Planning, Optimization and Control GmbH in Bremen. Since 1997 he is working as professor for production planning and control in the department of Electrical, Information and Media Engineering at the University of Wuppertal.

CLEMENS DRASCHBA got 1996 his degree of production engineering at the University of Bremen. He has industrial experience with the Meyer-Werft and AEG-Lloyd-Dynamo-Werke. For the SIPOC GmbH he performed simulation studies of production processes, such as semiconductor production and international catering facilities. Since 1998 he is general manager of the SIPOC GmbH and responsible for software development.

“Simulation Of Three-Dimensional Technical Textiles“

Prof. Dr.-Ing. Alexander Büsgen
 Prof. Dr.-Ing. Karin Finsterbusch
 Dipl.-Ing. (FH) Andrea Birghan

Niederrhein University of Applied Sciences, Department of Textile and Clothing Technology
 Webschulstraße 31, D-41065 Mönchengladbach, Germany

KEYWORDS

CAD-simulation, 3D weaving

ABSTRACT

Within the bounds of a research project, sponsored by the Nordrhein-Westfalen Innovationsfond 2001, will be created an appendix module for a 3D-capable standard CAD-system at the Hochschule Niederrhein. This module should describe and simulate three-dimensional textiles with regard to geometry and material data. The simulated textiles base on the three-dimensional weaving method, Shape Weaving.

INTRODUCTION

Textile fabrics in general

Conventional textile fabrics are two-dimensional. In many cases these fabrics are made of threads (yarns, cabled and folded yarns, monofilaments, wires) or fibres or filaments. Textiles made of threads are woven, knitted and stitched fabrics. Textiles made of fibres or filaments are needled, pads and wool felts. These fabrics are wound onto a cloth beam at the end of the production process.

Conventional design processes

Until now the production of textile fabrics, adapted to a three-dimensional form, were achieved by the following methods:

1. Transforming:
 deep drawing of a two-dimensional textile
limit: drape ability of the 2D-textile



Fig. 1: Transforming

2. Manufacturing:
 cutting/punching out and assembly (through sewing, sticking together etc.)

limit: quality and costs

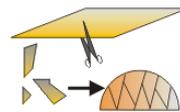


Fig. 2: Manufacturing

3. Fibre spraying:
 application of cut fibres through the spraying method
limit: mechanical load-bearing capacity

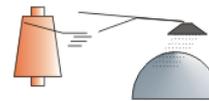


Fig. 3: Fibre spraying

By now there are possibilities to adjust textile fabrics already to a certain shape during their production process: the so-called 3D-textiles. Through this processes the later ready-made is not necessary. Disturbing and eventually quality reducing joints will be avoided. A high strength and form stability of the textile can be guaranteed. The final product has a lower weight and a more homogeneous surface. Furthermore, the lower number of personnel and the clear reduction of cutting waste save costs.

OVERVIEW OF 3D-TEXTILES

Multiply textiles with z-yarns (thick, evenly surfaced)	
Textiles, thin-walled geometries (component geometry)	
Woven volume (f.e. cube with 3D-thread orientation)	

Fig. 4: 3D-textile

SHAPE WEAVING

One process to produce 3D-textiles directly on the weaving machine is the Shape Weaving process developed by the company Shape 3 Innovative Textiltechnik GmbH. The integration of different long warp and weft yarns during the weaving process is the principle of this process. Through these surplus yarn lengths the textile expands into the third dimension.

Problems

The writing of a database for the 3D weaving is very expensive at this stage.

First, it has to be determined which length is needed for every warp yarn. This is done by manual measuring of the prototypes regarding warp and weft yarn lengths. Hence, the quality of a 3D-textile depends largely upon the accuracy of this measurement. In a second step the weave constructions – adjusted to the 3D shape - have to be defined. These will be defined on the basis of experience values. Only by weaving of the whole product the problem areas will be apparent. To control any made changes the complete product has to be rewoven. Until now, only after extensive single trials you got a sufficient result. There is also the fact, that the determined data are not reproducible and the database has to be completely redone by the use of minimal changes (yarn strength, fabric weight etc.)

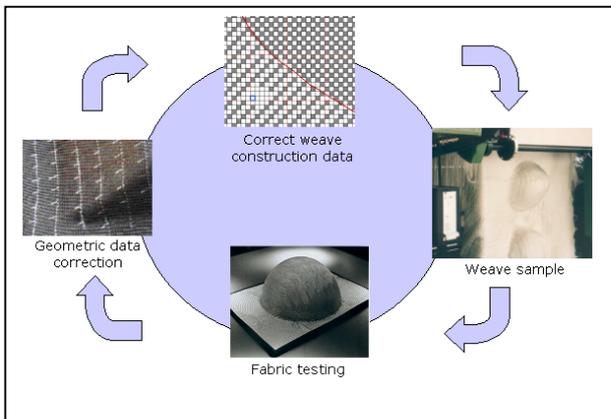


Fig. 5: Previous procedure, optimisation weaving trials

To reduce this expenditure, a CAD-appendix-module to the AutoCAD program was created, with them, in the future, a simulation will replace the weaving trials.

CAD-SIMULATION

Modules



Fig. 6: Menu simulation

1. drawing up of the warp yarns
2. drawing up of the weft yarns
3. drawing up of file with warp yarn lengths
4. drawing up of weave construction file
5. dialogue box to determine the limiting values
6. control and maintaining of the maximum weft and warp yarn distance
7. control of the maximum warp yarn disorientation
8. control of maintaining of the minimal cross angle of warp and weft

Production of warp and weft yarns

By activating the button “drawing up of the warp yarns “ a dialogue box starts, in which the user has to determine the textile data (yarn density, material, order of warp-ends, yarn courses). Hence, the simulation process is detailed shown in the following figures 7 up to 14

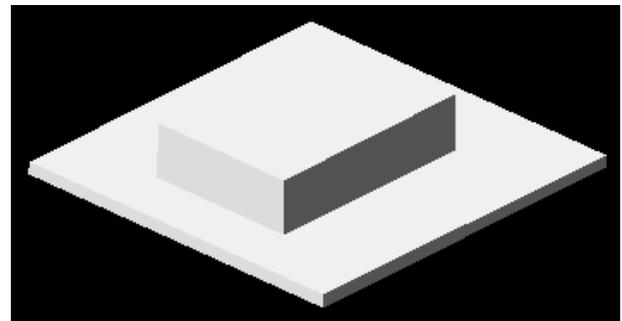


Fig. 7: Component geometry

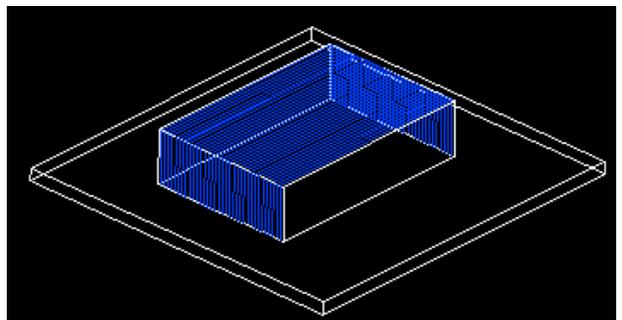


Fig. 8: Component analysis: cross-section in weft direction

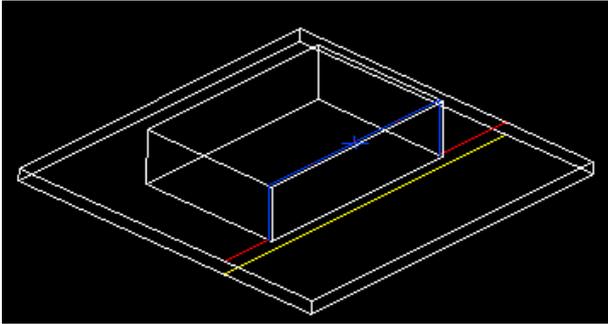


Fig. 9: Guideline for the spreading lays at the widest part of the component.

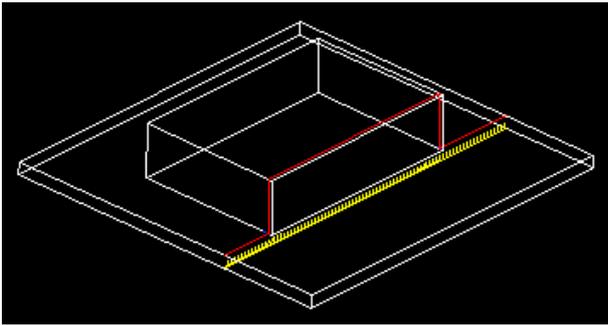


Fig. 10: Place points with the distance of the guidelines in the 2D-area.

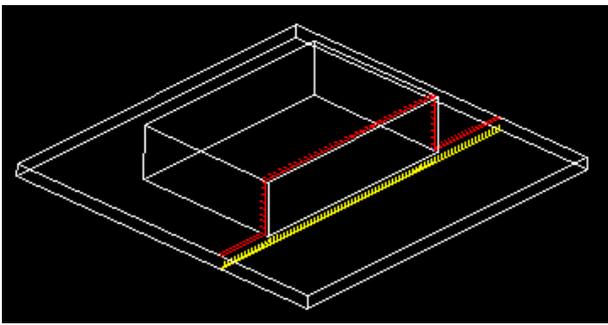


Fig. 11: Transmit 2D-distance in 2D- and 3D-area => spreading

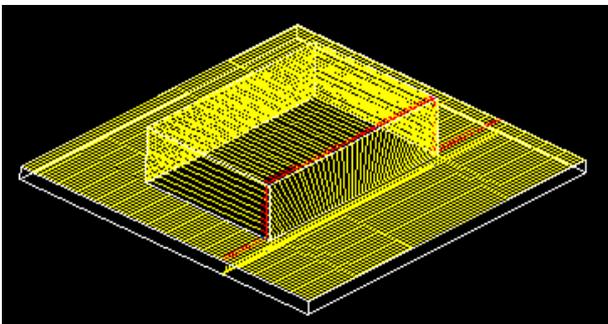


Fig. 12: Spreading of the guidelines

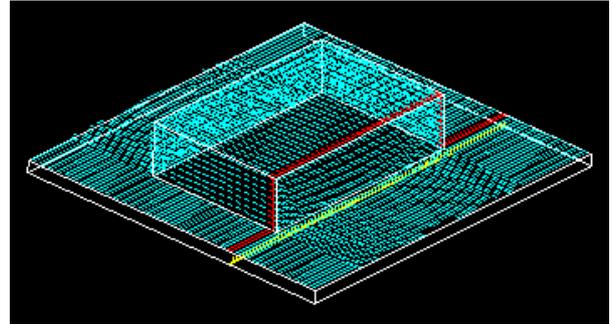


Fig. 13: Placing of the warp yarn points on the guidelines

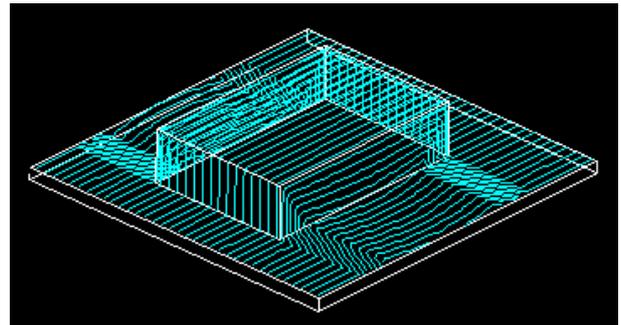


Fig. 14: Warp yarns by connecting the points

The drawing up of the weft yarns takes place in the same way as that of the warp yarns. It will be created guidelines, on which the guide points for the weft yarns are placed. The drawing up of both yarn systems consequently takes place completely independent of each other.

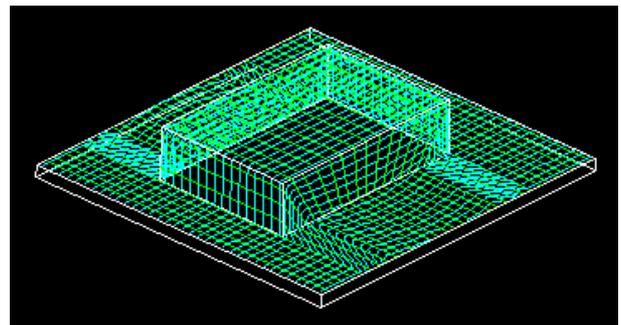


Fig. 15: Parallel warp and spread weft yarns

Examination of the limiting values

The production of a 3D-textile is tied to certain restrictions (limiting values):

- Yarn lengths and yarn distances:
The individual length of a warp and weft yarn, is dependent on the machine and is limited to a maximum value, which has to be determined here. This must not to be exceeded in the textile construction. Furthermore, the slipping strength and the cover factor limit are the biggest or rather the smallest permitted yarn distances.

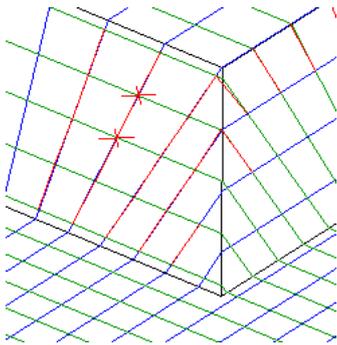


Fig. 16: Schematic depiction yarn distance

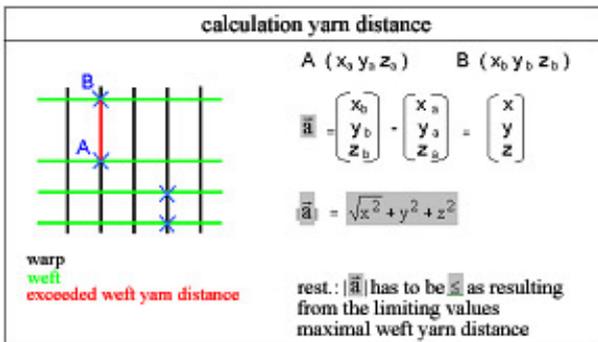


Fig. 17: Calculation yarn distance

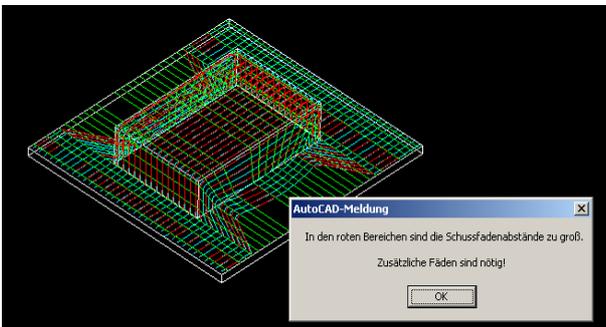


Fig. 18: Screenshot yarn distance

If the maximal yarn distance is exceeded, a red mark on the prevailing place appears in the simulation and the insert of minimal one additional weft yarn is necessary.

- Maximal yarn disorientation:**
 This value describes the change of the yarn orientation (warp = 0°, weft = 90°) both geometry wise and also caused through the user by means of some machine elements. The maximal permitted disorientation depends on the yarn strength, the yarn density and the material, here for example 40°.

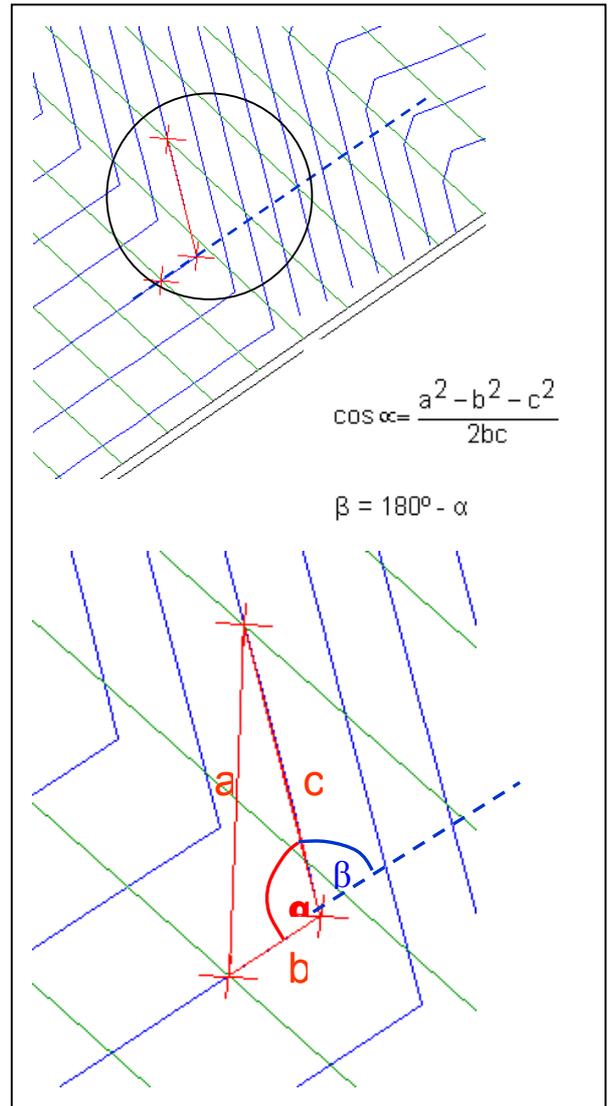


Fig. 19: Schematic depiction yarn disorientation

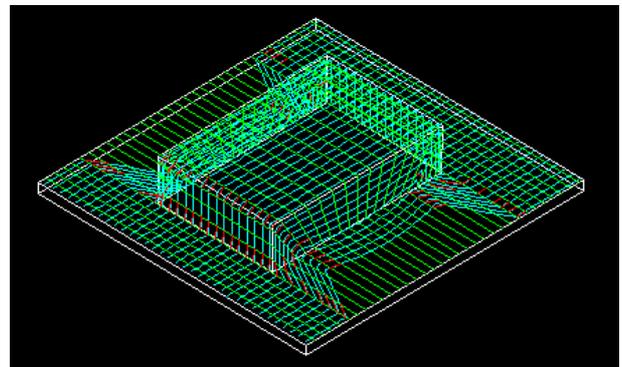


Fig. 20: Screenshot yarn disorientation

Yarns, which create the component without spreading, that means they have in every point the same distance to each other, will be disorientated from their original straight course because of the geometry. Too strong

disorientations lead to heading formation, creases or other fabric faults.

- Deviation of the cross-angles:
Owing to the 3D shape the cross-angles between warp and weft change from 90° to other values. This limiting value depends on the textile construction and has to be redefined depending on yarn strength and yarn density. For example, as limiting value can be tolerated a derivation of 5° for a 67tex Ramie yarn with 12 threads/cm in warp and weft.

Results of the project: comparison of simulation and practice

As an example a rectangle suitcase shell is simulated as 3D-carbon textile. The demands affect especially the visible surface, which as a clear-varnished design area should not contain defects and irregularities if possible. The warp yarns are laid with a constant distance across the whole shape. The weft yarns are planned in the 2D-area of the textile with constant distances and in the 3D-area of the textile with constant spreading, which is brought proportional into line with the surface enlargement.

The result of the simulation shows clearly the problematic areas of the 3D-textile. The surrounding 2D-textile is strongly compressed on both sides of the suitcase shell from the outer edges to the middle. The warp yarns do not anymore orientate themselves in production direction, but incline towards the textile centre. Simultaneously the warp yarn density increases up to 56% of the origin yarn density. It becomes clear, that especially the weave construction has to be designed in the right way to weave the shape satisfactory.

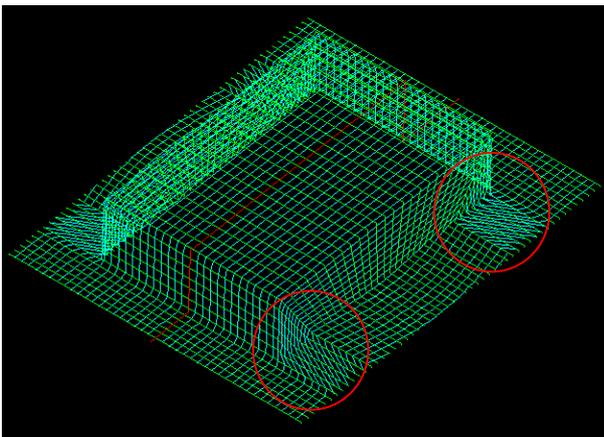


Fig. 21: CAD-simulation suitcase shell



Fig. 22: Result: 3D-woven suitcase shells

The geometry woven to control of the simulation shows in fact the feared defects on the described areas. Because the weave construction was here not adjusted, the textile bulges uncontrolled into the air caused by the increased yarn density outside of the original suitcase shell.

This first practice comparison makes clear, how the 3D textile construction can be improved through the simulation.

CONTINUATION

The current work on the simulation look into the graphical depiction of the weave construction as well as into the combination of weave constructions and yarn densities. Here the right prediction of the cover factors is particularly important. Furthermore, weaving trials should be carried out, which allow a comparison between simulation and practice.

MODELLING AND SIMULATION OF MAGNETIC CONTROL AND ITS APPLICATION ON ALSAT-1 FIRST ALGERIAN MICROSATELLITE

A.M. SI MOHAMMED

Centre National des Techniques Spatiales, Lab. Instrumentation Spatiale.
1, Avenue de la Palestine BP 13 31200 - ARZEW, ORAN, ALGERIE
E-mail : Arezki_s@yahoo.fr

KEYWORDS

Modelling, Simulation, Magnetic, Attitude, Control, Microsatellite.

ABSTRACT

The aim of this paper is the modelling and simulation techniques of a non gravitational force of the earth magnetic field and its application to the stabilisation of ALSAT-1 first Algerian microsatellite build by Surrey Satellite Technology Ltd. (SSTL), Guilford, United kingdom. This paper describes 1) the attitude dynamic, 2) the modelling of the earth magnetic field, 3) the magnetic torquer control. Simulation results will be presented.

INTRODUCTION

Attitude determination provides the information needed for attitude control. Attitude control is the process of changing the orientation of spacecraft. It roughly comprises two areas:

- Attitude stabilisation: maintaining an existing orientation;
- Attitude slew maneuver: controlling the spacecraft from one attitude to another.

However, the two requirements are not totally distinct. For example, the stabilisation of a satellite with one axis towards the Earth implies a continuous maneuver relative to its inertial orientation. The control accuracy typically depends on the actuators and control algorithms.

The limiting factor for attitude control is typically the performance of the actuator hardware and control software. Although with autonomous control systems, it may also be the accuracy of the orbit or attitude information.

An attitude control system is both the process and the hardware by which the attitude is controlled. In general, an attitude control system consists of the following four major components (as shown in Fig. 1):

- Attitude sensors ;
- Control logic ;
- Attitude actuators;
- Vehicle dynamics.

An attitude sensor locates known reference targets such as the Sun or the Earth's centre to determine when control is required, what torques are required, and how to generate them. The attitude actuator is the mechanism that supplies the control torques. Control systems can be classified as either an open loop system in which the control process includes human interactions (e.g. attitude data from the attitude sensors is analysed, and a control analyst occasionally sends command to the spacecraft to activate the control hardware), or a closed loop feedback system in which the control process is entirely electrical or computer controlled (e.g. attitude sensors sends attitude data to an on-board computer which determines the attitude and then activate the control hardware). Further there are two types of attitude control mechanisms: active attitude control in which continuous decision making and hardware operation is required (the most common sources of torques for active control systems are gas jets, electromagnetic coils, and reaction wheels) and passive attitude control which makes use of environmental torques to maintain the spacecraft orientation (gravity gradient and permanent magnets are common passive attitude control methods).

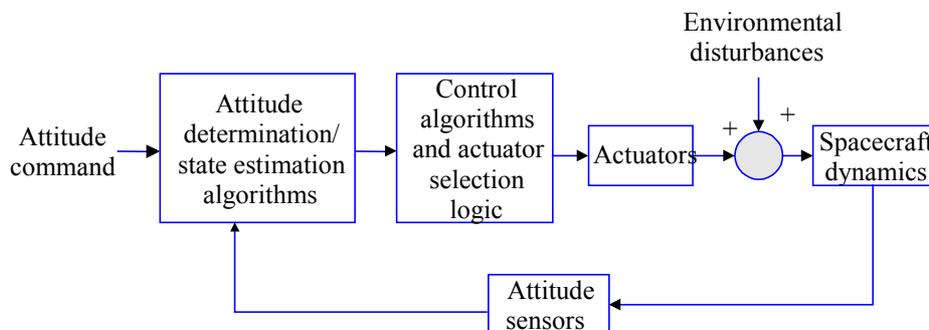


Figure 1 : Schematic Diagram of a Satellite Attitude Control System

ATTITUDE DYNAMICS

The dynamics of the spacecraft in inertial space governed by Euler's equations of motion can be expressed as follows in vector form

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^I = \mathbf{N}_{GG} + \mathbf{N}_D + \mathbf{N}_M + \mathbf{N}_T - \boldsymbol{\omega}_B^I \times (\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}) - \dot{\mathbf{h}} \quad (1)$$

Where $\boldsymbol{\omega}_B^I$, \mathbf{I} , \mathbf{N}_{GG} , \mathbf{N}_D , \mathbf{N}_M and \mathbf{N}_T are respectively the inertially referenced body angular velocity vector, moment of inertia of spacecraft, gravity-gradient torque vector, applied magnetorquer control firing, unmodelled external disturbance torque vector such as aerodynamic or solar radiation pressure.

The rate of change of the quaternion is given by

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}\mathbf{q} = \frac{1}{2}\boldsymbol{\Lambda}(\mathbf{q})\boldsymbol{\omega}_B^O \quad (2)$$

Where,

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega_{oz} & -\omega_{oy} & \omega_{ox} \\ -\omega_{oz} & 0 & \omega_{ox} & \omega_{oy} \\ \omega_{oy} & -\omega_{ox} & 0 & \omega_{oz} \\ -\omega_{ox} & -\omega_{oy} & -\omega_{oz} & 0 \end{bmatrix} \quad (3)$$

$$\boldsymbol{\Lambda}(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (4)$$

Where,

$\boldsymbol{\omega}_B^O = [\omega_{ox} \quad \omega_{oy} \quad \omega_{oz}]^T$ = body angular velocity vector referenced to orbital coordinates.

The angular body rates referenced to the orbit coordinates can be obtained from the inertially referenced body rates by using the transformation matrix A:

$$\boldsymbol{\omega}_B^O = \boldsymbol{\omega}_B^I - \mathbf{A}\boldsymbol{\omega}_0 \quad (5)$$

If we assume the satellite in a near circular orbit with average orbital angular rate ω_0 , then

$\boldsymbol{\omega}_0^B = [0 \quad -\omega_0 \quad 0]^T$ is a constant rate vector.

The kinematic equations can be derived by using a spacecraft referenced angular velocity vector $\boldsymbol{\omega}_B^R$ as follows:

$$\begin{aligned} \dot{\phi} &= \omega_{Rx} \cos \psi - \omega_{Ry} \sin \psi \\ \dot{\theta} &= (\omega_{Rx} \sin \psi + \omega_{Ry} \cos \psi) \sec \phi \\ \dot{\psi} &= \omega_{Rz} + (\omega_{Rx} \sin \psi + \omega_{Ry} \cos \psi) \tan \phi \end{aligned} \quad (6)$$

Where,

$\boldsymbol{\omega}_B^R = [\omega_{Rx} \quad \omega_{Ry} \quad \omega_{Rz}]^T$ body relative angular velocity in any reference coordinate frame.

EARTH MAGNETIC FIELD MODELLING

The earth's magnetic field \mathbf{B} can be expressed as the gradient of a scalar potential function V ,

$$\mathbf{B} = -\nabla V \quad (7)$$

The nature of solenoid to Laplace's equation

$$\Delta^2 V = 0 \quad (8)$$

V can be conventionally represented by a series spherical harmonics.

$$V(r, \alpha, \beta) = r_t \sum_{k=1}^n \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^n A_{mn} P_{nm}(\alpha) \quad (9)$$

Where

$$A_{mn} = g_n^m(\alpha) \cos m\beta + h_n^m(\alpha) \sin m\beta$$

A : Equatorial radius earth (6371.2 Km adopted for the International Geomagnetic Reference Field, IGRF)

g_n^m, h_n^m : Gaussian coefficients ;

R : Geocentric distance ;

α : Coelevation ;

β : East longitude from Greenwich which define any point in space.

MAGNETIC TORQUER CONTROL

Any reaction and momentum wheel 3-axis stabilised satellite must employ a momentum management algorithm to restrict the wheel momentum within allowable limits. Momentum build-up naturally occurs due to the influence of external disturbance torques, for example, the torques due to passive gravity gradient, aerodynamic and solar forces, and active control torques from magnetorquers. These disturbances to the body of an attitude-controlled satellite cause an accumulation of momentum on the reaction and momentum wheels. The added momentum may cause saturation of the reaction and momentum wheel speed. Moreover, the existence of large angular momentum in the satellite causes control difficulties when attitude controllers are implemented, because the momentum provides the satellite with unwanted gyroscopic stability. Therefore, the management of three-axis wheel momentum is required in order to counteract the influence of persistent external disturbance torques.

The following cross-product control law is used to achieve the control objectives stated above

$$\mathbf{M} = \frac{\mathbf{e} \times \mathbf{B}}{\|\mathbf{B}\|} \quad (10)$$

Where

\mathbf{e} : error vector for a magnetorquer cross-product controller;

\mathbf{B} : Magnetometer measured magnetic field vector.

The attitude is obtained from a full state Extended Kalman filter (EKF). This filter take measurement vectors (in the frame body) from magnetometer with 0.3 microTesla noise and sun sensor with 0.1 degree noise and by combining them with corresponding modelled vectors (in a reference frame), estimate the attitude and attitude rate. The EKF estimator is implemented for earth-pointing spacecraft undergoing only small rotation angles. The system model used in this estimator is based on Euler angles, and simplified in order to reduce the complexity and processing time for accommodation on an on-board processor that has limited memory space.

The assumptions of the simplified EKF estimator are listed as follows

- The spacecraft is nominally Earth pointing with either a certain spin rate in Z-.
- The spacecraft has a symmetric structure on X and Y-axes.
- The orbit of the spacecraft is near circular with an almost constant angular rate.

The system noise model has zero mean.

The state vector to be estimated is 6 dimensional such that

$$\mathbf{x} = [\phi \ \theta \ \psi \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}] \quad (11)$$

Using the cross-product control law with the error vector for a magnetorquer cross-product controller implemented on ALSAT-1 is given by

$$\mathbf{e} = \begin{bmatrix} Kd_x \frac{\omega_{ox}}{\omega_0} \\ Kd_y \frac{\omega_{oy}}{\omega_0} \\ \frac{Kd_z}{\omega_0} (\omega_{oz} - \omega_{oz_ref}) + Kp_\psi (\psi - \psi_{ref}) \end{bmatrix} \quad (12)$$

With Kp is the proportional gain, Kd is the derivative gain, $\omega_{0x}, \omega_{0y}, \omega_{0z}$ are the X, Y and Z orbit referenced angular rate of the satellite in radian/second, ω_{0zref} is the reference Z angular rate in radian/second, ψ is the Yaw angle in radian, and ψ_{ref} is the reference yaw angle in radian. The orbit reference angular rate and the angle in Eq. (12) are obtained from a full state Extended Kalman Filter (EKF).

SIMULATION RESULTS

A 98° inclination, circular orbit at an altitude of 860 km was used during the simulation tests. The following matrix of inertia is assumed for Alsat-1 during tests

$$\mathbf{I} = \begin{bmatrix} 153 & 0 & 0 \\ -0.25 & 153 & 0.0005 \\ 0.1 & 0 & 5 \end{bmatrix} \text{kgm}^2 \quad (13)$$

The magnetic moment in the orthogonal X, Y and Z-axes was assumed to be equal to 10 Am² each. The reaction/momentum wheels has a MOI of 8.10⁻⁴ kgm² and the maximum speed is ± 5000 rpm, this gives a maximum angular momentum of 0.42 Nms. The maximum wheel torque is 5 milli-Nm.

We assume that we have gravity gradient torque and aerodynamic torque as external torque.

An IGRF model was used to obtain the geomagnetic field values. A sampling period of TS = 10 seconds was utilised for the discrete filter algorithm. To initialize the filter we use the yaw filter.

The satellite is left to nutate and librate freely for the two orbits in order to converge the filter. At the start of the third orbit the magnetorquer is activated.

The satellite is left to librate freely for the two orbits starting from an initial attitude of 3 degrees roll, 0 degree pitch, 0 degree yaw, 0 degree/second roll rate, 0 degree/second pitch rate and 0.6 degree/second yaw rate. At the start of the third orbit the magnetorquer is activated and within one orbits the pitch and roll librations are damped to nadir pointing error of less than 1 degree, the yaw angle is controlled to 0 degree. At the start of the eighth orbit the yaw angle is commanded to 170 degree for six orbits.

The total accumulated on time of magnetorquer is approximately 13000 seconds during an active control window of 12 orbits (72000 seconds). This gives an average magnetorquer power drain of 0.15 Watt from the start until the attitude is achieved.

We obtain the following results

Table 1: Lists the Euler angles RMS for the last three orbits ($\psi_{ref} = 0.0$ degree).

	Roll [degree]	Pitch [degree]	Yaw [degree]
Average	0.52*10 ⁻²	10 ⁻²	0.88*10 ⁻¹
STD [1-σ]	0.24*10 ⁻¹	0.29*10 ⁻¹	0.19
RMS	0.25*10 ⁻¹	3*10 ⁻²	0.22

Table 2: Lists the Euler angles RMS for the last three orbits ($\psi_{ref} = 170.0$ degrees).

	Roll [degree]	Pitch [degree]	Yaw [degree]
Average	-8.7*10 ⁻⁴	-85*10 ⁻⁴	170.30
STD [1-σ]	0.94*10 ⁻¹	0.38*10 ⁻¹	0.33
RMS	0.95*10 ⁻¹	0.39*10 ⁻¹	170.30

Table 3: Lists the error Euler angles RMS for the last three orbits ($\psi_{ref} = 0.0$ degree).

	Error Roll [degree]	Error Pitch [degree]	Error Yaw [degree]
Average	-0.3*10 ⁻²	0.41*10 ⁻¹	-0.3*10 ⁻¹
STD [1-σ]	0.46*10 ⁻¹	0.18	0.17
RMS	0.46*10 ⁻¹	0.19	0.18

Table 4: Lists the error Euler angles RMS for the last three orbits ($\psi_{ref} = 170.0$ degrees).

	Error Roll [degree]	Error Pitch [degree]	Error Yaw [degree]
Average	11*10 ⁻⁴	0.4*10 ⁻¹	-54*10 ⁻³
STD [1-σ]	0.25*10 ⁻¹	0.17	0.14
RMS	0.25*10 ⁻¹	0.18	0.15

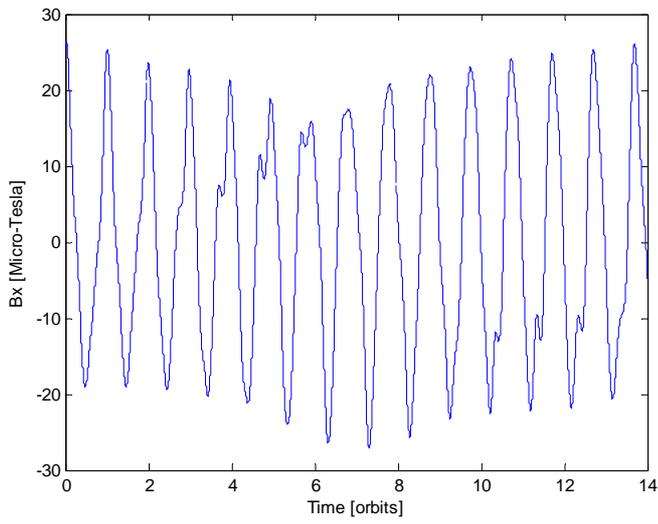


Figure 2 : B-Field (X-axis)

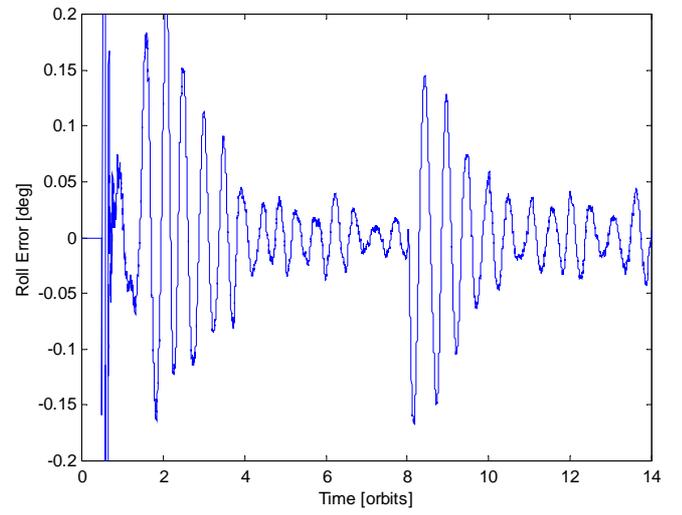


Figure 5 : Roll Attitude error during Magnetorquer Yaw Phase Control

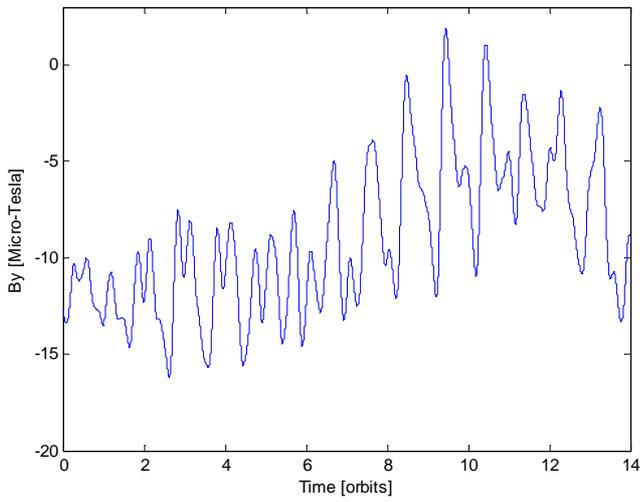


Figure 3 : B-Field (Y-axis)

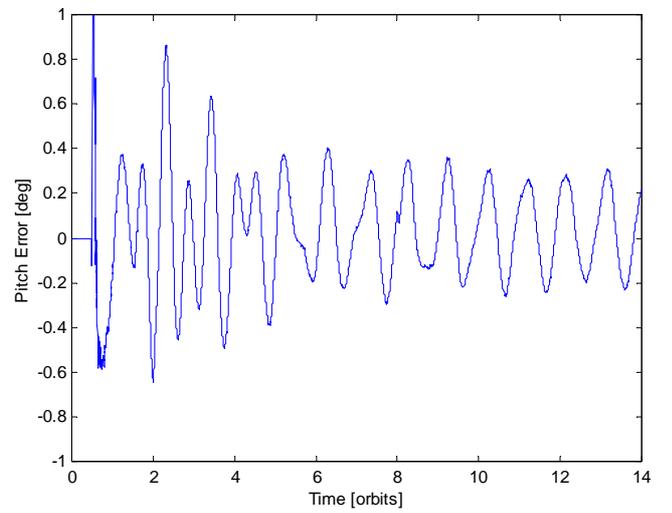


Figure 6 : Pitch Attitude error during Magnetorquer Yaw Phase Control

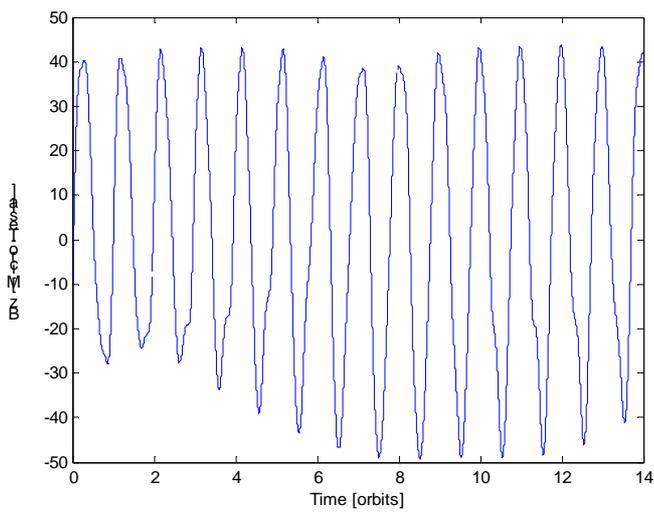


Figure 4 : B-Field (Z-axis)

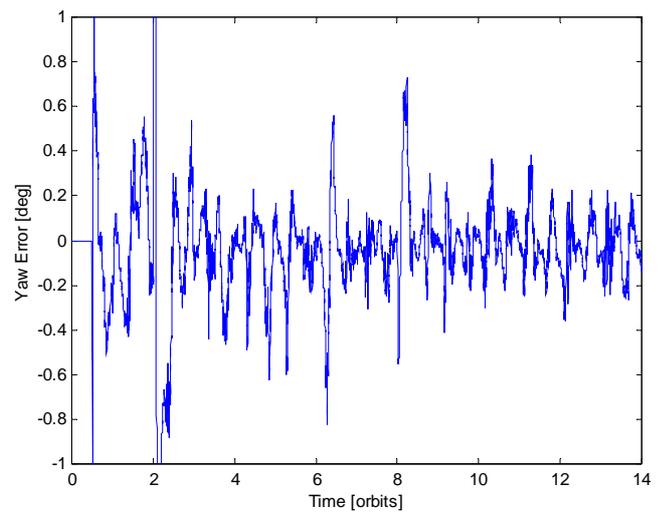


Figure 7 : Yaw Attitude error during Magnetorquer Yaw Phase Control

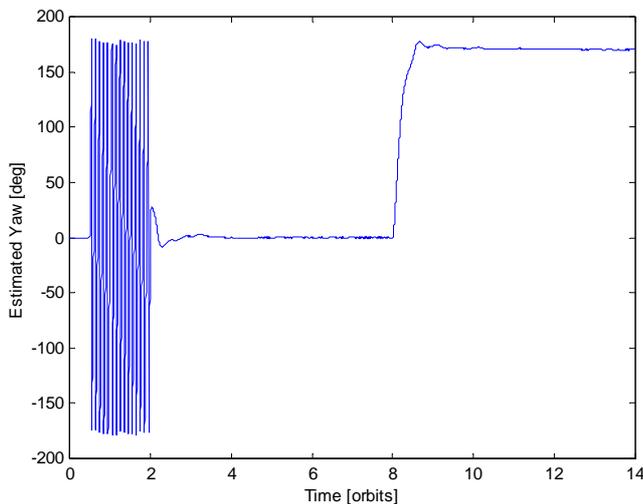


Figure 8 : Yaw Attitude during Magnetorquer Yaw Phase Control

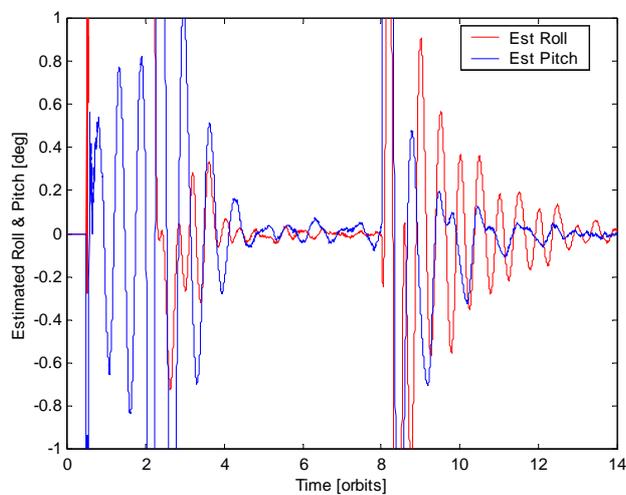


Figure 9 : Roll and Pitch Attitude during Magnetorquer Yaw Phase Control

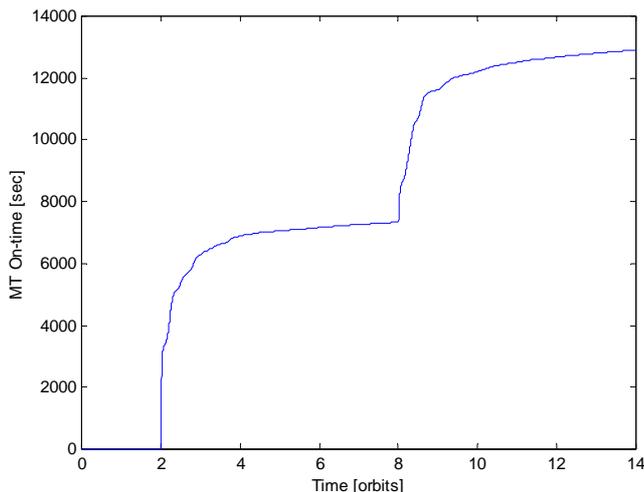


Figure 10 : Magnetorquer on time during libration damping and yaw phase control

CONCLUSION

Alsat-1 attitude requirements are tabulated below

Bore-sight pointing (Roll/Pitch)	≤ 1.0 degree (1σ)
Bore-sight rotation (Yaw)	≤ 0.5 degree (1σ)
Attitude stability (rate) during imaging	≤ 0.005 degree/second (1σ)

The results we have obtained indicate, the roll, pitch, yaw and yaw rate achieves the requirement values by using magnetorquer cross-product.

This controller was designed to keep the microsatellite in accurate nadir pointing attitude. A cross product magnetorquer control law will damp out undesired pitch and roll libration and control either a constant yaw rate or a fixed yaw angle. This control mode will be regarded as the nominal attitude determination control system mode for the microsatellite.

REFERENCES

- F. Martel, P.K. Pal and M. Psiaki, "Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft", *Proceedings of the 2nd Annual AIAA/USU Conference on Small Satellites*, Utah State University, September 1988.
- J.R Wertz, (ed.) "Spacecraft Attitude Determination and Control", *D. Reidel Publishing Company*, Dordrecht, The Nederland, 1985.
- J.J. Levalois et J. Kovalesky, "Géodésie Spatiale", Tome 4 1974, edition Eyrolles.
- M.S. Hodgart and P.S. Wright, "Attitude Determination Control and Stabilisation of UoSAT12", *Journal of the Institution of Electronic Radio Engineers*, Vol. 57 N°5 pp. 151, October 1997.
- O. Zarrouati, "Trajectoires Spatiales", CNES, Edition CEPADUES 1987.
- Vladimir A. Chabotov, "Spacecraft Attitude Dynamics and Control", *Krieger Publishing Company Malabar*, Florida 1991.
- W.H. Steyn, "A Multi-Mode Attitude Determination and Control System for Small Satellite", *Ph.D (Engineering)*, University of Stellenbosch, December 1995.
- W.H. Steyn, Y. Hashida, "In Orbit Attitude and Control Commissioning of the UOSAT-12", *Proceedings of the 4th ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, ESTEC, Noordwijk, The Nederland, 18-21 October 1999, (ESA SP-425, February 2000).
- W.H. Steyn, Y. Hashida, V. Lappas, "An Attitude Control System and Commissioning Results of the SNAP-1 Nanosatellite", *Proceedings of the 14th Annual AIAA/USU Conference on Small Satellites*, Utah State University, September 2001.

SIMULATION-BASED STATISTICAL ANALYSIS OF THE BULLWHIP EFFECT IN SUPPLY CHAINS

Yuri Merkuryev 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

Maris Buikis
Department of Mathematical Statistics
Riga Technical University
1 Kalku Street
LV-1658 Riga, Latvia
e-mail: mbuikis@latent.lv

KEYWORDS

Bullwhip effect, Simulation, Statistical analysis.

ABSTRACT

The paper proposes both statistical and simulation-based analysis and evaluation of the bullwhip effect in supply chains. The demand distortion, called the bullwhip effect, is considered as an important characteristic of supply chain operation stability. A mathematical justification of the stochastic demand as a cause of the bullwhip effect is discussed. Results of simulation studies to analyse the impact of information sharing strategies on the magnification of demand fluctuations as orders move up the supply chain are presented. An approach to measuring the bullwhip effect for the entire supply chain is proposed and practically applied for comparison of different supply chain's configurations.

INTRODUCTION

The bullwhip effect that describes the increase in variability of a demand through the entire supply chain is used, as a supply chain operation stability measurement. The fluctuation of orders across the supply chain is mainly caused by the uncertainty inherent in the system operation environment, such as customers demand and lead times. This means that even small disturbances in demand at the customer level causes the demand amplification for the next supply chain member. It is important to investigate the nature of this effect to avoid holding an excessive inventory, insufficient capacities and high transportation costs. The bullwhip effect characterises the behaviour of the entire supply chain and is used to measure operation effectiveness of the system (e.g., asset/inventory, financial metrics), but not service metrics (e.g. customer response to service, required service level).

The research is aimed to justify the increase in variability in placed orders under the stochastic customer demand using standard statistical tools as well as simulation technique and to propose a measure of the bullwhip effect magnitude for the entire supply chain.

BACKGROUND

Banks and Malave (1984) identify inventory control problems as one of the most frequent area of application

for simulation methodology. They propose six categories of simulation techniques usage assignments in modelling and analysing inventory systems: (1) analytic solution impossible or analytic solution extremely complex, (2) comparison of models, (3) verification of analytic solutions, (4) variance reduction, (5) model validation and verification, and (6) optimisation. A lot of simulation studies are performed in order to handle uncertainty inherent in system operation environment. For instance, simulation is used to enhance operational system decision making in an uncertain environment (Petrovic et al., 1998). Chandra et al. (2001) investigate information coordination influence on the demand forecast accuracy in a supply chain through simulation. Landeghem and Vanmaele (2001) study the behaviour of the supply chain under different sources of uncertainty using the Monte Carlo simulation approach.

In this paper simulation is used to analyse multi-stage, single-item, multi-period inventory system, called a supply chain, from the operation stability point of view. The bullwhip effect that describes the magnification of demand fluctuations as orders move up the supply chain is used as a supply chain operation stability measurement. In this research a background of the bullwhip effect occurrence as a result of the stochastic nature of the customer demand is considered. Simchi-Levi et al. (2000) explains that the increase in demand variability with the necessity for each supply chain stage makes orders based on the forecasted demand of the previous stage. Since variability in placed orders is significantly higher than variability in customer demand the supply chain stage is forced to carry more safety stock in order to meet the same service level. Proposed quantifying the magnitude of increase in variability between two neighbour supply chain stages is expressed as a function of a lead time between the orders receiving and a number of demand observation on which forecast is made:

$$\frac{Var(Q)}{Var(D)} \geq 1 + \frac{2L}{p} + \frac{2L^2}{p^2},$$

where

$Var(Q)$ – the variance of the orders placed by the supply chain stage;

$Var(D)$ – the variance of the demand seen by this supply chain stage;

L – lead time between the orders receiving;

p – number of observation on which further demand forecast is based.

As a result the bullwhip effect is magnified with increasing the lead time and decreasing the observations number. Chen et al. (1998) describe the causes of the bullwhip effect and show the relationship between the increase in variability and forecasting techniques. Metters (1997) concludes that the bullwhip effect affects many businesses in supply chains across a variety of industries. A method of optimal inventory policy calculation in case of stochastic and seasonal demand is proposed and found by a dynamic programming. To aid in the justification of determining causes and remedies, expressing the significance of the bullwhip effect in monetary terms is described. Lee and Padmanabhan (1997) identify ways to control and counteract the bullwhip effect using different information and management technologies, such as electronic data interchange (EDI), computer-assisted ordering (CAO), vendor-managed inventory (VMI), point-of-sale information (POS), etc. The main conclusion of the research is that methods for coping with the bullwhip effect can significantly reduce, but not eliminate it.

In this paper a statistical justification of the bullwhip effect occurrence in the inventory systems with stochastic demand is given. The proposed method allows statistically justify the increase in placed orders variability for the different types of inventory control policies. Simulation offers possibility to model and analyse system dynamic processes under a stochastic demand. Using simulation technique, the bullwhip effect characterised metrics (standard deviation of placed orders for each supply chain stage) is obtained. Experimental studies are performed with the simulation models, and a measure of the bullwhip effect for the entire supply chain is proposed to compare different supply chain's configurations.

In the next section, implementation of standard statistical tools to justify the bullwhip effect occurrence in the inventory systems is described. In further sections the four-stage inventory system is simulated under two information sharing strategies (centralised and desterialised), experimental results are described and an overall bullwhip effect measure for the entire supply chain is proposed. Conclusions follow in the last section.

STATISTICAL ANALYSIS OF THE BULLWHIP EFFECT IN INVENTORY SYSTEMS

Regular or cyclical in nature inventories with additional safety stock are considered. These are the inventories necessary to meet the average demand during the time between successive replenishments and safety stock

inventories are created as a hedge against the variability in demand for the inventory and in replenishment lead time. A method to control such inventories assume that the conditions of demand level, its variability and lead time are known and involves the following main steps:

1. find the current on-hand quantities at the stocking point;
2. establish the stock availability level at the stocking point after the demand satisfaction;
3. calculate total requirements that is the amount of cycle stock plus additional quantities needed to cover the uncertainty in demand;
4. determine an order quantity as the difference between the total requirements and the quantity on hand in case if the on-hand inventory drops below the allowed level when a replacement order should be placed.

The graphical representation of the above mentioned inventory control method is depicted in Figure 1.

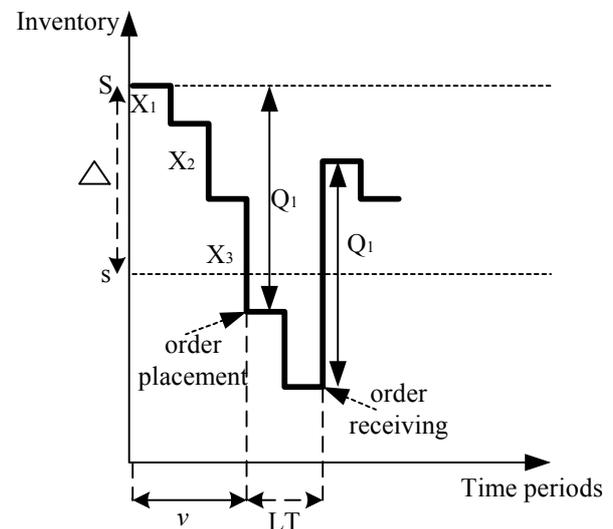


Figure 1: Inventory Control Method

It is assumed that the demand X_1, X_2, \dots, X_i is a discrete random sample observed from some population. Equivalently, these data are independent and identically distributed (IID) observations on some underlying random variable X whose distribution governs the population. Values that numerically characterise the population/distribution, such as an expected value $E(X)$ and a variance $D(X)$ of the discrete random variable X are given.

The inventory level to which inventory is allowed to drop before a replacement order is placed (reorder point level) is found by a formula:

$$s = E(X) * LT + STD(X) * \sqrt{LT} * z, \quad (1)$$

where

LT – constant lead time between replenishments;

$STD(X) = \sqrt{D(X)}$ - standard deviation of the mean demand;

z - the safety stock factor, based on a defined in-stock probability during the lead time.

The total requirements for the stock amount or goal stock level S is calculated as a sum of the reorder point level and a demand during the lead time quantity:

$$S = s + E(X) * LT \quad (2)$$

The order quantity Q_i is demanded when the on-hand inventory drops below the reorder point is equal to the sum of the demand quantities between the order placements:

$$Q_i = X_1 + X_i + \dots + X_v, \quad (3)$$

where

v – random variable, a number of period in which order is placed.

While the demand X is uncertain and implementing such a type of inventory control method, placed order quantity Q is expected to be a random variable that depends on the demand quantities. An expected value $E(Q)$ and a variance $D(Q)$ of the function $Q = \varphi(X)$ are estimated using the following formulas proposed by Feller (1967):

$$E(Q) = E(X) * E(v) \quad (4)$$

and

$$D(Q) = E(v) * D(X) + D(v) * [E(X)]^2, \quad (5)$$

where

$E(v)$ – expected value of a time period in which an order is placed;

$D(v)$ – variance of a time period in which an order is placed.

In described above inventory control method order placement frequency mainly depends on the replenishment lead time and the variance of the demand, since the organisation of the regular inventory stock should cover the demand during the lead time. In suchlike systems the order placement occurs approximately once in lead time period or even infrequently, therefore $E(v) \geq 1$. In case orders are placed once in each time period the variance of placed orders will be equal, but not smaller, than the variance of the demand. However, considering a stochastic nature of the demand, the variation of the order placement frequency depends on the demand variance, the smaller is this variance the more stable is the order placement process. That's why discrete random variable v that determines a period in which an order is placed could not be smaller than one and more likely will be equal to the lead time or even bigger. So far we can conclude that the statistical estimation of the placed

orders variance justify that the variability of placed order $D(Q)$ will be bigger than the variability of demand $D(X)$.

To investigate a probabilistic behaviour of the discrete random variable v is enough to estimate its numerical characteristics (an expected value and its variance). The difference between the goal stock S and reorder point s levels should be established to find a time period when an order should be placed:

$$\Delta = S - s \quad (6)$$

Multi-experimental realisation of the following algorithm:

if $X_1 > \Delta$ THEN $v=1$ AND STOP

ELSE generate X_2

if $X_1 < \Delta$ and $X_1 + X_2 > \Delta$ THEN $v=2$ AND STOP

ELSE generate X_3

...

if $X_1 + X_2 + \dots + X_{n-1} < \Delta$ and $X_1 + X_2 + \dots + X_n > \Delta$ THEN $v=n$

STOP

allows to collect statistics of v values ($v_i, i = \overline{1, n}$) and evaluate their probabilities p_i by relative frequencies \hat{p}_i of their occurrences in performed experiments.

Expected value of random variable is the weighted average of all possible values of the random variable, where the weights are the probabilities of the values occurring. The expected value $E(v)$ of the v value population is estimated by a formula:

$$\hat{E}(v) = \sum_{i=1}^n v_i * \hat{p}_i \quad (7)$$

and its variance $D(v)$ is estimated by a formula:

$$\hat{D}(v) = \sum_{i=1}^n v_i^2 * \hat{p}_i - \hat{E}(v)^2, \quad (8)$$

where

$\hat{E}(v)$ and $\hat{D}(v)$ - experimental estimation of $E(v)$ and $D(v)$ correspondingly.

Described statistical analysis could be implemented in inventory systems that control inventories by the two major policies – reorder point and periodic review methods, and allows to justify demand fluctuation magnification (the bullwhip effect) as orders move up the supply chain in case of stochastic demand.

In the periodic review method, a time period v in which an order is placed could be estimated with more certainty because an order could be placed only in the predefined review periods t . While in reorder point method an order placement could occur in any period.

The proposed statistical approach could also be used to evaluate the bullwhip effect numerically. For this aim a

dependence between a number of period in which order is placed v and realizations of the end demand X_i should be taken into consideration. That is, proposed formula (5) assumes v and X independence, but in the described inventory control system they are dependent in the way of conditional probability of v occurrence $p_v = P(X_1 + X_2 + \dots + X_v > S - s / X_1 + X_2 + \dots + X_{v-1} < S - s)$. The direction of the dependence (positive or negative) should be studied as well.

SIMULATION MODEL OF FOUR-STAGE INVENTORY SYSTEM

Conceptual Model

A four-stage, single-item, multi-period supply chain is considered. The structure of the considered supply chain corresponds to the well known “beer distribution game” where a supply chain consisting of a beer retailer, wholesaler, distributor and factory is simulated (Simchi-Levi et al., 2000). The practical re-order point method – min-max is used for the inventory management. In the min-max inventory control method, a replenishment order will be placed as soon as the inventory level drops below the reorder point (1). The order size is the difference between a target level (2), and the effective inventory level. It is important to remark that replenishment triggering will be based on the effective inventory level, which is the quantity on hand plus the quantity on order minus the unshipped backorders to customers or the quantity allocated to production. The amount of information that is passed on between stages will determine the information sharing strategy. Two types of the information sharing strategies are modelled: decentralised and centralised information. In the supply chain with decentralised information, each stage forecasts a demand based on the orders it gets from the previous stage. Except for the first stage in the

supply chain, there are no stages having access to the end customer demand data. In the second type of the supply chain, the first stage observes the end customer demand and shares all information about its size with the other stages in the supply chain. Other stages in the supply chain can use this end customer demand data to forecast the demand, instead of using the orders they get from the previous stage. All stages use the same forecasting technique – moving average of the demand during the last ten periods.

The inventory management objective is to manage stable operation of the considered supply chain, that is decrease the bullwhip effect. As alternative system configurations supply chain with different information sharing strategies are analysed (Figure 2).

Simulation Model

The system described above has an explicitly dynamic character and simulation is used to capture this behaviour of the system.

It is assumed that end customer demands arrive with fixed time-intervals and their size is variable and is derived from a normal distribution. A constant lead time between all stages is considered. No order processing delay is taken into account, so all demand events are treated immediately by the upstream stage. We also will assume no capacity constraints for the last stage of the supply chain. In this case, stockouts will not lead to lost sales, but to backorders. We thus assume that we have loyal customers.

The simulation model was developed using the ARENA 5.0 simulation modelling environment (Kelton et al., 2002). Evaluation of the inventory control parameters, as well as forecasting procedure was implemented using Visual Basic.

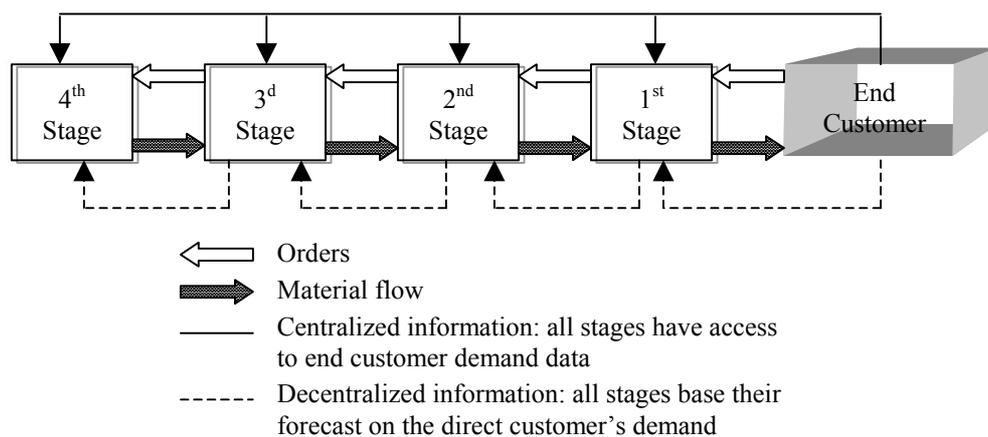


Figure 2: Conceptual Model

EXPERIMENTAL RESULTS

Objective of the experimental studies is to determine the demand variability that occurs at every stage of the

supply chain. The variability in the supply chain can be measured taking into account demand of the previous supply chain stage and orders placed to the next stage of the supply chain. All supply chain stages recalculate and

if necessary place an order in every period and this order, actually, is a demand for the next supply chain member. Since the demand changes every period, the mean and standard deviation can be calculated, on the basis of last p observations.

Performance of the supply chain is evaluated under various factors such as end customer mean demand $E(X)$ and its standard deviation $STD(X)$, lead time LT , safety stock factor z and number of observation on which further demand forecast is based p (Table 3) implementing both decentralized and centralized information sharing strategies.

Table 3: Experimental Design

Factors	$E(X)$	$STD(X)$	z	LT	p
Values	100	30	1.96	2	10

After the determination of the warm-up period, models were run for three replications, each replication lasting for 1,000 periods. As a measure of performance for these experiments the standard deviation of demand at each supply chain stage is calculated and results are presented in Figure 3.

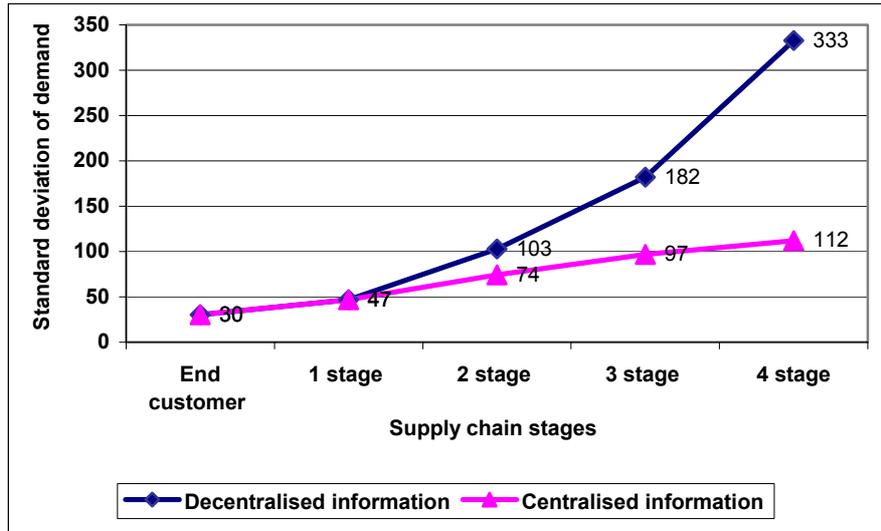


Figure 3: Standard Deviation at each Supply Chain Stage

Experimental results show that the bullwhip effect is present in both supply chain configurations. For the supply chain with centralised information sharing, the variation of placed orders is visibly smaller, but to give an indication of the seriousness of the bullwhip effect an overall bullwhip effect measure should be considered.

OVERALL BULLWHIP EFFECT MEASURE

The propagation of instability up the supply chain is proved by analytical study, the same is by simulation study. Therefore, it makes sense to quantify the bullwhip effect; that is to quantify the increase in demand variability that occurs at all stages of the supply chain. Bullwhip effect measure over the entire supply chain allows compare different system configurations from the stability point of view.

To identify the bullwhip occurrence at each stage of the supply chain it is proposed to compare a standard deviation of demand faced by the neighbour supply chain stages by calculating a ratio BE_i :

$$BE_i = \frac{STD(Q_i)}{STD(Q_{i-1})} \in (0, \infty), i = \overline{1, n} \quad (10)$$

- if $BE_i > 1$ then the bullwhip effect exists;
- if $BE_i \leq 1$ then the bullwhip effect doesn't exist;

where

n – number of supply chain stages;

$STD(Q_i)$ - standard deviation of orders placed by stage i to its supplier;

$STD(Q_{i-1})$ - standard deviation of demand received by supply chain stage i .

In case of supply chain stages do not perform the same operation strategy bullwhip effect between some stages could be eliminated. Therefore to evaluate a magnitude of the demand variability increasing over the entire supply chain two approaches could be offered. The first approach is to calculate overall increasing in demand variability taking into consideration ratios BE_i of each supply chain stage in respect of bullwhip existence. The second approach proposes that only the operation performance measures (BE_i ratios) of stages where the bullwhip effect is present should be taken into account. In this paper the second approach will be considered in details.

To calculate the overall measure of the bullwhip effect for the entire supply chain the average increase in variability for each supply chain stage should be determined. There are two the most frequently used ways to manage it (1) to calculate a geometrical mean or (2) to calculate an arithmetical mean of all observed BE_i

ratios that correspond to increase in variability between stages.

Calculating the geometrical mean of the BE_i ratios by a formula (11) determines the existence of the bullwhip effect between first and last supply chain stages. Values of orders placed by stages $i, n-1$ are the demand received by stages $i+1, n$ and they are cancelled performing multiplication operation. As a result the increase in variability of demand between the first, i.e. end customer and the last supply chain stages is found:

$$\begin{aligned} \overline{BE}_g &= \sqrt[n]{BE_1 * BE_2 * \dots * BE_n} = \\ &= \sqrt[n]{\frac{STD(Q_1)}{STD(Q_0)} * \frac{STD(Q_2)}{STD(Q_1)} * \dots * \frac{STD(Q_n)}{STD(Q_{n-1})}} = \\ &= \sqrt[n]{\frac{STD(Q_n)}{STD(Q_0)}}, \end{aligned} \quad (11)$$

where

$STD(Q_0)$ – standard deviation of placed orders by the end customer;

$STD(Q_n)$ – standard deviation of placed orders by the last stage of the supply chain.

The bullwhip effect existence in intermediate stages of the supply chain is not taken into account.

The above described property could be eliminated by performing addition operation, so all BE_i ratios, which determine the existence of the bullwhip effect will be taken into account. Calculating an average increase in variability demand of all supply chain stages, i.e. arithmetical mean of the increase in variability between all stages could be found by a formula:

$$\overline{BE}_a = \frac{\sum_{i=1}^n BE_i}{n}, \text{ for } BE_i > 1 \quad (12)$$

The proposed bullwhip effect overall measure allows adequately determine the stability of the entire supply chain considering only the situations when variability of the demand increases. The proportional splitting of the increase in variability between all supply chain stages makes possible to analyse different supply chain structures and configurations.

The standard deviation of the demand at each supply chain stage for the both supply chain configuration is presented in Table 4.

Calculated values of the ratio BE_i (10) for all four supply chain stages identify that the increase in demand variability is present in all stages (Table 5). The measure of the bullwhip effect over the entire supply chain is calculated as proposed in (11) and (12).

The supply chain with centralised information sharing strategy is more stable, a destabilisation effect of increasing the volatility of demand as it passes through the chain is smaller than in the supply chain with the decentralised information. This result is derived from comparison of the both overall bullwhip effect measures \overline{BE}_a and \overline{BE}_g .

The smaller is the value the less significant is the increase in variability as we travel up in the supply chain. Since there is a measure for the bullwhip effect over the entire supply chain the difference between the stability (\overline{BE}_a) for both supply chain alternatives could be expressed in percentages – variation of demand in the supply chain with centralised information is by 23% smaller than in the supply chain with decentralised information.

Table 4: Simulation Results

Supply chain alternatives	Standard deviation of the				
	End customer demand	1 stage demand	2 stage demand	3 stage demand	4 stage demand
Decentralised information	30	47	103	182	333
Centralised information	30	47	74	97	112

Table 5: Overall Bullwhip Effect Measure

Supply chain alternatives	BE_1	BE_2	BE_3	BE_4	\overline{BE}_g	\overline{BE}_a
Decentralised information	1.55	2.20	1.77	1.83	1.82	1.84
Centralised information	1.55	1.59	1.31	1.16	1.39	1.40

CONCLUSIONS

The statistical analysis of the demand distortion phenomenon in inventory systems with a stochastic demand has been performed. Performed statistical analysis allows analytical justification of the increase in variability between received demand and placed orders in inventory systems. Experimental studies show that the probability of the time period in which an order is placed is depended from the customer demand.

Analysis of the placed order variability for the all considered situations shows that even a small variation of the mean demand cause an increase in variability of the placed orders. The bigger is the initial value of the demand variation the more significant magnification of placed orders fluctuation will be observed.

The appropriate simulation model of single-item, four-stage, multi-period inventory system is developed to determine the demand variability that occurs at every stage of this supply chain. An impact of information sharing strategies on the bullwhip effect magnitude is evaluated through simulation.

To evaluate a magnitude of the demand variability increasing over the entire supply chain the bullwhip effect overall measures is proposed. The proposed bullwhip effect overall measures could be used to compare the operation stability of supply chains with different supply chain configurations, under different inventory control policies, etc.

Simulation studies indicate that a demand distortion as we travel up the supply chain with centralised information is less significant than in the supply chain with decentralised information, because of for a decentralised supply chain takes much longer time to react to the changing demand. However, different other factors should be taken into account as well for coping with the bullwhip effect, such as lead times, forecasting techniques, etc. in order to significantly reduce it in the supply chain.

REFERENCES

- Banks J. and Malave C. O. "The simulation of inventory systems: An overview". *Simulation Councils, Inc.*, June, 1984, pp 283-290.
- Chandra C., Chilov N. and Smirnov A. "Analysis of supply chain bullwhip effect using simulation modeling". *CONSA Research Report Series "Simulation: Applications, Research and Education in the Baltic Area"*, Presentations at the Special CONSA Conference at Linkoping University, 2001; Nr 5, June, Linkoping University Sweden.
- Chen F., Drezner Z., Ryan J., and Simchi-Levi D. 1998. "The bullwhip effect: managerial insights on the impact of forecasting and information on variability in a supply chain".
- Feller W. "An introduction to probability theory and its applications". New York John Wiley&Sons, Inc., London Chapman&Hall, Limited, Vol. 1, second edition, 1967.
- Kelton W.D., Sadowski R.P. and Sadowski D.A. "Simulation with Arena". McGraw-Hill Inc., New York, USA, 2002.
- Lee H.L. and Padmanabhan V. 1997. "The bullwhip effect in supply chains". *Sloan management review*, No.38(Mar), 93-102.
- Metters R. "Quantifying the bullwhip effect in supply chains". *European Journal of Operational Research*, 1997; Vol. 15, pp 89-100.
- Petrovic D., Roy R., and Petrovic R. "Modeling and simulation of a supply chain in an uncertain environment". *European Journal of Operational Research*, 1998; Vol. 109, pp 299-309.
- Simchi-Levi D., Kaminsky P., and Simchi-Levi E. "Designing and Managing the Supply Chain". McGraw-Hill Companies, Inc, USA, 2000.
- Van Landeghem H. and Vanmaele H. "Robust planning: a new paradigm for Supply Chain Planning" In: *Proceedings of the 15th European Simulation Multiconference*, June 6-9, 2001, pp 866-873.

AUTHOR BIOGRAPHIES

YURI MERKURYEV is Habilitated Doctor of Engineering, Professor of the Institute of Information technology at Riga Technical University, Head of the Department of Modelling and Simulation. His professional interests include a methodology and practical implementation of discrete-event simulation, supply chain modelling and management, and education in the areas of modelling, simulation and logistics management. He is Programme Director of the Master-level curriculum "Industrial Logistics Management" at Riga Technical University. Prof Merkuryev has wide experiences in performing research and educational projects in the simulation area, at both national and European levels. He regularly participates in organising international conferences in the simulation area. He is Track Co-Chair for "Simulation in Logistics, Traffic and Transport" at the annual European Simulation Symposium, General Chair of the 2005 European Simulation Multiconference, to be held in June, 2005 in Riga, Latvia. Prof. Merkuryev has about 180 scientific publications, including 2 books. He is a Board member of the European Council of the Society for Modeling and Simulation International, President of the Latvian Simulation Society, and Board member of the Latvian Transport Development and Education Association.

BUSINESS PROCESS REENGINEERING AT THE HOSPITALS: A CASE STUDY AT SINGAPORE HOSPITAL

Arun Kumar and Linet Ozdamar
School of Mechanical & Production Engineering
Nanyang Technological University
50 Nanyang Avenue, SINGAPORE-639798
Email: makumar@ntu.edu.sg

KEYWORDS

Business process reengineering, Healthcare.

ABSTRACT

As health care costs increase, there is a need for healthcare service providers to look for ways to contain costs and to achieve a higher efficiency at their operating facilities without sacrificing quality. This paper studies a case in employing business process reengineering techniques on one aspect of a health care service – surgical work. The system is simulated focusing on the processes that contribute to the effective functioning of an operating theatre.

INTRODUCTION

Business process reengineering (BPR) has become increasingly important in recent years. Customers now have the choice of different product and service providers, to provide them with the same core product or service that they want. Over the last fifteen years, companies have been forced to reengineering their business processes to stay competitive because customers are demanding better products and services. Improving and redesigning business processes is paramount for businesses to stay competitive.

With the escalating health care costs, healthcare service providers in Singapore are also continuously seeking ways to stay competitive and provide quality service to the customers. Little research has been done on the employment of BPR in healthcare systems. Healthcare industry has traditionally emphasized on breakthroughs in operating procedures and technology in the bid to stay competitive. Healthcare service providers are beginning to understand that BPR initiatives could be a better solution to achieving competitive advantage.

The operating theatre suite is a critically important segment of any healthcare organization

that delivers surgical care to patients. It can consume multitudes of resources, but at the same time can generate significant revenue if managed properly. The conflict between the national goal of healthcare and the high cost of surgical operations is a powerful incentive to improve the quality of management of the surgical suite. For this reason, many hospitals are reengineering their operating theatre processes in an effort to establish, restore or boost profitability while retaining quality (Harris and Zitzmann 1998; Gabel et al. 1999). Reengineering techniques enable healthcare service providers to take a careful look at the processes involved within the organization, identifying redundancy and inefficiency that can be removed from the system. This research employs the concept of BPR to improve the efficiency and effectiveness of certain processes involved in surgical operations.

This paper intends to explore the possibilities of cost containment/reduction in a particular aspect of the healthcare industry with the application of BPR. A simulation model has been formulated to reduce any inefficiencies or bottlenecks inherent in the system under study. The scope of this research is limited to an operating theatre suite within a hospital.

LITERATURE REVIEW

The aggregate per capita healthcare expenditure in Singapore has risen consistently for the last three decades from about S\$150 in the 1960s to S\$800 in the 1997 (Tan and Chew 1997). The healthcare industry in Singapore, like its global counterparts, has been facing tremendous pressures since the turn of the last century. The challenges faced by the industry in the near future are as follows.

The accelerated population ageing will have serious implications to the provisions of health care for the elderly population who will occupy

most of the hospital beds with a low turnover rate. Moreover, the entry of more private-sector hospitals and medical service will lead to more attractive opportunities to health care professionals (Zhang 2001).

There is a lack of health care professionals in Singapore. The local doctor-to-patient ratio was 140 doctors for every 100,000 of the population for the year 2000. According to OECD data, the average ratios for the decade of the '90s for Australia and New Zealand were 240 and 218 respectively (Wee 2002).

Business Process Reengineering in Healthcare

Managers use process reengineering methods to discover the best processes for performing work, and that these processes be reengineered to optimize productivity (Weicher et al. 1995). Hammer and Champy (1993) state that BPR refers to the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality and speed. Business processes are sequences and combinations of activities that deliver value to a customer (Coulson-Thomas 1996). A core business process usually creates value by the capabilities it gives the company for competitiveness. A limited number of such core business processes can be identified in any company, and enhancing those processes can lead to business improvement.

Over the last few years, the reengineering concept has evolved from a "radical change" to account for the contextual realism (Caron et al. 1994, Earl 1995). Davenport and Short (1990) prescribe a five-step approach to BPR. They argue that process reengineering requires taking a broader view of both IT and business activity, and of the relationships between them. The rhetoric of BPR also encourages fundamental step, or frame-breaking change (Coulson-Thomas 1996). BPR is increasingly recognized as a form of organizational change characterized by strategic transformation of interrelated organizational sub-systems producing varied levels of impact. This organizational change perspective recognizes that business process reengineering is not a monolithic concept but rather a continuum of approaches to process change (Kettinger et al. 1997). The faster the speed of change the more difficult and stressful it is to manage (Edwards and Walton 1996).

With 80 percent of the expenses tied to patient care activities, hospitals and healthcare systems can garner substantial savings and improve clinical practices by better managing their labor, supplies, equipment, and facilities. The benefits of reinventing hospitals hold the tangible and realistic promise of radically reducing cost while dramatically increasing the quality of care provided (Harmon 1996).

A case study at Karolinska Hospital in Sweden by Jacob (1995), and Hout and Stalk (1993) reveals that rising costs and a weakened economy in 1990s were forcing the government to reassess and reduce health care expenditures. Karolinska followed Boston Consulting Group's (BCG) Time-Based Management methods to reengineer the way work was done. BCG reorganized work at the hospital around patient flow by creating a new position of "nurse coordinator" in most departments. By redesigning operating procedures and staffing patterns, Karolinska was able to cut the time required for preoperative testing from months to days, close 2 of 15 operating rooms and still increase the number of operations per day by 30 percent.

Operating theatre management often involves human resources, information systems, finance, physical plant design and utilization, capital equipment, clinical quality and efficiency and regulatory (Merriam-Webster 2002). Furthermore, surgical cases are conventionally classified into elective and emergency. An elective case is one whereby the patient can wait at least three days without sustaining morbidity or mortality. A surgical group comprises of several surgeons who share allocated operating theatre time. The term *block time* is the time allocated to each surgical group into which only the surgeons belonging to that surgical group can schedule their patients.

Managing operating theatre suites is a difficult task, because individual theatres and the entire suite are highly complex and tense environments. Many personnel working in the suite are not under the direct control of the operating theatre manager. The operating theatre schedule sets the stage for the daily flow of patients and staff. Once the day starts, however, deviations from this schedule are frequent and expected. Emergency cases must be accommodated, cases may be longer or (rarely) shorter than scheduled, patients may be late or fail to arrive at all, and personnel

may call in sick or become ill during the course of the day (Gabel et al. 1999).

Modern operating theatre management requires an information system that includes an effective scheduling system. Such a system has two basic but critical functions: performing the actual scheduling of cases, which involves finding out the time available on the schedule, whether that time occurs in a surgeon's specific block time, and to facilitate intelligent management of resources. It must provide data on how resources are being used in relation to their availability (Harris et al. 1998). Block scheduling assigns a surgeon (or a surgical group) a block of time that is exclusively for his cases.

The anaesthesia service is often a separate department; in some hospitals it is a division under surgery department. In contrast with surgical sub-specialties, anaesthetists specializing in specific clinical areas such as pediatric anaesthesia, obstetric anaesthesia and cardiac anaesthesia are not typically organized into distinct departments. The anaesthesia department must be organized in such a way as to ensure availability of a sufficient number of anaesthesia providers for elective and emergency cases, which requires 24-hour-a-day coverage (Gabel et al. 1999).

Simulation in the Health Care Industry

The health care industry is a dynamic system with complex interactions, in which the simulation technique would play an indirect but vital role to achieve the optimal result (Zhang 2001). Kelton et al. (1998) state that the real power of the simulation technique is fully realized when it is used to study a complex system. Numerous healthcare service providers such as D. R. Hospital in North Carolina, and St John Hospital in Detroit, U.S.A. have successfully employed the simulation technique to help them in understanding their processes and to optimize them (ProModel Corporation 2002).

CASE STUDY

The Department of Surgery at the Singapore Hospital oversees the operations of the surgical theatres. The main operating theatre complex at Block 3 of the hospital grounds is where surgical operations of different specialties take place. The local demand for surgery services has increased over the last two decades. The capacity of the operating theatres at the complex has reached

high levels of utilization, and action is necessary to ensure that the department is able to cope with increasing patient load. Due to the increasing demand by patients on the services provided by this operating theatre complex and the acute shortage of manpower in the local health care industry, the Department of Surgery has to employ reengineering practices to achieve more efficient and effective utilization with its existing resources.

There are a total of 21 operating theatres at the main OT complex at Block 3 of the hospital. In the year 2000, the number of surgical operations conducted at the hospital was 59,377, of which about 45% were outpatient (day) surgeries. The daily average was 162. Out of the 21 theatres, 19 are allocated for elective surgery and operate 8 hours a day (from 8:30 to 17:30), and the remaining 2 are employed as emergency operating theatres and operate 24 hours a day. Historical data was extracted from the hospital's scheduling database for the period January to September 2001. The data includes the percentage utilization of all the operating theatres, and the surgeons' log of all the surgical operations conducted within the same period.

Every day, each operating theatre is reserved for a specific clinical discipline to carry out surgical operations. Some of the operating theatres are exclusively reserved for a particular discipline, whereas others may be used by different disciplines for each day of the week.

MODELING OF THE OPERATING THEATRE COMPLEX

MedModel is a simulation-based powerful software tool for evaluating, planning or re-designing hospitals and other healthcare systems. It provides a basis for the comprehensive evaluation of large and complex health care systems. MedModel is also equipped with an impressive collection of pre-programmed constructs. Before a model for the operating theatre complex can be developed, a flow chart of the operating theatre process is provided in Figure 1 to illustrate the entities, resources and locations involved. Figure 2 shows the layout of the completed simulation model. The proportion of elective surgical operations for each clinical discipline varies greatly.

To keep simulation as simple as possible, this model deals with only 8 operating theatres, each

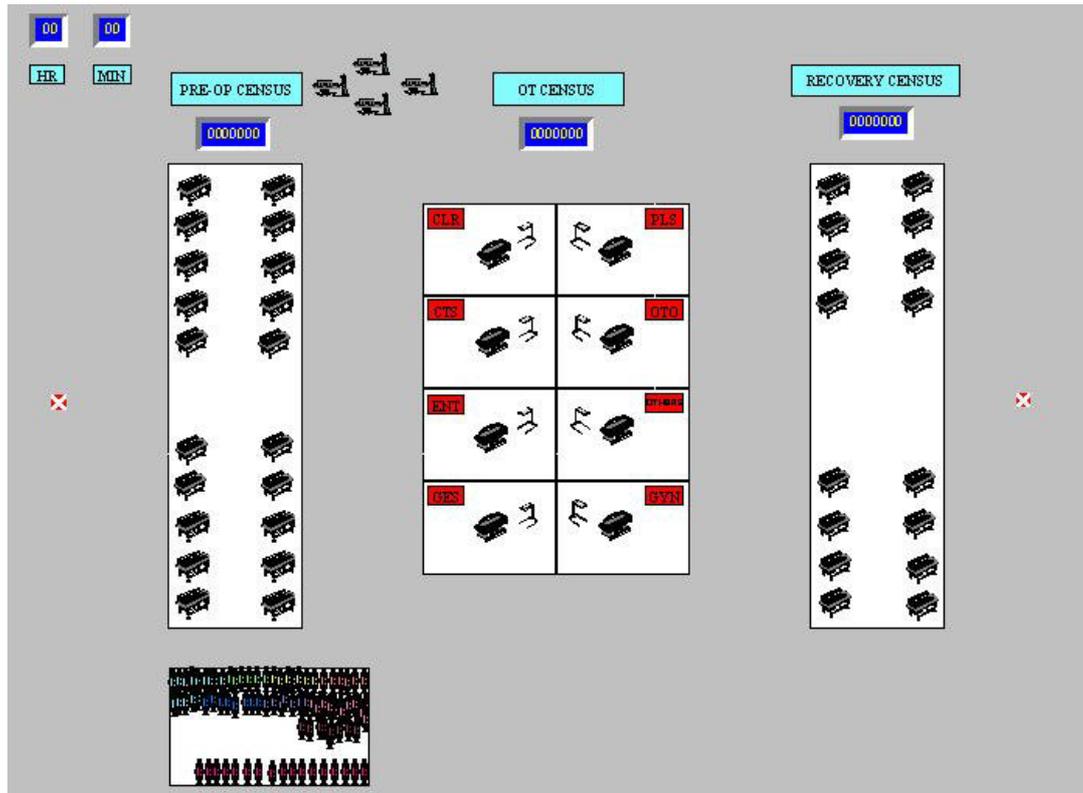


Figure 2: Layout of Completed Simulation Model

Table 1: Patient Types and Distribution

Patient Type	Clinical code	Percentage(%)	Surgery required
1	CLR	12	Colorectal surgery
2	CTS	6	Cardiothoracic surgery
3	ENT	9	Ear, nose and Throat surgery
4	GES	26	General surgery
5	GYN	11	Obstetrics and gynaecology
6	OTHERS	9	Other surgery
7	OTO	22	Orthopaedic surgery
8	PLS	5	Plastic surgery

reserved for a different category of surgery. As such, the number of entities and resources used in this model will be scaled down from the real-life numbers obtained. There are 2 entities in this simulation model, namely *Patient* and *Setup*. In accordance with the 8 categories of surgical cases, the patient is classified into 8 different types using the attribute *aPt_Type* and the user-defined distribution *dPt_Type*. The patient types and distribution are listed in Table 1. It should be noted that the number convention assigned to each type of surgery (such as “1” for CLR, “2” for CTS) is the same throughout the simulation.

Before the entity *Patient* is routed into the operating theatre, the entity *Setup* is first routed into the operating theatre, together with the resource *Anaesthetist*. This is to model the pre-operation procedures required to get the operating theatre ready for surgery on the incoming patient. These pre-operation procedures take 0.5 hours or 30 minutes. As such, there is no need to classify this entity into 8 different types as for the entity *Patient*. The entity *Setup* stays in the operating theatre for 30 minutes with the resource *Anaesthetist* before the entity *Patient* is summoned into the operating theatre to join them.

Locations represent fixed places in the system where entities are routed for processing. This model has 5 locations. Moreover, entrance is the point of entry for the entity *Patient*. The number of entries (or the number of arrivals of this entity) at this location is determined by an arrival cycle. The entity *Patient* is next routed to the pre-operation area (Pre-op), where it waits for 30 minutes before it is called to the next location on the process logic, which can be any of the 8 operating theatres. Should this next location be full, the entity remains at this location until the next location becomes available. The location *Pre-op* is a multi-capacity location; its capacity is 20 patients. The location *Recovery* has a capacity of 16. The model assumes that each patient spends 0.25 hours or 15 minutes in the recovery area.

A resource is a person or piece of equipment used for one or more of the following functions: treating or moving patients, assisting in performing tasks for entities at locations, performing maintenance on or for locations or other resources. In this model, there are 10 groups of resources. Of the ten, eight types represent 8 groups of surgeons from the eight different

surgical specialties, *Surgeon1* to *Surgeon8*. The other two groups are *Anaesthetist* and *Gurney*.

Reengineering the Operating Theatre Complex

It has been noted that the level of utilization for the operating theatres at the complex is rather high. The next step is to improve the efficiency of the system, such that it can achieve greater output with utilization of the same amount of resources.

Currently, the elective operating theatres at the hospital operate eight hours a day, from 08:30 to 17:30. Despite this, surgical operations often end beyond 17:30, due to delays occurring in the individual operating theatre throughout the course of the day. Sometimes it could simply be due to the complexity of the surgery.

In an effort to improve the efficiency of the complex, operating theatre personnel have suggested the possibility of implementing a shift system in place of the current system. By making changes to the variables used, the operating theatre process is reengineered to incorporate the shift system and investigated using the simulation model developed above. In simulating the shift system, 2 changes are made to the original model.

- Arrival Cycle: Instead of patients arriving between 08:00 and 18:00 over a 24-hour period starting at 08:00, patients now arrive between 08:00 and 04:00 over the same period and with the same distribution. This represents 2 shifts with 10 hours to each shift.
- Number of resource units: Since the new 2-shift model utilizes the same amount of resources, as before, the pool of resources has to be shared between the two shifts. This results in less number of surgeons and anaesthetists on duty at any one time. This is incorporated in the new model by halving the number of resource units available.

In implementing a shift system, the system might not have sufficient resources to cope with the increased workload. In an extreme scenario, twice the amount of resources is needed to maintain the level of effectiveness of the system. This is simulated in a third model, by maintaining the number of resource units with the implementation of the shift system.

When two specialties are allocated the use of an operating theatre on the same day, one uses the theatre in the morning and the other in the afternoon. In the extreme scenario, declassifying

all the operating theatres means that no surgical specialty has the exclusive right to any operating theatre. This facilitates the allocation of surgical on a first-come-first-served basis. To model the new system with no classification of operating theatres, the attribute *aPt_Room* and its assignments are removed from the model. Removing it would allow the entity *Patient* to go to any operating theatre location regardless of the patient type.

RESULTS AND DISCUSSION

The simulation model was run for 168 hours (7 days), with a warm-up period of 48 hours, with 20 replications. Table 2 gives a summary of the utilization of the locations. It can be seen that high utilization occurs at OT (OTO), which is the OT reserved for orthopaedic surgery. The pre-operation area is also highly utilized due to the number of patients waiting for orthopaedic surgery. This creates a bottleneck at the pre-operating area, and leads to patient arrival failures. This important issue suggested the possibilities for reengineering.

Figure 3 shows the utilization of resources for the simulation. As the crucial resources in our model are the surgeons and the anaesthetists, it was assumed that gurneys are always available when needed in developing the model. It can be seen that of all the resources available, the group of anaesthetists within the system is the most highly utilized at 18.25%. On top of this, anaesthetists also have teaching and research responsibilities. Thus, the actual utilization hours for the resources used in this simulation model is higher than reflected in Figure 3.

The three suggested models for reengineering were similarly run for 168 hours with 20 replications. The location utilization of the reengineered models is compared with the original model in Table 3. We will refer to the shift system model as Model 1, the shift system with increased staff model as Model 2 and the declassified operating theatres model as Model 3. It should be noted that for Model 3, the operating theatres have been renamed to OT1-OT8. The resource utilization of the reengineered models is compared with the original model in Figure 4. Table 4 summarizes the relevant entity states and efficiency for the 4 models.

Based on the results of simulation, the most efficient model is Model 3, which declassifies the

operating theatres and allows any surgical specialty to conduct surgical operations in any operating theatre. This method reduces the utilization of the pre-operating area from over 90% to 69%, which indicates alleviation of the bottleneck seen previously at this location. The efficiency of this proposed system is found to be 64.8%, an improvement from 45.6% of the current model.

CONCLUSIONS

An in-depth study of the operating complex at the Singapore Hospital has been conducted with the use of simulation software, MedModel. The utilization of the operating theatre complex and its two main resources, the surgeons and anaesthetists, were analyzed in detail. The software modeled the complex operating theatre system accurately and with confidence in results. Due to the comprehensive nature of the simulation software tool, assumptions and shortcuts that have routinely characterized health care and hospital simulations were no longer necessary. The software has allowed modeling the gamut of operating theatre activities quickly and efficiently, from patient admission to disposition.

Several possibilities for process reengineering were proposed to reduce the utilization of the operating theatres within the complex. These possibilities were implemented on the simulation model. The results of the simulation have indicated that operating theatres servicing certain surgical specialties within the operating theatre complex are highly utilized. The surgeons belonging to those specialties are also in high demand. The results also indicate that the anaesthetists serving the complex are highly utilized, possibly due to their anaesthetic responsibilities outside the operation theatre and the pre-operative and post-operative work they conduct for surgical cases.

Thus, in order to maximize the productivity of the operating theatre complex without increasing the workload of the surgeons and anaesthetists, the management needs to look for a way to redesign the operating theatre process. It is also recommended that data collection with regards to operating theatre utilization be reviewed periodically for accuracy and transparency in the data collection process. This is crucial in order to obtain a true representation of the utilization states of the operating theatres, and in turn an accurate productivity index can be derived.

Table 2: Location Utilization

Location	Capacity	Total Entries	Avg minutes per entry	Utilization (%)
Entrance	1	45.65	80.69	33.26
Pre Op	20	63.65	3012.29	92.43
Recovery	16	50.85	31.49	0.99
Exit	1	50.85	0.00	0.00
OT (CLR)	1	5.20	199.75	10.24
OT (CTS)	1	2.75	286.62	8.18
OT (ENT)	1	4.90	286.56	14.26
OT (GES)	1	18.45	379.87	69.09
OT (GYN)	1	6.10	194.34	11.58
OT (Others)	1	6.60	1478.34	88.87
OT (OTO)	1	6.10	1790.40	100.00
OT (PLS)	1	2.25	350.80	8.56

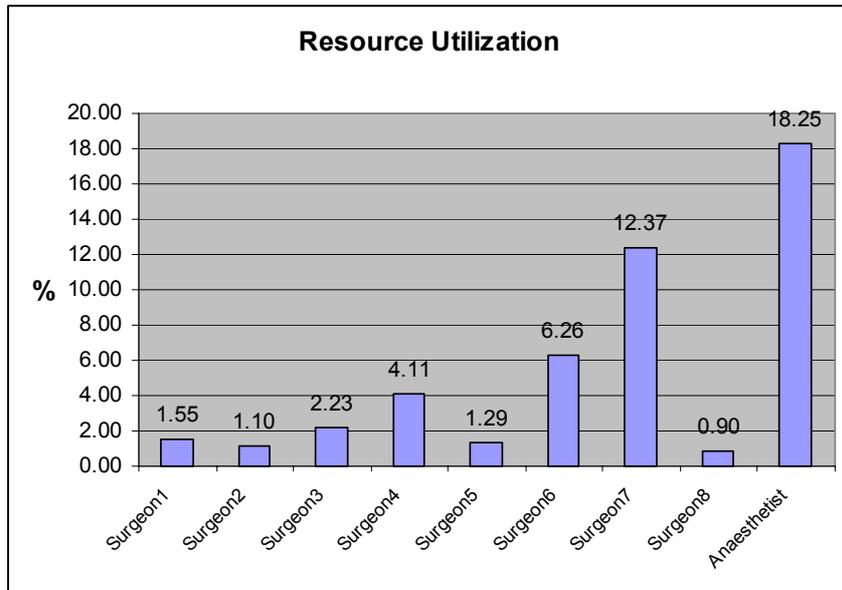


Figure 3: Resource Utilization

Table 3: Location Utilization – Comparing the 4 Models

Location	Utilization (%)			
	Original Model	Model 1	Model 2	Model 3
Entrance	33.26	59.16	55.76	23.00
Pre Op	92.43	96.40	96.06	69.36
Recovery	0.99	1.05	0.97	1.90
Exit	0.00	0.00	0.00	0.00
OT(CLR)/OT1	10.24	9.23	10.29	92.44
OT(CTS)/OT2	8.18	10.34	9.18	91.13
OT(ENT)/OT3	14.26	13.05	15.85	92.49
OT(GES)/OT4	69.09	78.37	69.21	93.94
OT(GYN)/OT5	11.58	10.46	9.42	92.77
OT(Others)/OT6	88.87	91.04	83.37	94.27
OT(OTO)/OT7	100.00	100.00	100.00	94.15
OT(PLS)/OT8	8.56	7.34	7.45	94.12

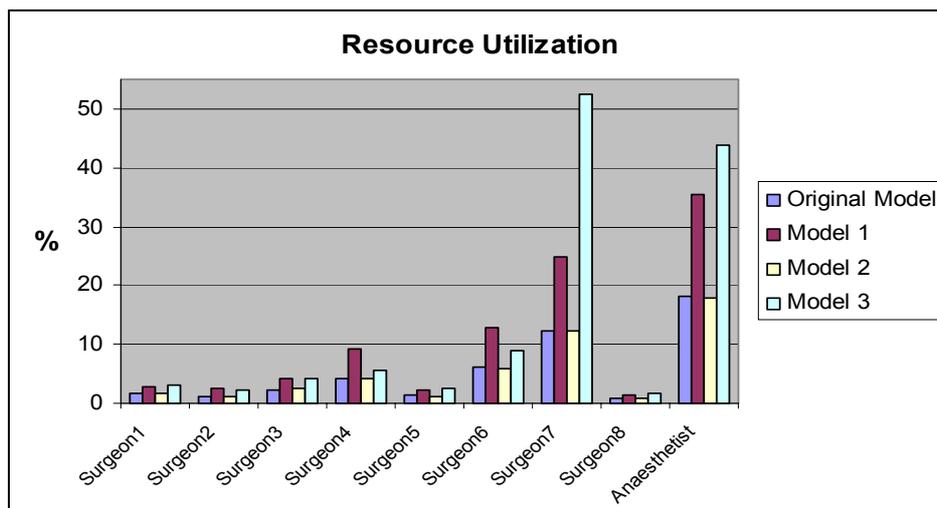


Figure 4: Resource Utilization – Comparing the 4 Models

Table 4: Entity States and Efficiency – Comparing the 4 Models

Entity	Original Model	Model 1	Model 2	Model 3
Average time in system (mins)	1237.80	1207.46	1291.56	1314.72
Average time in blocked state (mins)	874.34	852.37	920.46	864.70
Total number of exits	99.40	100.10	96.55	186.95
Total remaining in system	16.90	17.50	17.30	1.90
Total number of failed arrivals	138.95	141.85	141.90	102.5
Efficiency (%)	45.6	45.3	44.5	64.8

REFERENCES

- Caron, M.; S.L. Jarvenpaa; and D.B. Stoddard. 1994. "Business Reengineering at CIGNA Corporation: Experiences and Lessons Learned from the First Five Years". *Management Information Systems Quarterly*, pp. 233-250.
- Coulson-Thomas, C. 1996. *Business Process Reengineering: Myth and Reality*. Kogan Page: London.
- Davenport, T.H. and J.E. Short. 1990. "The New Industrial Engineering: Information Technology & Business Process Redesign". *Sloan Management Review*, pp. 11-27.
- Earl, M.J.; J.L. Sampler; and J.E. Short. 1995. "Strategies for Business Process Reengineering: Evidence from Field Studies". *Journal of Management Information Systems*. 12(1). pp. 31-56.
- Edwards, C. and G. Walton. *Change and the Academic Library: understanding, managing and coping*.
University of Northumbria at Newcastle.
http://online.northumbria.ac.uk/faculties/art/information_studies/impel/
- Gabel, R.A.; J.C. Kulli; B.S. Lee; D.G. Spratt; and D.S. Ward. 1999. *Operating Room Management*. Boston: Butterworth-Heinemann.
- Hammer, M. and J. Champy. 1993. *Reengineering The Corporation: A Manifesto For Business Revolution*. New York: HarperBusiness.
- Harmon, R.L. 1996. *Reinventing the Business*. The Free Press: New York.
- Harris, H.P. and W.J. Zitzmann, Jr. with 18 contributors. 1998. *Operating Room Management: structure, strategies and economics*. St Louis: Mosby.
- Hout, T.M. and G. Stalk, Jr. 1993. "Time-based Results". *The Boston Consulting Group*.
<http://www.bcg.com>.
- Jacob, R. 1995. "The Struggle to Create and Organization for the 21st Century". *Organization Planning and Design, Inc*.
<http://www.organizationdesign.com>.
- Kelton, D.W.; P.R. Sadowski; and A.D. Sadowski. 1998. *Simulation with Arena*. WCB/McGraw-Hill: New York.
- Kettinger, W.J.; J.T.C. Teng; and S. Guha. 1997. "Business Process Change – A Study of Methodologies, Techniques and Tools". *Management Information Systems Quarterly*. Vol. 21 No. 1.
<http://theweb.badm.sc.edu/bpr/index.htm>
- Merriam-Webster, Inc. 2002. <http://www.m-w.com/>
- ProModel Corporation. 2002. *MedModel Reference Guide and User Handbook*.
- Tan, T.M. and S.B. Chew. 1997. *Affordable Health Care Issues and Prospects*. Prentice Hall.
- Wee, E. 2002. "The female doc quota: What some say". *The New Paper*.
<http://newpaper.asial.com.sg>.
- Weicher, M.; W. Chu; W.C. Lin; V. Le; and D. Yu. 1995. *Business Process Reengineering Analysis and Recommendations*. Baruch College.
<http://www.netlib.com/bpr1.htm>.
- Zhang, C. 2001. *Supply Chain Redesign in Health Care Industry*. Nanyang Technological University: Singapore.

AUTHOR BIOGRAPHIES

ARUN KUMAR is an Associate Professor of System & Engineering Management at Nanyang Technological University (NTU), Singapore. He received his Ph.D. degree in Operations Research from Virginia Tech. Prior to joining NTU, he taught at State University of New Jersey, USA for thirteen years.

A Proposed Standardization of the Navy VV&A Process through the Application of VVML & the VDT

David H. Broyles

SPAWAR Systems Center (SPAWARSYSCEN), Charleston
P.O. Box 190022
North Charleston, SC 29419-9022
(843) 218-4834

david.h.broyles@navy.mil

Jennifer Park

SPAWAR Systems Center (SPAWARSYSCEN), San Diego
53560 Hull St.
San Diego, CA 92152-5001
(619) 553-2848

jennifer.park@navy.mil

Greg Quedenfeld

Senior System Engineer
Eagan, McAllister Associates, Inc.
4854 O'Hear Avenue
P.O. Box 5349
N. Charleston, S.C. 29405
(843) 218-7123

Greg_Quedenfeld@emainc.com

Keywords:

Standards, standardization, VV&A, documentation, technology, innovation, XML, tool

ABSTRACT: *As in the commercial world, modeling and simulation (M&S) is increasingly pervasive throughout the Navy. There is a constant drive to expand the utility of these M&S through distributed M&S, federations, and other combinatorial simulations in the High Level Architecture (HLA), FORCENet, and other such programs. The DOD and commercial sectors' increased reliance on M&S has necessitated a method to provide documented proof of M&S' credibility. Although the Navy has mandated M&S Verification, Validation, and Accreditation (VV&A) to provide this credibility, no single standard exists to execute the documentation process.*

Clearly, this lack of coordination and consistency begs the development of a single, clear, standard method and repository to facilitate the execution of this mandate. The development of Department of the Navy (DON) VV&A Documentation Tool (VDT) and its VV&A Markup Language (VVML) seeks to fill this need by standardizing and streamlining the M&S VV&A documentation process. The web-enabled VDT can not only house all VV&A documentation and its associated information in one repository, but can also standardize and expedite the documentation process through intuitive templates and inherent flexibility via its VVML base. This paper builds the case for a VV&A documentation standard and details the role that VVML and the VDT play in the realization of standardization.

1. Introduction

The ever-increasing use of M&S across all areas of global communication, education, business, and day-to-day functions has come to define the twenty-first century. Seemingly all areas of everyday life has come acquisitions to assessment and management, weapon systems testing in lieu of live firing, and ship and aircraft design. As this M&S presence continues to evolve into reuse, greater federations, and combinatorial simulations, the need for a standardized means to prove the credibility of these M&S becomes more urgent.

to rely on M&S to enrich our world with greater capabilities at lower costs. It is not surprising then that the US Navy has led the way in incorporating this highly malleable technology to build, assess, and meet their technological advancement: from training and education to operations and experimentations,

1.1 Background: Verification, Validation, and Accreditation (VV&A)

The DON has issued a mandate for M&S VV&A (*SECNAVINST 5200.40* [1] and *5200.40A draft* [2]) to address this need. The key focuses of Naval M&S VV&A are two-fold: 1) to ensure the M&S

developers are producing credible models that fulfill their intended uses and 2) to present accurate and correct information to the users so that they can make conscientious and informed decisions regarding their use. *De facto* accreditation has often been used due to the lack of V&V documentation. Because of this lack of a formalized VV&A standard, the credibility of the M&S's pedigree is generally unsubstantiated. However, as M&S is increasingly relied upon to make realistic representations of the "real world" as well as critical decisions with substantial risks, it is clear that "it has worked so far" no longer suffices. With such factors as the safety of human lives and the reduction of cost and time dependent upon the accuracy of M&S, it is clear that VV&A must be accurate and consistent.

The need for standardization of VV&A is thus a formalized set of processes designed to ensure the safety, cost effectiveness, and reliability of M&S. To facilitate the execution of these processes, a VV&A framework and processes have been outlined and in the *Department of the Navy Modeling and Simulation Verification, Validation, and Accreditation Implementation Handbook (2001)* [3] and *The Modeling and Simulation Verification, Validation, and Accreditation Implementation Handbook, Volume I: VV&A Framework* [4]. Thus, the vehicle for creating M&S credibility is set into motion through VV&A.

1.2 Necessity: Standardized VV&A Documentation

Although VV&A creates the framework and processes required to ensure the credibility of M&S, VV&A may be considered an incomplete effort without structured documentation. VV&A documentation must address the concerns of the M&S user by demonstrating its credibility with solid evidence. In other words, VV&A documentation must build on the M&S requirements, design and development, and reporting and documentation of the M&S performances' end results. The efficacy of VV&A hinges upon its thoroughness and consistency.

The simplest way to ensure this pedigree of VV&A is to standardize the documentation process. The Navy Modeling and Simulation Master Plan includes standards as one of the enablers to "promote the development and use of standards, in order to reduce M&S production costs, while enabling consistent and comprehensive representation" [5]. As mentioned above, although the need for standardization, facilitation, and expedition has been identified, no one tool, template, or language exists to facilitate such standardization. Therefore, it is not surprising that the VV&A documentation process remains daunting, inefficient, and generally unpopular. By extension, the potential for legacy M&S reuse under such conditions has been heretofore hindered by this lack of consistent documentation and the arduous amounts of research needed to assess the applicability of an M&S in question for the new user's purpose.

Credible and consistent VV&A documentation has remained elusive. Since hesitation to conduct VV&A continues despite a SECNAVINST mandate and practitioners' guides, something further is still needed to activate and facilitate the VV&A documentation process. In an effort to capture the problem, the Navy has been collecting data on VV&A implementation needs for the last three to four years. The dominant stumbling block has been the facilitation of its execution. VV&A agents have consistently indicated that the most difficult aspect of implementation was the initiation: when and where to start, what to consider, and how to bring about structured management to produce precise, complete, and useful documentation that users can use to make informed decisions. The fundamental problem is therefore clear: implementation guidance alone cannot meet the needs of consistency and reuse. A concrete, tangible standard for reporting information that is easy to use is needed.

1.3 The Utility of VVML and the VDT in VV&A

The clamor to simplify the documentation process with an intuitive format is the foundation for the creation of VVML. The evolution of the Navy's effort to support VV&A documentation has brought several key, M&S-wide desiderata to light: the possibility of VV&A standardization; the need to promote M&S reuse, and the need for easy, on-line VV&A collaboration. Richly structured VV&A processes are possible by utilizing the Extensible Markup Language (XML) to create universal access to M&S data. The VVML concept directly addresses these VV&A needs by leveraging the XML-based framework into a VV&A-specific language, thereby providing a bridge between M&S requirements and open/commercial web-enabled standards. Further, the execution of VVML in the VDT, with its intuitive templates and user interface, solves both the initial documentation and reuse standardization problems. The associated benefits of such a marriage are numerous: adaptability to changes in VV&A structure and data, portability to enable universal, web-enabled access to VV&A information; the capacity to link to the Navy Modeling and Simulation Resource Repository (NMSRR) [6]; and easy maintenance and reference.

2. Purpose:

To date, no standards specify what information, documentation, or format is required for M&S VV&A. However, observation of the common needs of numerous M&S VV&A programs reveals that an efficacious and accommodating VV&A documentation process hinges on two key factors: a common documentation standard and a method by which to achieve this documentation. This paper will show that the VVML provides the extensible language

necessary to generate a common VV&A template and that VDT provides a standard content and format for all VV&A implementation. Continued discussions, collaboration, and incorporation of M&S VV&A needs will only provide ever greater utility of the VVML and VDT.

Thus, this paper will argue the need and justification for a single M&S VV&A documentation standard; provide a walkthrough of the VVML concept and its application through the VDT; and delineate the evidence needed to place VVML and the VDT as considerations for M&S VV&A standards. By extension, the need for VVML and the VDT in the context of the Navy's emphasis on M&S reuse and future endeavors will be highlighted. VVML and the VDT will prove to be inextricably vital to the creation of a formal and consistent means to evaluate M&S and the creation of federative, distributive, Defense Information Infrastructure Common Operation Environmental (DII COE), and composable M&S.

This argument will be laid out in three parts: the presentation of the VVML concept, its utility as facilitated by the VDT, and the justification for its consideration as the foundation for future standardization. Finally, the discussion will conclude with future VVML/VDT capabilities and areas for consideration by the VV&A community in expanding VVML/VDT utility.

3. Revolution: The Utility of VVML and the VDT

VVML and its application, the VDT, provide the standardization and flexibility needed to create consistency in VV&A documentation by leveraging XML and providing users with a physical and tangible object to guide them through the VV&A documentation process. VVML creates consistency with Navy mandate by paralleling its contextual structure to *The Department of the Navy Modeling and Simulation Verification, Validation, and Accreditation Implementation Handbook, Volume I: VV&A Framework* [7] while the VDT creates VV&A metadata files to record the VV&A process and progress. For each M&S project a single metadata file is created that contains four related reports: the Accreditation Plan and Report, V&V Plan and V&V Report. The resulting VVML file establishes a searchable index that facilitates configuration management, traceability and a host of other benefits that will be addressed in the body of this paper. The flexibility of this web-based language and tool will also save the Navy valuable time and money by uniformly delivering M&S projects for accreditation and creating M&S repositories with searchable entities for future reference.

3.1 VDT: the VV&A Implementers' Interface with VVML

VV&A implementers require a quick and efficacious means to understand and perform the VV&A process. The VDT fulfills this need by formatting the required reports directly from the VV&A template and creating a manageable VV&A task list. Correlation with the VV&A handbook [8] and SECNAVINST 5200.40A [9] provide additional clarification of those sections that new VV&A implementers may not understand and facilitates the easy creation of required documents and reports. Once the VV&A agent familiarizes him/herself with the VDT template fields and creates a VV&A task list, the VVML supports the execution of these tasks by connecting all members involved in the M&S development and VV&A effort through a single collaborative documentation standard. The VDT captures the textual documentation within specific contexts defined by the VVML. The collection of VVML files is then stored in a single repository, such as the NMSRR, creating a valuable asset to the DON M&S communities.

This standardization of process inherently makes systems development more formalized and rigorous. VVML adheres to the standard the VV&A template for documentation by representing the required M&S VV&A information while the VDT uses VVML to surround the VV&A content. With the VDT portal, VV&A metadata can be captured for multiple projects and will be able to be linked with the Navy's NMSRR. Thus, VVML promotes project visibility and facilitates information exchange. This specialized portal and/or the NMSRR will provide a robust web interface for searching M&S projects using specific VV&A criteria.

4. Evolution: The Creation and Development of VVML and the VDT

The VDT's first iteration was as a stand-alone Microsoft Access database and has evolved into the DON VV&A Documentation Tool, Versions 1 and 2. Developed by SPAWAR Systems Center, Charleston, multiple users tested Version 1; their comments were incorporated in the final Version 1. With the release of Version 1, numerous other comments called for collaboration and information-sharing utilities. The success of VV&A hinges upon participation from various members within the program and organizational resources, both internal and external. Hence, responses to the VDT v.1's utility and capabilities were addressed to several users including: OPNAV, Marine Corps, Navy and Army threat systems offices and other organizations.

Strong consideration was also given to interoperability with the NMSRR. As the NMSRR was created to promote M&S reuse by providing a centralized collection point where a registered user can enter

information about their M&S, the linking and portability of XML can allow certain fields of the VDT to be directly linked to NMSRR and populating the database with current information with the consent of the users. The automation of this VDT/NMSRR posting will relieve M&S VV&A agents from having to manually duplicate the entry of this vital information. Running historical information that other users can access in the NMSRR facilitates the evaluation of existing M&S for reuse based on the VV&A documentation.

4.1 VVML: Metadata

Metadata can be stored as data; as well as a resource indicated as a URL, i.e. <http://www.navmsmo.navy.mil/>; one resource may therefore contain information about itself and/or

another resource. One goal of writing structured Metadata is to include as much of the syntax and semantics such that maximum information can be acquired when referencing a Metadata document. In the context of the NMSRR, structured content in the metadata is paramount to populating the database with useful information.

To make this XML data interchange a reality, the establishment of a metadata standard is necessary so that its content conforms to the Navy's VV&A policy and complies with the World Wide Web Consortium (W3C) XML Schema [10]. As the primary goal of the VVML is inherently to provide a framework for tracking V&V activity, the source code in Figure 1 is provided to show the attribute tags used to track events:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by
David Broyles (Home) -->
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <xs:element name="ACC_ACTIVITIES">
+ <xs:complexType>
- <xs:keyref name="ACC_ACTIVITIES_VV_AND_A"
refer="VV_AND_A_MAIN_VV_AND_A_ID">
<xs:selector xpath="." />
<xs:field xpath="VV_AND_A" />
</xs:keyref>
</xs:element>
- <xs:element name="ACC_EVENTS">
- <xs:complexType>
- <xs:sequence>
<xs:element name="ACC_EVENTID" type="xs:integer" />
<xs:element name="VV_AND_A" type="xs:integer" />
- <xs:element name="COMPLETION_DATE">
- <xs:simpleType>
- <xs:restriction base="xs:string">
<xs:maxLength value="10" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="RESPONSIBLE_PARTY"
type="xs:integer" />
<xs:element name="EVENT_TYPE" type="xs:integer" />
</xs:sequence>
</xs:complexType>
```

Figure 1- XML Code segment

4.2 VVML: Hyperlinks and XML Sub-languages

Further supporting the use of XML and web-based applications as the basis for VV&A documentation standardization are hyperlinks, which provide a simple, yet powerful, means of linking documents. As such, VVML applies the the XML Linking Language (XLink), XInclude and XPointer languages to facilitate the VVML's role of Metadata Collector and

future capabilities discussed later in this document. Because XLink allows more abstract linking than standard HTML, users can insert and move elements into XML documents with greater flexibility, the creation and description of links between resources becomes more robust. "XLink provides a framework for creating both basic unidirectional links and more complex linking structures[,] allow[ing] XML documents to assert linking relationships among more

than two sources, associate Metadata with a link, and express links that reside in a location separate from the linked resources [11].” By using the XML syntax to create structures that can describe both simple, unidirectional hyperlinks (similar to HTML) and more sophisticated links, XLink’s value to VVML and the VV&A practitioner is that it can include a wide array of documents. Audio, video, database data, training media, schedules, and even XML-enabled test results can thus be added to any VV&A file. This will be particularly useful in the realm of regression testing by linking statistical applications directly to the VDT.

While XLink is an XML-based language that specifies constructs for advancing linking in XML documents, XPointer is a non-XML language that allows the user to address the internal structures of XML documents by specifying link sources and targets [12]. The XPointer language can be used as the basis as a fragment identifier for any Uniform Resource Identifier (URI) reference. By definition, a URI is a series of characters used to differentiate names. URIs locate a resource whose Internet media type is one of text/xml, application/xml, text/xml-external-parsed-entity, or application/xml-external-parsed-entity [13]. The XPointer’s capacity to harness URI fragments allows the VDT to manage validation test parameters. In contrast, XLink allows the management of parameters such as parameters as schedules and costs is shown in Figure 03 below as a requirements matrix that links pre-test predictions to test results. Using the XPointer specification, the VDT can reference test results pointing directly to another application by simply pointing to the URI of the application.

Finally, the VVML incorporates the XInclude specification to provide the mechanism for organizing and capturing non-text data such as graphics, embedded objects or files. XInclude provides a link with the attribute value of embedded resources in documents generated by the VDT. Such links provide syntax independent of media type to indicate that a resource is to be embedded graphically within the display of the document [14]. This capability has far-reaching implications. For example, it can add the ability to link live schedule information or even model information directly from the M&S applications to the VV&A project file.

5. UTILITY: The Benefits & Current Limitations of the VDT/VVML

The benefits of VVML and the VDT are numerous: from one-step standardization and aggregation to universal applicability. Further utility and expansion into the area of M&S scheduling and management covers only one feature that the VDT may hope to achieve in the future. However, it is also important to note the current limitations of VVML and the VDT, not only its benefits. As such, this section seeks to

provide the road ahead as well as the current status of VVML and the VDT’s utility.

5.1 One-Step Standardization and Aggregation

The main benefit of using the VDT over Microsoft Word® is that only the VDT facilitates standardized preparation of VV&A documentation *and* supports the aggregation of V&V information in an M&S repositories, such as the NMSRR. The result is similar to using tax-preparation software to fill in your tax forms and file your income tax return electronically. Both the preparer and the government easily achieve greater levels of consistency and efficiency by using this standardized method.

The overall VDT process produces standardized representations that can be scaled to accommodate both small to large M&S projects. Specifically, VVML is structured to provide the four report templates articulated in the *M&S VV&A Implementation Handbook Volume I* [15] in electronic form:

- Accreditation Plan
- Verification And Validation Plan
- Accreditation Report
- Verification And Validation Report

These four reports contain documentation of the specific VV&A plan details, correlating to the VV&A process. These processes include:

- Data Verification
- Data Validation
- Design Verification
- Implementation Verification
- Conceptual Model Validation
- Results Validation
- V&V Reporting

Thus the VDT and its VVML underlay form the only tool that not only creates standardized documentation, robust flexibility, and XML-based utility, but also ensures compliance with Navy M&S mandates.

5.2 EASE OF USE & UNIVERSAL APPLICATION

As noted above, the Web-based VDT application utilizes XML to enable its underlying database to create an easily accessible user interface that supports mandated Navy VV&A standards. The VDT is accessible from any browser and connects to a specialized Navy website that serves the latest VDT version. The web interface supports multiple platforms to ease and assist the user to produce standardized Navy VV&A documentation. In order to consistently fulfill this tall order; verify that all VV&A standards are met; and ensure that any user, regardless of familiarization with the VV&A process, can easily accomplish all tasks, the VDT represents a tree of documentation headings with user directions and an edit pane that accepts formatted text and pictures, as depicted in Figure 02:

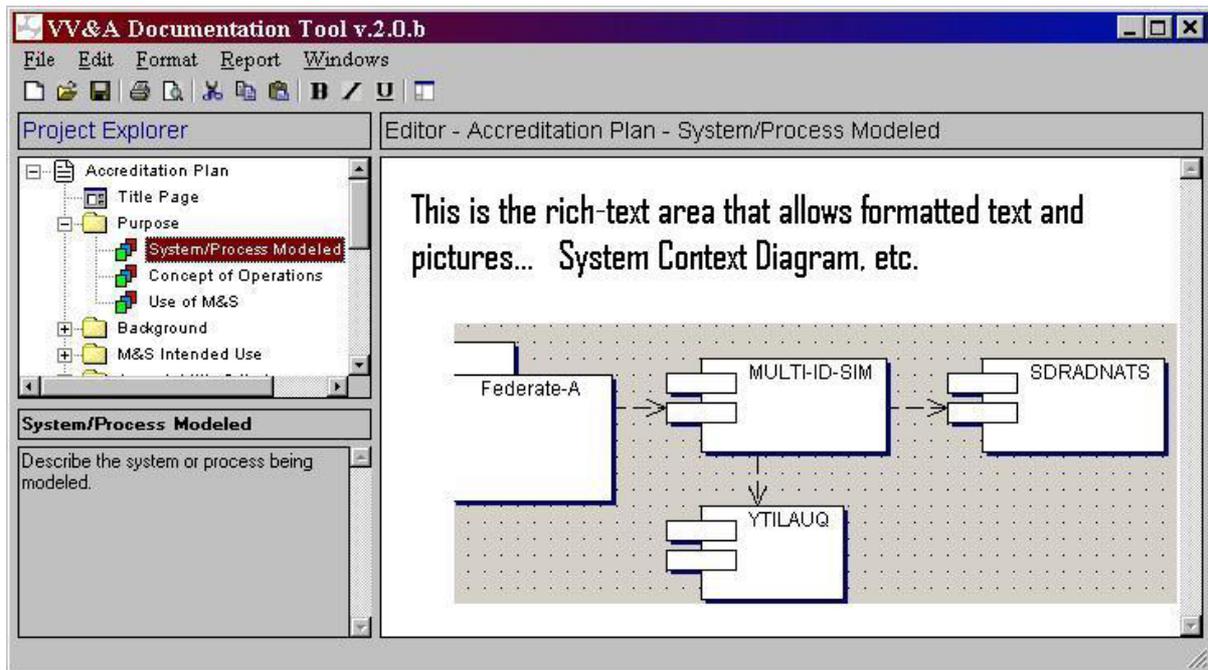


Figure 2- The VDT main window displays the content associated with the “System/Process Modeled” subject of the V&V Accreditation Plan.

Additionally, the current iteration of the VDT interface is sufficient to represent all M&S acquired, developed, managed, and used by DON activities. The fact that all categories of M&S used by DON activities (including live, virtual, and constructive simulations; distributed simulations; federates and federations; emulators; prototypes, simulators; and stimulators) can harness this tool with relative ease with little additional effort is in itself a revolution in M&S VV&A documentation. In the increasingly demanding world of Navy M&S management, the importance of the VDT for project developers, system engineers, system architects, testers, and management attempting to balance their traditional responsibilities, challenging schedules, and tight budgets with the new demands of M&S VV&A can hardly be understated.

As discussed earlier, the VDT creates each report separately by user request to the format prescribed by the Navy standards for VV&A documentation. For easier access and standard formatting, project VV&A content is sequentially placed between indexed documentation headers while maintaining the user-supplied formatting intact. Each report is then generated and can be previewed using the VDT using the universally recognizable “Print Preview” menu selection. The “Print” menu selection generates and sends a standardized report to the specified printer. The following is an example of the Print Preview window:

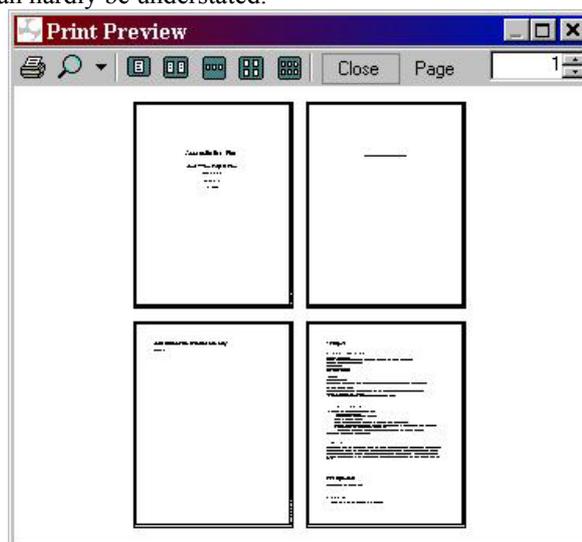


Figure 3- Standard Print Preview window

5.3 Current VDT Limitations

The VDT directly manages the VVML structure and document content to the new VV&A standard. Because the VDT can include “cut-and-paste” data from other editors and tools, the VDT inspires thoughts of composability and substantive interoperability between M&S components. The VDT can represent and relates a user’s high-level perspectives of the usefulness of M&S. However, further development is needed for the VDT to provide users with these advanced features and capabilities. Some known issues include:

- Dependent on Web connectivity
- The addition of a spell check feature
- The capacity to add a table of contents, page numbers, and indexes
- Binary representation of images that occupy less disk space than XML-wrapped rich text

5.4 The Tailoring of V&V

The process of VV&A can be exhaustive and costly, so it is suggested that this judgment process be conducted using common sense, sound business management practices, and considerate of time and resources efficiencies. We believe that organizations can adhere and comply with regulations to do VV&A, follow the guidelines, and produce meaningful VV&A documentation efficiently with the use of the VVML Documentation Tool. Documentation content is the sole responsibility of VV&A practitioners. As such, portions of the VV&A content may be left blank, resulting in abridged VV&A reports.

6. Further Fruition: Future Capabilities

6.1 Flexibility and the Additional Functionality

In 2004 the release of a new standard for Navy VV&A documentation clarifies, characterizes, and evolves the VV&A process. In synergy with this evolution, VVML emerges as the single standard to represent project VV&A artifacts. Because VVML is a language that can easily adapt to future requirements and evolving VV&A methods, software development, and VV&A process maturity, higher levels of validation categorization and new means of requirements traceability can be established and standardized through the VDT. VVML also has the capacity to adapt to additional XML tags. VVML extensions open new possibilities to represent complex relationships between M&S components. Often expensive, proprietary tools appear to fill the current void. However, continued improvements to the VDT will extend and improve the management of the VV&A process, including:

- Monitors and metrics of VV&A activities and progress

- The provision of essential feedback regarding the organization of VV&A activities with M&S development lifecycle.
- The collection and summarization of lessons learned to extrapolate possible changes and improvements to future VV&A processes and standards
- The ability to expand data import and export methods including requirements traceability matrixes (RTM) and other tool matrixes.

6.2 VDT Support of DON Web-Centricity and Streamlined Schedules

With regard to the DON’s emerging web-centric approach, the VDT application has the possibility of improving the management of VV&A processes by providing timely visibility of progress, audits, summary analysis, and judgments of credibility. By using the VDT, M&S managers can coordinate their efforts to respond to critical issues of performance and effectiveness, thereby streamlining the effort to quantify and mitigate risks. Future versions of the VDT will manage a template that extends for each of the project M&S components. After the VV&A process is tailored to fit the project, a schedule can be created that associates VV&A activities and time with costs. Thus, the VDT has the potential to quantify and track costs. This functionality will help evaluate project risk as a relationship of historical costs, budgeted costs, and projected costs.

As the VV&A Documentation Tool suggests a roadmap for the direction and completeness of the documentation based on the stage of the M&S development cycle, the VDT can evolve to promote each specific V&V process M&S development stage. The V&V calendar can be used as a simplified resource planner to organize high-level V&V tasks. As a Web-accessible tool, the VDT can support a standardized process to efficiently capture and share V&V data for streamlined documentation and collaboration while V&V activities are audited throughout the execution. Hence, the VDT emerges from a documentation template program to a full service VV&A assistant that supports an organization’s advance into process maturity.

The VDT organizes VV&A information that identifies how the M&S project can sufficiently execute VV&A and meet acceptability criteria. When actual results of V&V testing are available, M&S management gains a cohesive understanding of the M&S progress by tapping into this rich resource, thus casting away the vagueness and opacity so often noted as a frustration of VV&A. With such openness and clarity, the acceleration of the overall M&S schedule is likely as developers will be able to identify and focus on higher-priority tasks which directly relate to the most important M&S acceptability criteria. Because VVML has the capability to conjoin system-level tests with

expected results, the applications of VVML in the VDT can be expanded ad infinitum to create and maintain a matrix of relationships that represent the acceptability criteria and the status of system-level tests and analysis.

6.3 Composability

The capacity for any one standardized template to be deployed without knowledge of the M&S developer's methodology (object-oriented, functional decomposition, or fast-prototyping), has been noted during the VDT development effort. By encapsulating the details VVML uses to represent a structure of data, models, and simulation component documentation in the extensive template framework, documentation interoperability can be substantively evolved via VVML generation and links. VVML supports true composability among M&S iteratively feeding the VDT new tools. This iterative process also achieves technical interoperability.

A concerted effort by the SISO Extensible Modeling and Simulation Framework (XMSF) Group [16] promotes the concept of composable components using the). In order to arrange and compose new M&S projects and configurations, M&S will require consistent and standardized nomenclature, high-level descriptions, and the designation of each M&S's specific solution. It is important to note that although VVML is designed to support associations in the Navy's standard of the VV&A documentation, certain sections of VV&A documentation describe unique qualities: system components and their associated requirements, acceptability criteria, assumptions, qualifications, constraints, and test results. Those who would compose an M&S need a strong understanding of the various implementation techniques and their purposes. In such circumstances, there may be no better representation of the documented, analyzed, and approved M&S, than the project's VVML file.

6.4 M&S Reuse and Cost Savings

The consistent use of the VDT and VVML directly aids in effort to reuse M&S by standardizing original VV&A documentation for ready accessible for reference, reducing the cost of the overall costs of legacy VV&A. VV&A documentation provides a concrete, value-added deliverable for that benefits the entire M&S community. Developers who evaluate M&S for reuse will not only have easily accessible and consistent VV&A documentation, but also access to the original validation test results and expected outcomes in one file, thus streamlining the future efforts to understanding of the legacy's M&S performance and capabilities.

7. Summary

The Navy Simulation Baseline Assessment [17] for FY02 reported a DON-wide expenditure on M&S of over \$380M in the areas of training and education, operations and experimentations, acquisitions, assessment, and management, science, and technology. With so much riding on the line, the standardization of VV&A is no longer a nicety; it is a necessity. VVML and its implementation via the VDT specifically address this exigency by developing a single Navy VV&A documentation standard.

Together, VVML and the VDT streamline VV&A by automatically generating standardized documentation and saving the information as constant VVML files. VVML provides a specialized XML-based language to create a unique Navy VV&A documentation template. The VDT dynamically applies VVML to aggregate the VV&A content. It is the VDT's use of VVML that solidifies a solution that aggregates the VV&A documentation process and provides a more meaningful representation of VV&A information. As the foundation of VVML is the highly malleable XML, future growth in addressing the growing demands of M&S VV&A, its management, and processes and flexibility is virtually limitless. VVML, *the* foundation of the VDT, may lead to advancements in how VV&A results are composed, organized, distributed, searched, shared, and reused.

8. References

- [1] Secretary of the Navy: "SECNAVINST 5200.40 Verification, Validation, & Accreditation (VV&A) of Models and Simulations," 19 April 1999.
- [2] Secretary of the Navy: SECNAVINST 5200.40A: Department of the Navy Verification Validation and Accreditation (VV&A) of Models and Simulations" draft.
- [3] Office of the Chief of Naval Operations Navy Modeling and Simulation Management Office: "Department of the Navy Modeling and Simulation Verification Validation and Accreditation Implementation Handbook", February 2001 (hereafter notated DON VV&A Handbook 2001).
- [4] Office of the Chief of Naval Operations Navy Modeling and Simulation Management Office: "Department of the Navy Modeling and Simulation Verification Validation and Accreditation Implementation Handbook Volume I, VV&A Framework" Draft (hereafter notated DON VV&A Handbook Vol. I Draft).
- [5] Office of the Chief of Naval Operations Navy Modeling and Simulation Management Office: "Navy Simulation Master Plan" Pre-release Draft, 21 October 2003.
- [6] <http://navmsmo.hq.navy.mil/index.cfm>, M&S Resources
- [7] DON VV&A Handbook Vol. I Draft
- [8] Ibid and DON VV&A Handbook 2001

- [9] SECNAVINST 5200.40A
- [10] <http://www.w3.org/XML/Schema>
- [11] Michael Morrison: "Part V: XML and Web" SAMS Teach Yourself XML in 24 Hours, Sams Publishing, Indianapolis, IN 2002.
- [12] Michael Morrison: "Part V: XML and Web" SAMS Teach Yourself XML in 24 Hours, Sams Publishing, Indianapolis, IN 2002.
- [13] <http://www.w3.org/XML/Schema>
- [14] Ibid
- [15] DON VV&A Handbook Vol. I Draft
- [16] <http://www.sisostds.org/>
- [17] Office of the Chief of Naval Operations Navy Modeling and Simulation Management Office: "Navy Simulation Baseline Assessment," 30 April 2003

9. Acronyms

DII COE	Defense Information Infrastructure Common Operating Environment
DON	Department of the Navy
HLA	High Level Architecture
HTML	Hyper Text Markup Language
M&S	Modeling and Simulation
NAVMSMO	Navy Modeling and Simulation Management Office
NMSRR	Navy Modeling and Simulation Resource Repository
OPNAV	Office of the Chief of Naval Operations
SECNAVINST	Secretary of the Navy Instruction
T&E	Test and Evaluation
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VDT	DON Verification, Validation, and Accreditation Documentation Tool
VV&A	Verification, Validation, and Accreditation
VVML	Verification, Validation, and Accreditation Markup Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XMSF	Extensible Modeling and Simulation Framework

10. Biographies

DAVID H. BROYLES is a scientist with Space and Naval Warfare Systems Center, Charleston, SC and is program manager for the VV&A Documentation Tool. He has 10 years experience in software development in the area of Geographic Information Systems, document management systems, and accounting systems. He has supported NAVMSMO Navy Modeling and Simulation Office for the past two years.

JENNIFER PARK is the Navy Modeling and Simulation Management Office (NAVMSMO) Verification, Validation and Accreditation (VV&A) Lead who is responsible for the Department of Navy policy and implementation guidance. She is currently with SPAWAR Systems Center San Diego and graduate of University of Missouri and Naval Postgraduate School with BSEE and MSEE. She has served in the United States Navy and her primary focus is in the CRISR.

GREG QUEDEFELD is a Senior System Engineer at Egan, McAllister Associates with over 20 years experience in software development. His project work included Navy electronic warfare verification systems, Army and simulation projects, as well as commercial web-based information systems built with XML, Java™, and J2EE. He has constructed software by applying tools that facilitated UML, C/C++/Java, XML, and database schemas. He has experience with the software development lifecycle and related technical documentation, under DoD and IEEE standards.

10.1 Acknowledgements

The authors would like to acknowledge Hanae K. Hara for her contributions in amalgamating the concepts of this paper into this finish product.

SIM-SERV CASE STUDY: SIMULATION-BASED PRODUCTION SCHEDULING AND CAPACITY OPTIMISATION

Galina Merkuryeva
Department of Modelling and Simulation
Riga Technical University
1 Kalku Street, LV-1658 Riga, Latvia
E-Mail: gm@itl.rtu.lv

Nigel Shires
Preactor International Ltd
Combrash Park, Bumper Way
Chippenham, Wilhire, SN14 6RA, UK
E-mail: nigel@preactor.com

KEYWORDS

Simulation, Business Process Optimisation, Simulation-based Scheduling

ABSTRACT

The simulation tools developed to support customised solving of a production scheduling and capacity optimisation problem for medium-sized UK based company are presented. The improvements of the simulation-based scheduling approach and its benefits in practice are given. Decorpart case study presented in the paper was developed within Sim-Serv Thematic Network project 'Virtual Institute on Production-Oriented Simulation' under the EU-funded GROWTH research programme.

INTRODUCTION

Modern production scheduling tools are very powerful and offer a vast range of options and parameters for adapting the tool's behaviour to the requirements of the real process. However, the more options exist, the more difficult it becomes in practice to find the best configuration of the tool. Even experts often cannot predict the effects of many possibilities. Testing out even a small number of possible configurations in reality, and studying their effects on the real production process might take months and might severely reduce the overall performance. Hence such tests are not feasible in practice. It is much faster, easier, safer and cheaper to test and optimise a production scheduler using a simulation model than using the real process.

In order to make the best use of an advanced and sophisticated scheduling tool in the piece-part SME manufacturing and to find an optimal configuration of its rules and parameters, modular simulation models of the entire business/production process and production anodising stage were built to test out the effects of various scheduler configurations. Testing and optimisation of the scheduling tool configuration was carried out off-line using the models. The real production process was not disturbed, and the optimal configuration was found very quickly at low cost.

Simulation-based production scheduling and capacity optimisation case study presented in the paper was developed within Sim-Serv Thematic Network project

'Virtual Institute on Production-Oriented Simulation' (www.sim-serv.com).

SIM-SERV: VIRTUAL CENTRE OF SIMULATION SERVICES

Simulation technology may be applied over a vast range of industrial, commercial, infra-structural and general service areas. The above-mentioned Thematic Network project 'Virtual Institute on Production-Oriented Simulation' itself focuses (Krauth 2002) on product- and product-oriented simulation. Among the significant areas of manufacturing the project sector range includes the following areas: Machine Tools, Transports, Power Plants and Mechanical Engineering. The main applications are related to Advanced Control of Manufacturing, Product Simulation, Waste Minimisation, Business Process Models, Environmental Protection, Accident Analysis, Process Redesign and Engineering and Logistics.

It is widely accepted that simulation in spite of obvious power and benefits is not widely used in industry as it could be. The estimated potential for saving and improvement in European Industry, which could be achieved by proper use of simulation, is enormous. The main objective of Sim-Serv is to turn potential into real benefits. The central service of Sim-Serv on its web site presents a database of technical and scientific information about simulation, relevant case studies and success stories, a list of supplies and links to them, different services to support both potential users and suppliers as customised solution and technology providers. Industrial users could also find independent consultation and advice about possibilities of applying simulation technology to their business, the expected costs and benefits. An in-depth analysis of the problem leads to a recommendations of suitable solutions and suppliers who are able to deliver them.

Moreover, the supply side consists of numerous small companies that offer in many cases highly specialised tools and solutions. They tend to have problems getting Europe-wide visibility and finding customers from a broad range of industry sections. Through dissemination and marketing activities, Sim-Serv facilitates access to European market even for small and medium-sized providers. As successful could be mentioned cooperation of two technology suppliers - Department

of Modelling and Simulation of the Riga Technical University from Latvia and Preactor International from UK, that in a very short period of less than three months provided customised problem solutions to UK-based Decorpart company that are presented in the paper.

DECORPART CASE STUDY

Decorpart that is a medium-sized company produces a wide range of different small, pressed aluminium parts in large quantities to a range of other consumer-focused businesses. Typical applications include spray assemblies for perfumes and dispenser units for asthma sufferers. The business lies in a highly competitive sector and its success depends on achieving high efficiency and low cost of manufacturing in all production steps. Production scheduling becomes therefore very critical. In the past, this company had already installed software tools supporting the scheduling of individual areas of the production process.

To improve the overall company performance, increase its output and reduce the product lead-time, an automatic supply chain server, - an overall scheduling system coordinating all local systems, was planned to be introduced. In order to deliver the best possible solution to the customer, the supplier of the scheduling tool, Preactor International decided to use simulation for finding the optimal configuration of the scheduling tool.

The problem is to build simulation tools, which will embrace the arrival of orders and sequencing of production to meet these demands. An important aspect is to model the production process itself in order to ensure that its main stages are optimally loaded at all times. The important anodising stage has to be modelled in most detail. The overall impact of simulation is expected to be higher plant throughput with lower unit costs.

The following key objectives are stated in the problem: to model interrelated business and production processes at the company, to determine the overall lead time of orders, to test the sensitivity of the overall lead time of the production process to optimisation, particularly at the anodising production stage.

This simulation tool to be introduced is aimed to use for testing configuration of the finite-capacity scheduling and advanced planning Preactor software tool and for iterative optimising its performance off-line prior to its implementation and integration at the customer's site. The envisaged scheme is aimed to complement and link together, localised advisory systems previously installed on individual areas of the production process.

IMPROVEMENT APPROACH

A custom-built business/manufacturing model was built that simulates the arrival of orders; it shows the queuing of the orders for processing. The individual machines in

each process stage were modelled as a group of machines with an overall capacity per day or per week. The model was built in a modular style so that each production stage could be further modelled to a greater level of detail. The anodising stage of the production process was modelled in a greater level of detail following successful validation of the initial model.

The batch anodising process stage sub-model was refined to model the individual anodising tanks, so that colour changeover and set-up times have been modelled. Queue ranking rules were developed to minimise the colour changes to test whether the overall lead-time of orders is sensitive to optimisation of the anodising process stage. The Production Simulation System Promodel that allows easily to built-in Excel files was used as a basic tool for simulation software development.

The simulation scope then required an amalgamation of the Preactor scheduling tool with: (1) a high-level business/ manufacturing system model, and (2) production process anodising stage sub-model detailed representation.

These two models that developed by the simulation supplier, i.e. Riga Technical University, were used for testing an initial configuration of the Preactor scheduler and iterative optimisation of its parameters and rules off-line prior to its implementation at the customer's site.

BUSINESS/MANUFACTURING SIMULATION MODEL

The high-level business/manufacturing model is aimed to model interrelated business and production processes at the company, to analyse and optimise business processes at the planning department dealing with processing of incoming enquiries and planning production orders already confirmed by customers. Comparison of two alternatives planning scenarios using the simulation model was done to check the benefits of introducing at the company an automatic advanced production planning and capacity optimisation tool with a maximum response time of 0.1 hour per inquiry.

This custom-built entire business/manufacturing model (see, Figure 1) was developed that: (1) simulates the arrivals of enquiries and their processing time, (2) generates orders becoming confirmed by customers and orders planning time, and (3) shows the queuing of the production orders for processing. There are two types of incoming enquiries – PH_Enquiries or Pharmaceutical Enquiries, and PC_Enquiries or Personal Care Enquiries. Production itself consists of the following processing stages: Pressing, Degreasing, Jigging, Anodising, and Packing. In each stage the individual machines were modelled as a group with an overall capacity per day or per week.

Based on analysis of historical data and taking accounts their stochastic nature, the following probability distributions (Table 1) were derived in order to generate in the model the time between arrivals of Enquiries, processing times of the Enquiries, average response time from the customer and actual planning time of Confirmed Orders. About 33 % of all incoming enquiries are PH_Enquiries. Probability of Enquiries becoming an Order decreases as function of planning response time including queuing time and is given in Table 2. The value of confirmed orders received by the company increases as a function of the planning response time and average value per enquiry is defined.

Table 1: Probability Distributions (all values are given in minutes)

Data	Distribution Type	Distribution
Time between arrivals of Enquiries		
PH_Enquiries	Exponential	E(60)
PC_Enquiries	Exponential	E(20)
Processing time of Enquiries	Uniform	U[35,5] U[4,6]
Response time from a customer	Constant	24 * 60
Actual planning time of Confirmed Orders	Uniform	U[55,5]

Table 2: Probability of Enquiries Becoming an Order

(Enquiry becoming Confirmed)	Planning Response Time
50%	<1 Hour
20%	1 – 8 Hours
10%	24 – 48 Hours

Average lead-time for an order in each production stage was defined by the triangular distribution with the following parameters: min=1080, mode=1440 and max=1800. No queues were defined for model locations such as Pressing, Degreasing, Jigging, Anodising, and Packing.

Currently PH_Enquiries are processed by 1 planner, and PC_Enquiries are processed by another 3 planners that spent about 70% of their time on planning operations. Working day duration is equal to 8 hours per day, from 9:00 till 5:00. Employment costs per year for planning staff are fixed.

The entire business/manufacturing model diagram is illustrated in Figure 1. The following controllable variables are defined in the model: 1) a number of planners that process enquires, response to customers and plan confirmed orders for production; 2) the response time for enquiries, and 3) planning time for confirmed orders. The time between arrivals of enquiries, customer response time to confirm or cancel enquiries, the probability an enquiry becoming

confirmed or becoming an order, and order processing time for different production stages are regarded as environmental variables in the model.

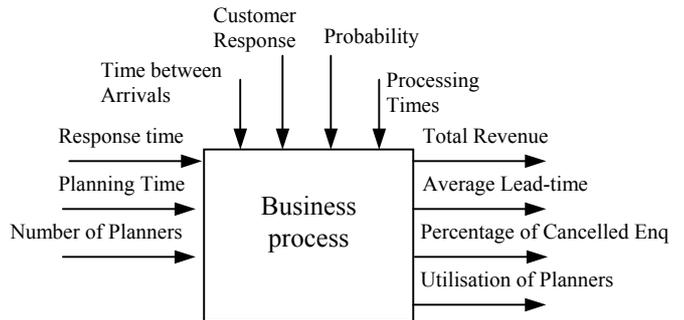


Figure 1: Black-box Diagram

The following simulation output data such as Total Revenue, Average Lead-time, Percentage of Cancelled Enquiries and Utilisation of Planners and are defined as the model performance indicators.

Visualization of the entire business/manufacturing simulation model is presented in Figure 3. On-line and off-line statistics is provided in the model. Outputs reflecting model dynamics could be followed on the model main screen. Simulation results are also automatically saved in a database and formatted in Excel sheets.

ANODISING STAGE SIMULATION MODEL

The anodising stage sub-model (Merkuryeva, et.al.) is aimed to determine whether the implementation of specific production orders queue ranking rules will improve the processes at a batch anodising plant. The model itself simulates the individual anodising tanks so that colour changeover, set-up operations and processing times can be modelled. Based on the historical data about order processing the most probable list of incoming orders to be weekly processed is generated in the model.

Orders scheduling rules are simulated and tested in order to decrease their total processing time of all aluminium parts to be anodised. Production rate that is an average number of flight bars processed per hour and frames utilisation coefficient are used to measure effectiveness of the anodising plant itself. The real system to be simulated is conceptualised in Figure 2.

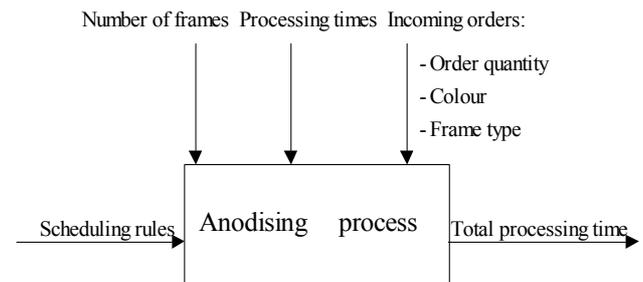


Figure 2: Anodising Model Diagram

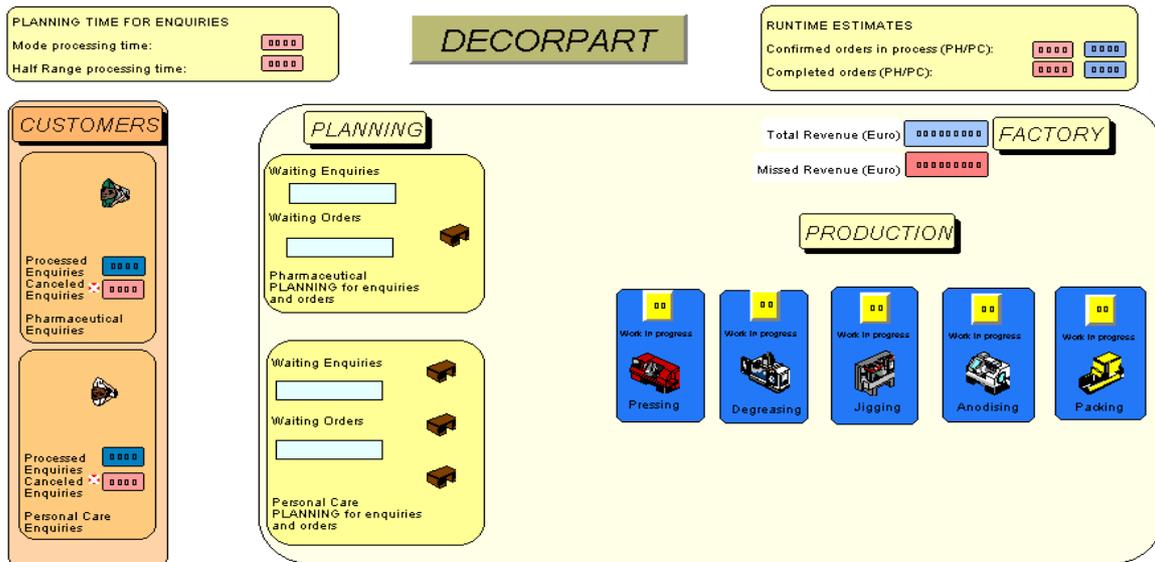


Figure 3: A High-level Business/Manufacturing System Simulation Model Visualisation

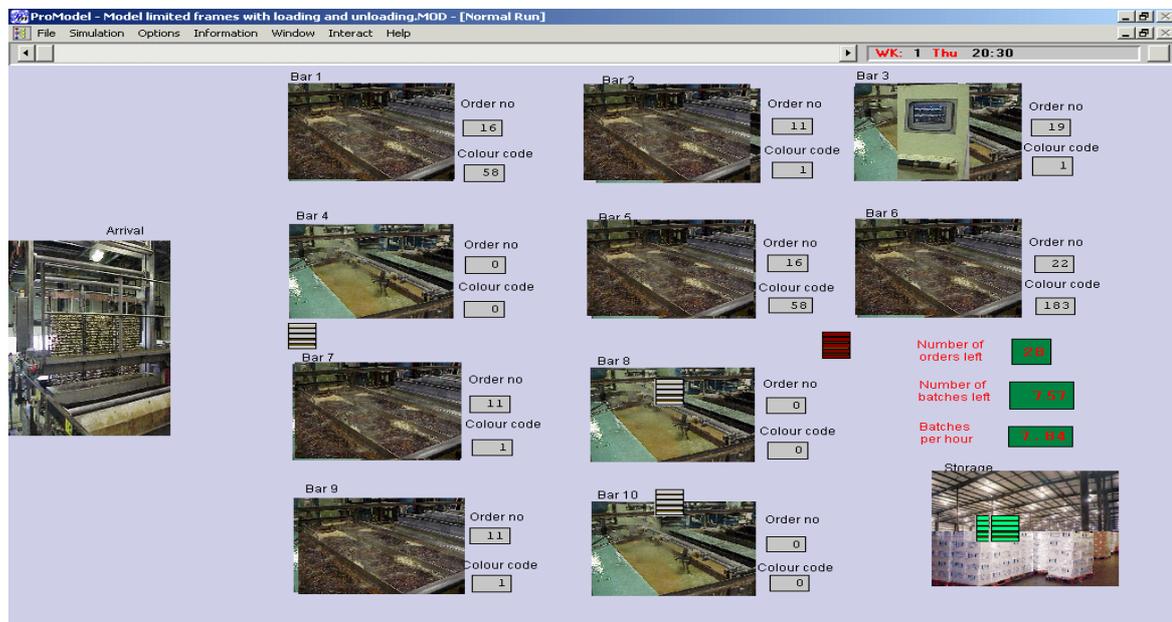


Figure 4: The Anodising Stage Simulation Model Visualisation

The following controllable variables are defined in the model: sequence numbers of orders to be processed in a week and the number of frames of specific types that are available at the plant.

The order quantity, part colour and used frame type for incoming orders are regarded as environmental or independent variables. If these properties are given, the other properties of orders in the order list can be determined. Other environmental variables are the number of frames in stock, the time it takes to load and unload flight bars, the time it takes to set-up flight bars between the processing of different colours and the processing time necessary to anodise one batch of components.

The most important performance indicator is defined as the total processing time of all orders in the order list. Among other performance indicators that could be used to control an anodising process in the real system, the following performance measures can be mentioned: an average production rate, frame loading efficiency, flight bars utilization and plant productivity.

The General Order List is created based on analysis of historical data about the orders that were planned and processed at the plant in a certain period. It includes the following input data: Week Number, Order Number, Order Quantity, Colour, Frame Type and Capacity, Frames in Stock, Number of Batches and Sequence Number (Tables 3, 4).

Table 3: Fragment of General Order List (1)

No	Week	Order No	Order Qty (x 1000)	Colour	Frame Type
1	1	1135	10001	Bright Silver	C1
2	1	1135	10134	Bright Gold	C1

Table 4: Fragment of General Order List (2)

No	Week	Frame Type	Frame Capacity	Frames in stock	Number of batches	Seq No
1	1	C1	1292	15	26	8
2	1	C1	1292	15	26	6

The last four digits of the order number Order No in Table 3 refer to the code of the colour that the components should get. Each frame type had a different number of components that can be placed upon it, which is called the frame capacity. The number of frames of specific type available is called as Frame in stock. Only three frames can be loaded on each flight bar. Processing time of one batch of components in a flight bar depends on the program that is used in the anodising process is defined by a Sequence Number (Seq No) in Table 4.

Based on the General Order List analysis done within Excel environment, processing times are described by the triangular distribution and generated in the simulation model. For example, for sequence 8, which is used by orders with colour code 0001 the triangular distribution with the following parameters min=54, max=72 and mode=58 minutes is used in the model. Frequency of order colour and order quantity in the General Order List as well as of the frame type to be used are defined by empirical distributions. For simplification it is assumed that order quantity and frame type depends on the product colour to be anodised.

Based on fitted probability distributions the Input Order List to be processed in a week is generated using so called transformation tables (Merkuryeva, et. al.). Example of the completed Input Order List is presented in Tables 5, 6. Let note that parameters of the probability distribution that fit processing times (such as minimum, maximum and most likely value), the number of batches that order should be split up in and the number of frames necessary to process all batches are also included in the Input Order List.

Table 5: Fragment of the Input Order List (1)

No	Colour code	Qty x 1000)	Frame Type	Frame Capacity	Processing time (min)
1	0058	28	7	3456	54
2	0003	225	2	1900	64
3	0001	224	6	3456	54

A screenshot of the model visualisation presented in the Figure 4 was created by animation of four pictures from the real company that simulates in the model order arrivals and storage as well as colour change over, set up and order processing operations. The user could follow the flow of batches from the arrival location and analyse the current stage of the anodising process for each order. Different colours are used for incoming and processed entities. Entities that are processed move on to the storage location.

Table 6: Fragment of the Input Order List (2)

No	Processing time mode	Processing time max	Batch No	Frames No	Frames left
1	58	72	3	9	0
2	87	92	45	135	0
3	58	72	22	65	2

On-line statistics is provided by three counters display the following performance characteristics of the anodising plant: the number of orders that are left to process, the number of batches left to process and the average number of processed batches per hour. Two additional counters along with the flight bars indicate the current number and the colour of the order that is currently being processed. Total processing time of all incoming orders, frames loading efficiency and plant utilisation can be found in the General Simulation Output Report.

OPTIMISATION OF BUSINESS PROCESSES

Based on the sensitivity analysis of the above-described high-level business/manufacturing system model it was concluded that decreasing response time for enquiries by 5 % would lead to increase of the company total revenue by about 10 %.

Response surface based simulation metamodelling analysis performed with MiniTab software showed that for both types of Enquiries the simulation model outputs are more sensitive to Enquiries processing time rather than to orders planning time. For example, for PC_Enquiries the following business/manufacturing regression-type simulation metamodel was received:
 $Lead-time (PC) = 9277.03 - 21.05 * Enq + 4.83 * Ordr + 0.62 * Enq^2 + 0.41 * Enq * Ordr.$

Optimisation of the model parameters within available system recourses that was performed using Promodel Simrunner Optimiser showed that total revenue maximal value could be received if inquiry response time would not exceed 6 minutes (Table 7). Actually this response time could be only achieved by introducing an automatic PREACTOR Supply Chain Planning Server at the company. Note, that the second optimal design defines optimal combination of enquiries and orders planning time minimising lead-time model indicator.

Table 7: Comparison of Two Optimal Designs

	Enquiry planning time	Order planning time	Revenue €	Lead-time PH (min)	Lead-time PC (min)
Maximised revenue	U(4,6)	U(2,8)	49,900,000	9218.2	9261.1
Minimised lead-time	U(1,11)	U(3,7)	48,210,000	9244.4	9134.7

Finally the following two planning scenarios were compared in the case study using the entire business/manufacturing simulation model: (1) Scenario 1 that corresponded to the current situation with maximum response time equal to 1 Hour per enquiry, not including queuing time; (2) Scenario 2 that uses automatic Preactor Supply Chain Server to respond, this time does not exceed. 0.1 Hour.

Results of simulation experiments while comparing above planning scenarios showed increasing of the Total Revenue, as well as decreasing the Average Lead Time, percentage of cancelled enquiries and essential decreasing utilisation of planners (Table 8). Shorter enquiry processing time provides faster response to the customer leading to higher probability for enquiries to become an order.

Table 8: Comparison of Alternative Planning Scenarios

	Lead Time (min)		Total Revenue (€)	Cancelled enquiries (%)	
	PH	PC		PH	PC
Scenario 1	10805	10414	17.170.588,24	57%	57%
Scenario 2	9793	9617	41.758.823,53	44%	39%

	Utilization			
	PH Planner	PC Planner 1	PC Planner 2	PC Planner 3
Scenario 1	93%	99%	98%	97%
Scenario 2	52%	73%	61%	52%

The total revenue was calculated only for replication at the process stable stage. The counters for completed orders were stated for the replications including the model warm-up period. The last one was estimated almost by three weeks. The replication length was defined as twice as warm-up period. While planning department works only on weekdays, production process is carried on 24 hours a day, seven days a week. After ten replications the variance in the output variable such as average lead-time was small enough to get a half range of five percent average.

TESTING PRODUCTION ORDERS PROCESSING SEQUENCING RULES

The scheduling of orders processing at a batch anodising plant could be interpreted as a finite capacity scheduling problem. The last one is defined as “the process of creating an operation schedule for a set of

jobs that are to be produced on a limited set of resources”. There is a limited set of resources in the case study, i.e. the number of frames in a stock and the number of flight bars that the frames are loaded on. Generally, in a batch anodising plant the following two problems can occur.

The first one is caused by multiple performing of set-up operations between the processing of orders with different colours. So, decreasing the number of necessary set-up operations will result in reducing the total lead-time at the plant. The second problem relates to a limited number of frames that are available for a specific frame type. Let assume that the last production orders in the list make use of the same frame type. Since this frame type is limited, it will cause queues of orders waiting for free frames, while the flight bars could be empty. The same problem could occur in case when the last order in the list that request a limited frame type is quite large.

As a result, the following order processing sequencing rules that provided simulation scenarios in the case study were analysed. Scenario A0 represents the initial situation, in which no specific sequencing rules are applied and the incoming orders are processed according to their arrival mode. In the other three scenarios, different sequencing rules are introduced. In scenario A1, the orders with the largest quantity of components are processed first. In A2, one handles the orders of one colour first, followed by the next colour. It is expected that this will reduce the total set-up time and hence will solve the first problem mentioned above. In scenario A3 the colours that appear less frequent in the list are processed first while within the group of the same colour, the orders with the largest number of components are processed first. Because the orders are grouped per colour, this could partly solve the first problem and processing of the largest order within one colour could solve the second one.

In order to determine if the implementation of one of these sequencing rules improves the anodising process, the orders in the input files are rescheduled in the way the scenarios describe. Then difference in the total processing time of all incoming orders is calculated for scenarios with sequencing rules and the scenario in which no specific rule is applied. In this case, for each replication, common random numbers are used to simulate both scenarios that lead to a lower variance in the estimation of the difference in total processing time between different sequencing rules.

While treatment simulation experiments, the length of the anodising model run is accepted equal to the time between the start of the week that represents the initial situation in the real system and the time that all week orders were processed. As probability distributions are used both in the generation of input data and in the simulation model itself to define the time it takes to

anodise the parts at flight bars, the number of necessary replications was determined while generating the input data as well as running simulation experiments. The number of random seeds initially set to twenty was reconsidered while comparing alternatives.

In the case study, mean difference between each specific sequencing rule and the initial scenario was estimated with 0.95% confidence interval. While comparing scenarios A0, A1, 20 replications were performed and the resulting mean difference was estimated by 11,51 hours with 95 % confidence interval equal to (3.82, 19.9) hours. As a result, it was concluded that a significant improvement could be provided at the plant if the rules of scenario A1 are used in for order planning.

Performing What-if analysis allowed testing whether the implementation of the A1 scheduling rule would still be an improvement if the number of frames in stock could be increased. In this case frames are not considered as limited resource in the real system. The results of comparison of specific sequencing rules with unlimited frames showed that in the last case A1 scenario will not make a significant improvement compared to scenario A0 (Table 9). Actually, sorting the orders by colour according to scenario A2 will decrease the total processing time at least by 5.65 hours. At the same time, there will be no significant difference between scenario A2 and A3.

Table 9: Comparison of Alternative Sequencing Rules

Scenarios		Mean difference (hours)	95% confidence interval	Significant
A0	A1	0.01	(-0.55,0.58)	No
A0	A2	6.27	(5.85,6.89)	Yes
A0	A3	6.23	(5.59,6.86)	Yes

CASE STUDY BENEFITS

The modular simulation models provide an inexpensive tool for an overall guidance of piece-part SME manufacturing and testing advanced scheduling middle-scale software packages prior to their implementation at the customer's site. To test and optimise a production scheduler using the simulation software is much faster and easier than using the real process.

System performance output such as overall lead time and process stage lead time by order, average overall lead time of production and average lead time at each stage, received from the simulation models provides system set-up advice on: 1) production activities in order to maximise equipment utilisation decreasing unit manufacturing cost; 2) forward projections for delivery times for new orders, and 3) schedules for product change over at each stage.

Comparing two alternative planning scenarios using the entire business/manufacturing model proved the

benefits of introducing automatic PREACTOR Supply Chain Server with maximum respond time 0.1 hour per inquiry. In this case percentage of cancelled enquiries could be decreased by 14-18 % providing increase of the total value of confirmed orders at least twice. Another benefit is expected in utilisation of planning staff as instead 4 planners only one could be needed. Amount of employment costs to be saved is evaluated by 150 000 Euro per year.

The results of simulation experiments with the anodising stage sub-model demonstrated that introducing new specific sequencing rules for incoming orders could provide significant improvements. The total lead time of anodising all aluminium parts from a week order list could be decreased at least by 4 hours up to 19 hours. As a result, production rate at the anodising stage could be increased by 10%, and a significant increase in equipment utilisation and reduction in a unit manufacturing cost could be received.

CONCLUSIONS

The simulation tool presented in the paper could be proposed for piece-part SME manufacturing in order to test advanced planning and capacity optimisation middle scale software packages. Iterative optimisation of initially configured scheduler parameters and rules could be performed by testing them using business/manufacturing simulation models built in a modular style. It should be noted here that the approach used – to test and optimise planning and control tools off-line by using simulation models rather than using the real process – can be applied to many other software tools, to higher level (MRP; ERP tools) as well as to lower level control tools (MES, warehouse control systems). On other side, development of such simple simulation tool in different industrial sectors could provide also an inexpensive approach to an overall guidance of SME manufacturing towards the optimal conditions without resource to high cost integration of expensive ERP systems and downstream control systems.

ACKNOWLEDGEMENTS

The authors take pleasure in acknowledging the contribution made by Morrison Richard, IT Manager of Decorpart Ltd company. The authors also take pleasure to thankful Johannes Krauth, Operational Manager of the Sim-Serv Virtual Centre for Simulation for support of the performed research.

REFERENCES

- Krauth J. 2002. . "Sim-Serv the Virtual Simulation Institute". *Simulation News Europe*, Issue 35/36, (Dec).
- Merkuryeva G., Shires N., Morrison R., Mark de Reuver. 2003. "Simulation based scheduling for batch anodising processes". In *Proceedings of the International workshop on Harbour, Maritime and Multimodal Logistics Modelling & Simulation*; Riga, Latvia, Riga Technical University, 170 – 176.

IMPROVING THE REMOTE SCHEDULING OF MANUFACTURING AND INSTALLATION OF LARGE CUSTOM-MADE PRODUCTS

Romeo Bandinelli
Dipartimento di Energetica
“Sergio Stecco”
Università di Firenze
Via S. Marta 3
50139 Firenze, Italy
r.bandinelli@ing.unifi.it

Mauro Iacono
Dipartimento di Ingegneria
dell’Informazione
Seconda Università di Napoli
Via Roma 29
81031 Anversa (CE)
mauro.iacono@unina2.it

Alessandra Orsoni
School of Business Information
Management
Kingston University
Kingston uponThames
Surrey KT2 7LB, UK
a.orsoni@kingston.ac.uk

KEYWORDS

Supply Chain Management, Remote Production Scheduling, HLA Federations, Manufacturing, Railways Switch Point Assemblies.

ABSTRACT

This paper illustrates the use of statistical inference techniques to improve the scheduling performance of a distributed simulation and scheduling tool based on the XLA-RTI architecture. Specifically, the timing of federation synchronisation events can be customised according to the local failure characteristics of the physical system so as to reduce the necessary delays in information and communications exchange among federates. Ultimately, these improvements can only be beneficial if they have measurable effects on the production performance of the physical system. The paper refers to an industrial application case to measure the production benefits that may be accrued by implementing the proposed methodology. The application context is the manufacturing and installation of railways switch point assemblies, which, to a large extent, are custom-made products in that they come in a multiplicity of assembly types and ultimately need to be customised to fit the specific application requirements.

INTRODUCTION

The research presented in this paper builds from the outcomes/developments of a project aimed at improving the manufacturing and installation processes of railways switch point assemblies. The original project focused on the activities of an Italian company that produces railways switch point assemblies for the national railway network as well as for industrial users. Part of the assemblies are produced for installation by third parties while others are installed directly by the company’s own installation crew. Earlier studies on the performance of the company’s operations led to the development of an improved HLA federation of simulation and scheduling modules, which reduces the impact of communication delays on scheduling performance, while retaining the benefits of de-coupled scheduling procedures and process control. The initial project involved the development of a simulator to analyse the current performance of the company and

search for possible process improvements. An initial set of simulation-based experiments targeted the assessment of a number of performance measures including the percentage of on-time deliveries, resources utilisation, and production lead-times (Gunasekaran, 2001). During the course of the analysis, the drilling and the planning machines were identified as critical because their production rates appeared consistently lower than the others’, thus creating the conditions for potential bottleneck effects. While this observation suggested that an investment might be advisable for the company in order to increase the capacity of these machines, the lack of a structured approach to the scheduling of incoming orders emerged as a critical performance issue. The company, which is relatively young (they have been in the business for approximately two years), had never really invested time and effort in the development of appropriate scheduling policies, and production performance clearly suffered, as reflected by the large percentage of late product deliveries (over 35%). A scheduling tool was then developed and linked to the existing simulation models of the manufacturing and installation processes to test the impact of alternative scheduling policies in relation to the current status of the machines on the shop floor and to the status of the installation process on the designated site (Chang and Maskatsoris, 2001). When the system is used in the loop with the production and installation processes, the first step is to rank the existing orders by due date and estimated production time. Then, based on the progress status of the current production and installation activities, the system decides whether to produce an assembly for third party or for own-crew installation.

The additional research presented in the paper refers to the integration of the simulation and scheduling modules into an HLA-RTI federation architecture, which enables the maintenance and update of the modules in separate locations: the scheduling module near the decisional centre and the simulation modules near the production centre. The integration of the modules into a single federation raises important performance issues related to the frequency of communication exchanges among federates (Juhasz et al., 2003; Hwang, 1993). To address these issues the

paper proposes an approach based on statistical inference techniques to tailor the timing of synchronisation to the process characteristics of the simulated system with the double objective of reducing wasted simulation time and thereby improve the performance of remote scheduling (Bandinelli et al., 2004). The paper shows how this is achieved by applying the methodology to the scheduling of manufacturing jobs in the production and installation of railways switch point assemblies as described in the first part of this introduction.

Prior to illustrating the methodology a brief description of the actual production and installation processes will be provided to highlight the key characteristics of the processes and provide the context for the statistical analysis. As far as results the paper will provide a comparison between the scheduling performance corresponding to the standard synchronisation approach and the one measured using process statistics to time synchronisation.

MANUFACTURING AND INSTALLATION OF RAILWAY SWITCH POINT ASSEMBLIES

Railway switch point assemblies are pre-fabricated units including all the rail elements and connecting devices that are required to lead a train from its current track onto a different one, an example is shown in figure 1.



Figure 1: Example of Railway Switch Point Assembly

Each assembly consists of seven or more individual rail elements, depending on the particular design (single, double, or intersection) and on the required deviation angle. According to the industry practice, some rail elements may be recycled from old, decommissioned rail systems, others are processed from scratch starting from standard twelve-metre long rail segments. The different rail components pertaining to a given order follow similar manufacturing steps prior to their assembly into the final unit. These steps include cutting to the required length, drilling, bending, upper surface planning, and lower surface planning. The last two activities, depending on the type of component and on the design specifications, may be followed by a quality check on the deviation angle and re-bending, if needed. Most of the machining jobs take one to two hours to

complete with the exception of upper and lower surface work that may take four to seven hours, depending on the particular size and design of the assembly. The production sequence includes cutting, drilling, bending, planning and milling, final bending, and partial assembly. The layout of the shop floor is shown in figure 2.

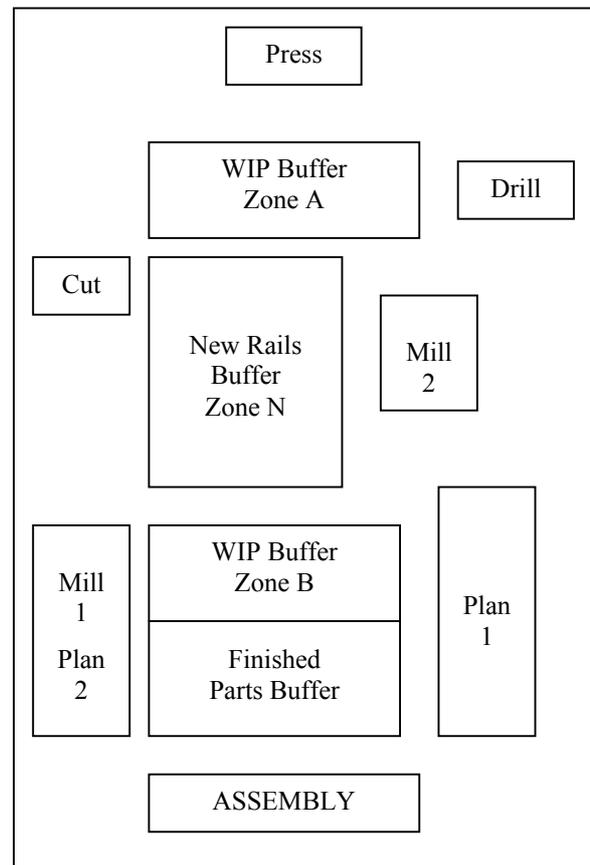


Figure 2: Layout of Manufacturing & Assembly Floor

The installation steps include excavation, placement of transverse units, placement of the switch point assembly, filling with gravel, levelling and alignment, and connection.

PROCESS SIMULATION & PRODUCTION SCHEDULING

Simulation is a widely accepted means of supporting real-time production scheduling because it provides estimates for the state of every machine at the time when a new order needs to be sent to production (Bandinelli et al., 2003; Williams et al., 1999). The scheduling tool provides correct results when fed with the current production parameters, machine status, and order status for the entire production system (Proud, 1999). The simulation models for the representation of the manufacturing and installation processes described in the previous section were developed using a commercial simulation package, ARENA 7.0. When

simulation is used in the loop for on-line scheduling, time-performance becomes critical towards the choice of the right scheduling policies: the simulation estimates need to be available prior to the completion of the current job on each machine (Bandinelli et al., 2003; Bandinelli et al., 2004).

In the distributed simulation of stochastic processes, a major driver of time-performance is wasted simulation time related to the timing of synchronisation events among federate simulators. This is especially true when the simulated processes are stochastic: for instance the current production capacity of a plant depends on the number of machines that are currently down due to failures, while the ability to manufacture a particular item depends on the availability of the required raw materials in stock. The state of a machine (up or down) and the stock levels for different raw materials are random variables that need to be described through their probability distribution in the process simulation models.

For the representation of the mean times between failures of the individual machines, triangular probability distributions were built based on both historical data and on-site observations. Historical data for the entire period of operation of the company (nearly 2 years' worth of data) were also used to build probability distributions for the monthly order inter-arrivals. The strong variability in the data collected for this project, partly explained by the fact that the company is relatively young, causes the required length of the simulation run to be fairly long (57 months) as shown by the MSPE curve.

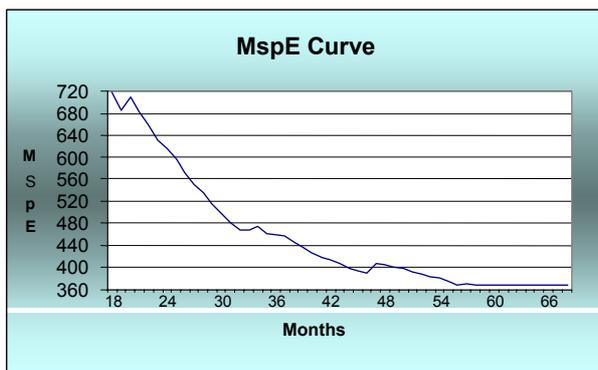


Figure 3: MspE Curve for Production Lead-Times

On the other hand, the scheduling tool was devised outside the ARENA modelling environment and physically located in the vicinity of the decisional centre. This approach provides the advantage of separating the development and the maintenance of the physical model from those of the scheduler. On the implementation side, this approach requires an infrastructure to handle the exchange of information between the physical model and the scheduler. Due to

the high level of control that needs to be achieved over the production plan, and due to the choice of HLA-RTI as a framework for Inter Process Communication (IPC), an additional software had to be developed for the implementation of the framework. The additional software handles the interactions between the ARENA simulator and the RTI environment, and between the RTI environment and the control system. Specifically, the role of this software is to collect the information exchanged by the RTI environment in an asynchronous way, and to communicate this information to the physical model and to the controller, which work in a synchronous way.

The realisation of this software component may follow two different approaches (Tucci et al, 2001). The first one can be summarised in the definition of a Delegated Simulator module, which co-ordinates the logic of information exchanges among federates. The second solution proposes the introduction of a software tool “living” between the simulator and the RTI. This software, namely a proxy, is in charge of the two-way communications between the RTI environment and the simulator (Orsoni et al., 2003). The additional flexibility provided by the use of a proxy led to choose this second approach, which makes the Proxy entirely responsible for the information exchange between the federates and the RTI. While the simulator performs communication tasks in a synchronous way, by TCP-IP, the RTI performs these task in an asynchronous way: the proxy is then required to store the information coming from the RTI and transfer it to the relevant federates at the nearest synchronisation event and vice-versa.

For the purposes of time-management, a “next event” approach was chosen because the times when information exchanges would be needed could not be known a priori. In order to optimise the times when the federation stops simulating to allow for information exchanges between the physical model and the controller, a methodology based on statistics from the MTBF distributions of the critical components has been used. Details on the methodology and its application to the industrial case are provided in the following section. An important advantage of the devised infrastructure is the possibility to develop the scheduler as a separate unit using either a simulation package, such as ARENA, or an independent programming language.

A typical iteration of the communication process could be described as follows. Upon completion of a production/installation activity an event occurs which stops the simulation clock. The new state of the physical model is communicated to the proxy and thereby passed on to the RTI environment, which in turn delivers this information to the scheduler. The jobs sequence determined by the scheduler is then communicated to the physical model by the RTI, where the relevant job attributes (i.e. type of activity, activity identification code, work centre identification code, start time, and

activity duration) are stored as an XML string. Additional details on this infrastructure can be found in [Bandinelli et al., 2003].

In order to validate the model and obtain suitable test results, some simplifications were introduced. In the first instance, the scheduling procedure was based on simple rules and the communication between the scheduler and the simulator was emulated to validate the proposed scheme and test the performance benefits of the methodology (Sargent, 1999). Because in the authors' opinion, HLA is still the most complete and mature IPC Standard, a simplified emulator of such infrastructure was built to reproduce the Next Event time management procedures for the communications between the scheduler and the physical model. In this respect it is important to remember that the aim of this work was to provide a new methodology to reduce delays in communication and information exchange, and not a new IPC standard.

PROCESS STATISTICS FOR THE TIMING OF SYNCHRONISATION

Changes in state and in the values of the dynamic process attributes are first recorded at the federate level and can only be communicated to the other federates (i.e. other simulation models, control system, scheduling module) when a synchronisation event occurs at the federation level. Because of the random nature of state and variable changes it is impossible to schedule synchronisation events a-priori that match the timing of such occurrences. The correct frequency of synchronisation and thus of communication/information update among federates is always the result of a trade-off between the time spent on communication (too long if communication is frequent), and the likelihood of not sharing the occurrence of asynchronous events with other federates for too long a time (if communication is infrequent), which may invalidate large batches of simulated time. Ideally, the timing of federate synchronisation should reflect the characteristics of the simulated processes building from the most critical points, intended as the ones that are most likely and thus most frequently affected by asynchronous occurrences. Prior research work by the authors developed a methodology to improve the performance of federation runtime infrastructure in the scheduling of manufacturing processes (Bandinelli et al., 2004) The methodology addresses the issue of communication delays among federates by introducing predicted failure events that are tailored to the process statistics of the physical system. Specifically, the methodology refers to the Mean Time between Failures (MTBF) distributions for the critical process components to generate histories of predicted failure events that can be fed to the federation control level to customise the timing of synchronisation. By these means, the impact of asynchronous events - generated locally at the individual federate level - on the time performance of the federation can be significantly reduced. Preliminary

results for a simple manufacturing application case (Bandinelli et al., 2004), have shown a potential for the reduction of wasted simulation time up to 35%. Building from these results, the example analysis presented in this paper, establishes how the time-performance benefits accrued at the simulation level may improve scheduling and production performance in the physical system. With reference to the critical machines (i.e, the planning and drilling machines) a series of random extractions were made out of the corresponding MTBF distributions to obtain statistical estimates of their likely "next-failure" events. The timing of each predicted failure event was obtained comparing the average time to the next failure for each machine over a set of 20 extraction. The shorter value was then corrected adding a fraction of the corresponding standard deviation, as predicted failure events should follow the actual occurrence. A series of test runs were conducted to experimentally set the value of this fraction so as to minimise communication delays. Ideally, the objective is to schedule each predicted failure event right after the occurrence of an actual failure in the physical system, so that the occurrence can be seen by the federation with a minimal delay. The fraction of the standard deviation that is used to correct the average estimated time to next failure for the system is context specific, and for this application was experimentally set to 0.4. The process is iterated until predicted failure events have been scheduled to cover the entire simulation horizon.

EXAMPLE APPLICATION & RESULTS

For the purposes of this example application, the timing of federation synchronisation was customised using the process statistics identified in the previous section. By running the federation in the new condition it was observed that the percentage of on-time deliveries had increased by 40% (from over 35% late deliveries to less than 20%) and the cumulative delays in late deliveries has decreased by 28%, which show a relative improvement in the performance of remote scheduling with respect to the original federation.

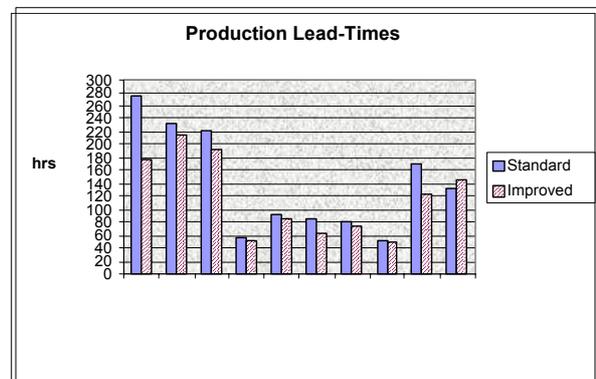


Figure 4: Improvements in production lead-times

A comparison between the production lead times observed in the physical system when scheduled using

the standard and the improved synchronisation time management is presented in figure 4. The figure compares the lead-times for the main types of switch point assemblies produced by the company in the two cases.

Figure 5 presents a comparison between the percentage of late deliveries when scheduling production using the standard and the improved synchronisation time management approaches, respectively.

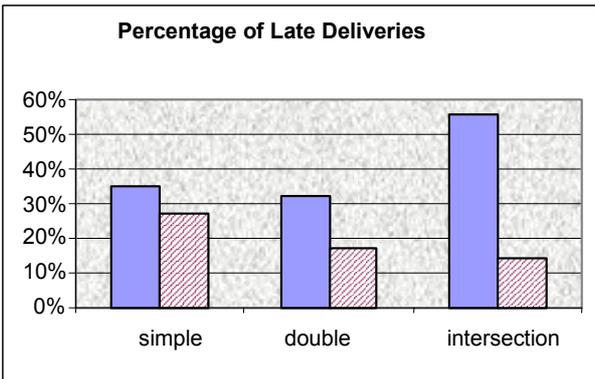


Figure 5: Reduction in the percentage of late deliveries

In this figure the types of switch point assembly are grouped into three main categories: single, double and intersection types. A relative improvement can be observed for each of these categories.

In summary, the analysis applied to the manufacturing and installation of railways switch point assemblies shows that the improved communication among simulation federates makes the scheduling tool more responsive to the dynamic needs of production and installation. Both scheduling and production performance are improved as indicated by the increase of on-time deliveries and by the reduction of the cumulative delays on late deliveries.

CONCLUSION

The availability of timely updates on process state variables makes scheduling of manufacturing and installation processes far more effective, as indicated by the relative increase in the number of on-time deliveries and by the reduction on cumulative delays for late deliveries.

The strength of the statistical inference method used to determine the appropriate timing of communications among federates is that it reflects actual process characteristics, which makes the timing of communication as responsive as possible to the occurrence of otherwise asynchronous events.

Future work will focus on the development of a detailed fault model to capture patterns of failures

across production units and processes. AI techniques based on Artificial Neural Networks will be used to establish correlations between local failure modes and their impact on the productivity of other production units with the aim of supporting rescheduling activities around predicted failure events. Further research will also look at AI techniques as a more efficient way of generating histories of predicted failure events based on the characteristics of the MTBF distributions for the critical process components, and on the nature and location of the last failure event.

REFERENCES

- Bandinelli, R., M. Rapaccini, S. Terzi, and M. Macchi. 2003. "Proposal for a Framework for Production Plants Remote Control: a Preliminary Test Case". In *Proceedings of the 2003 European Simulation Symposium*. SCS-Europe, Delft, NL, 343-348.
- Bandinelli, R., M. Iacono and A. Orsoni. 2004. "Anticipating Critical Events to Customise and Improve the Performance of Federation Runtime Infrastructure". In *Proceedings of the 2004 UKSim Conference*. Oxford, UK, 59-64.
- Chang, Y and H. Maskatsoris. 2001. "Supply Chain Modeling using Simulation". In *International Journal of Simulation Systems, Science, and Technology*, 2(1), 24-30.
- Gunasekaran, A., C. Patel and E. Tirtiroglu. 2001. "Performance Measures and Metrics in a Supply Chain Environment". In *International Journal of Operations and production Management*, 21(1), 71-78.
- Hwang, K. 1993. *Advanced Computer Architecture – Parallelism, Scalability, Programmability*. McGraw-Hill, New York, NY.
- Jain, R. 1991. *The Art of Computer Systems Performance Analysis*. J.Wiley & Sons, New York, NY.
- Juhasz, Z., S. Turner, K.Kuntner and M. Gezson. 2003. "A Performance Analyzer and Prediction Tool for Parallel Discrete Event Simulation". In *International Journal of Simulation Systems, Science and Technology*. 4(1), 7-22.
- Orsoni, A., Bruzzone A. and R. Revetria. 2003. "Framework Development for Web-based Simulation Applied to Supply Chain Management". In *International Journal of Simulation Systems, Science and Technology*. 4(1), 1-6.
- Proud, F.J. 1999. *Master Scheduling. A Practical Guide to Competitive Manufacturing. Second edition*. J.Wiley & Sons, New York, NY.
- Sargent, R.J. 1999. "Validation and Verification of Simulation Models". In *Proceedings of the 1999 Winter Simulation Conference*, IEEE, Piscataway, N.J., 39-48.
- Tucci, M. and R. Revetria. 2001. "Different Approaches in Making Simulation Languages Compliant with HLA Specification". In *Proceedings of the 2001*

Summer Computer Simulation Multiconference.
(Orlando, FL, Jul. 15–19). 622-628.

Williams E.J. and R. Narayanaswamy.1999.
“Application of Simulation to Scheduling,
Sequencing, and Material Handling”. In *Proceedings
of the 1999 Winter Simulation Conference.* IEEE,
Piscataway, N.J., 861-865.

AUTHOR BIOGRAPHIES

ROMEO BANDINELLI received his laurea degree in Mechanical Engineering from the University of Florence in 2002 discussing the thesis “Remote Factory Control with Distributed Simulation”. He is currently a PhD candidate in the Department of Energetic, Plants and Industrial Technologies at the University of Florence. His research interests include parallel and distributed simulation for industrial and supply chain applications, ICT, and business process re-engineering.

MAURO IACONO is adjunct professor and research associate at the Second University of Naples (SUN). He obtained his Laurea degree in Ingegneria Informatica in 1999 from the Federico II University of Naples. 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 has been a consultant of the Italian Ministry for Innovation and Technology (MIT) within the e-government national plan.

ALESSANDRA ORSONI is currently a senior lecturer in the School of Business Information Management at Kingston University (Kingston, UK). She received both her MS in Mechanical Engineering and her ScD in Engineering Systems Design and Innovation from the Massachusetts Institute of Technology (Cambridge, MA). Prior to joining Kingston University she was a research associate in the Department of Materials Science and Metallurgy at the University of Cambridge (Cambridge, UK) and worked as an assistant professor at in the Department of Production Engineering at the University of Genova (Genova, Italy). Her research interests include modelling, simulation, and AI techniques applied to the design and management of complex industrial systems.

AGENT-BASED SIMULATION

DISTRIBUTED OPTIMIZATION OF REFERENCE TRAJECTORIES FOR ACTIVE SUSPENSION WITH MULTI-AGENT SYSTEMS

Eckehard Münch
Oliver Oberschelp
Thorsten Hestermeyer
Mechatronics Laboratory Paderborn -
University of Paderborn
Pohlweg 98, D - 33098 Paderborn, Germany
E-Mail: Oliver.Oberschelp|Thorsten.Hestermeyer@mlap.de
Eckehard.Muench@mlap.de

Peter Scheideler
Andreas Schmidt
Heinz Nixdorf Institute -
University of Paderborn
Fürstenallee 11, D - 33102 Paderborn, Germany
E-mail: Peter.Scheideler|Andreas.Schmidt@hni.upb.de

KEYWORDS

multi-agent systems, mechatronics, distributed optimization, railway systems

ABSTRACT

Self-optimization in mechatronic systems is not restricted to inner processes within a technical system. By communication between different technical entities of the same or similar kind using appropriate communication structures, a technical system can use the experiences of other entities to optimize its own behavior. Multi-agent systems together with agent-based controllers are an excellent means to model the software of such collaborating systems. This paper suggests a multi-agent system for the self-optimization of the active suspension system of small railway vehicles. It also presents an architecture for the distributed optimization of this complex mechatronic system. The shuttles use their experience to find an optimal trajectory for the active suspension along the track, thus using the ability of the active suspension to deviate a small distance from the actual track trajectory.

INTRODUCTION

Mechatronic systems are complex technical systems whose dynamics are controlled using actuators, sensors and modern information processing. The continuous improvement of microprocessors and control units allows a more and more complex information processing, which is used to further improve the dynamic behavior of mechatronic systems. In the end, systems are supposed to adapt to changing environmental and internal conditions on their own, which leads to self-optimizing systems (SFB614 2004; SFB376 2004). In order to achieve this goal it is necessary to combine modern control theory with methods of artificial intelligence. The optimization can be done locally within the control system, where each controller is managed, monitored and optimized by its own agent (Oberschelp et al. 2002). These systems will be called agent-based controllers from here on.

It is quite obvious to interconnect agent-based controllers, in order to exchange experiences between systems of the same kind or to reformulate the objectives. While doing so it is of

utter importance to consider as many influences as possible. The collaboration of agent-systems leads us to multi-agent-systems (Ferber 1999). The following pages present a concept that shows self-optimization by multi-agent-systems. As application example the authors choose the *Railcab*-system (Railcab 2004) - a novel railway-system based on driverless autonomous shuttles driven by linear motors, which transport passengers and goods directly to their respective final destinations. The focus lies not on the logistics but rather on the organization of knowledge within a complex, distributed system and how to use this information to optimize the dynamic behavior of the system.

The system behavior can be improved in many ways. The physical plant, consisting of actuators, sensors and supporting mechanical structure, can be improved, but usually not during operation. The main handle for optimization is thus the controller. Here, usually the controller parameters are changed. The overall behavior of a system depends however not only on internal processes, but also on external reference values. Many control systems have to move from one state to another by moving along a given trajectory (Föllinger 1994). If this trajectory can be chosen freely, as with industrial robots, the overall behavior of the system can be changed by optimizing the trajectory. If the quality of the trajectory can be assessed by given accessible objectives, optimization can be done with a model or directly with the real system.

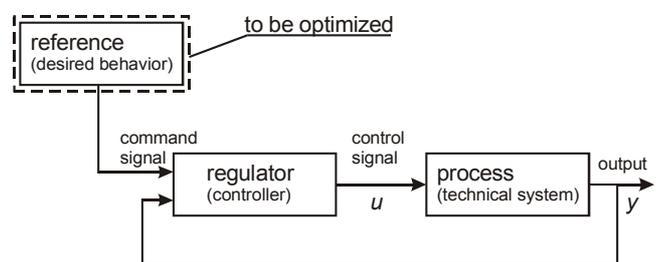


Figure 1. Common Control Structure

Fig. 1 shows the general structure of a control loop according to (Aström et al. 1989). The procedure described in this paper optimizes the desired behavior by means of the modification of trajectory defaults.

Optimization of a process can be performed online only if

the process can be repeated often enough. The recorded values of computed objectives for each repetition can be called *experience* of the technical system, that can be weighted and used to optimize the system behavior. In the example here a shuttle shall run along a fixed track sector with optimum comfort, limited only by the technical constraints of the suspension. As all shuttles are considered here to be basically the same, the experience of the shuttles can be used for optimization. Each shuttle is thus a probe for comfort. In order to use the shuttle experience it is necessary to communicate its knowledge and optimize in a distributed way.

In conventional systems the reference trajectory is computed offline based on static data (Wahl and Gilles 2003). In addition to this it is even not usual to control behavior of an active suspension via given trajectory (Streiter 1996). The usage of a static approach is only feasible if the environment is static and easy to measure. A centralized trajectory optimization on a global model is also possible but due the fact that the optimization problem results into many local optimization problems a decentralized optimization is more sufficient.

This paper presents a multi-agent-system for distributed optimization of trajectories, which practically corresponds to learning through individuals.

The remainder of the paper is organized as follows:

First we give an overview of the railway system and classify its relations to multi-agent systems. Then we present the active suspension system in detail. After this we discuss the use of the multi-agent approach to optimize the course planning. At last we give a short conclusion and an outlook to our future work.

STRUCTURE OF RAILWAY SYSTEMS

Modern railway systems have to compete with individual transport in respect of comfort, flexibility and cost. A research project which accepts the challenge is the Neue Bahntechnik Paderborn. The project has been initiated and worked upon by several departments of the University of Paderborn and the Heinz Nixdorf Institute. In the project, a modular rail system is being developed; it is to combine modern chassis technology with a new linear motor similar

to that of the Transrapid and the use of existing rail tracks. The interaction between information technology and sensor/actuator technology paves the way for an entirely new type of mechatronic rail system. The vehicles designed apply the linear drive technology used in the Transrapid, but travel on existing rail tracks. The use of existing rail tracks will eliminate an essential barrier to the proliferation of new rail bound transport systems (Hestermeyer 2003).

Distributed Information Processing. While classic railway-systems use centralistic approaches to control the trains on the railway network, this is virtually impossible with the high number of vehicles in the NBP system which move without time table on routes according to their demands. Thus, several decentralized approaches are being tested for realization of the logistic system, which manage shuttles and track using varying structures. However, all have in common, that the overall functionality is structured in a decentralized and function-oriented way. This requires the task distribution to different autonomous components, which have to exchange information, execute plans and perform actions on their own initiative. An approach that can fulfil these requirements on the information processing is the multi-agent-technology. It is therefore obvious to model the whole information processing on the planning level according to this scheme (Epple 2000).

The example in this paper uses the following hierarchical structure: A regional track network forms a *region* on the uppermost level. This region comprises a multitude of stations, junctions and tracks, which are again assigned to a *local region*. The local region is arranged in stations, switches and *track segments*. The latter are formed by *track sections* between stations and switches. This example concentrates only on the level track segments and track sections.

The overall system is administered by a multi-agent system, which performs partly hierarchically, partly cooperatively functions of the administration, as track allocation, energy management or shuttle localization. As a consequence, the resulting agent structure corresponds to the hierarchical structure of the railway system. The shuttles themselves form autonomous elements within the system. The liberty is limited, however, as each shuttle has to submit to a local management and cannot act in a completely free way.

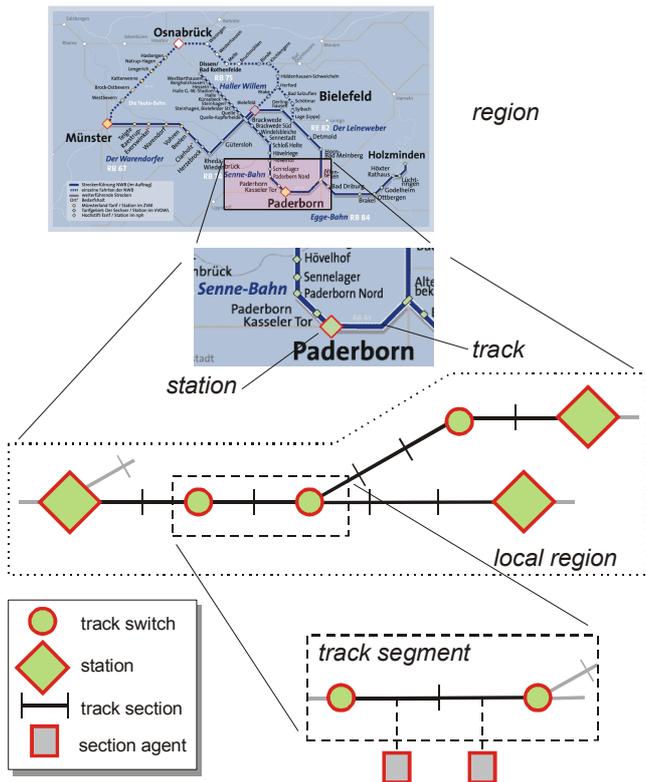


Figure 2. Hierarchical organization of the rail system

This information structure - set-up according to the functional and technical structure - now allows a clever distributed optimization using learning through individuals. The described approach assumes that a large number of vehicles can move nearly autonomously (not freely!) on a complex track network. During its journeys a vehicle collects data, which can be used as experience. This can be - like in the example here - measured data about the dynamic behavior during the transit of a certain track. This data is passed on to other vehicles. Separate measurement runs are not necessary any more.

At first look it seems a good idea to have each shuttle with all the others in order to exchange their experience. However, finding a shuttle with a certain feature, like one that has measured data about a certain visited track section is costly. In addition a shuttle has to save data about a given track section even though this data might be already obsolete because of experience made by other shuttles - an adjustment of the data of all shuttles would be necessary. It is therefore preferable to collect data at a track section and to have it administered by an agent that is assigned to a fixed local track section. Each passing shuttle communicates its experience in form of measurement data to this local knot and requests already processed experience of other shuttles.

The shuttles and track sections therefore form a multi-agent-system, that can be used for the management of the experience made by different shuttles. The connection of the agent technology to the technical processes in a shuttle requires a closer look of the technological and information structure of the vehicle and the integration in the overall system.

Structure of the Multi-Agent-System. Basically, the shuttles have a cognitive behavior to individually process the knowledge coming from other entities. However in this example an essential feature of a cognitive agent, its autonomy, is replaced with a Master-Slave-Cooperation (Nwana et al. 1996). For two agents connected by this relationship, the master sends its commands or arrangements to the slave and the slave will act as the master instructs, providing the feedbacks and results to the master. This means that the planning of the trajectory is done by the track sections (Masters). They send the trajectory parameters to the shuttles (Slaves) which have to execute them (see Fig. 3).

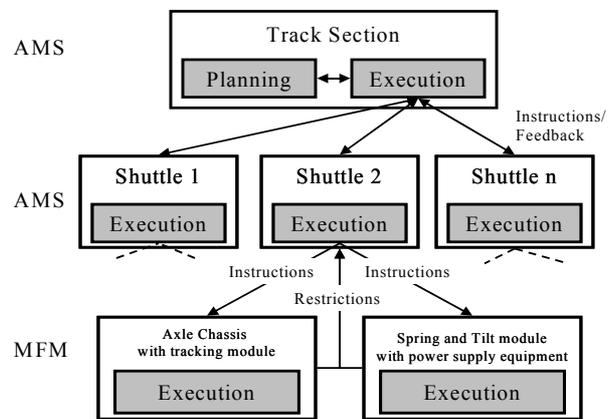


Figure 3. Master-Slave-Cooperation

Agent Micro-Achitecture. The software realization of the described structure requires a modular approach that fulfills the requirements of control theory (real time requirements, stability) while offering a link to multi-agent systems. In addition, single modules shall also be equipped with the ability for self-optimization. An approach to fulfill this requirement is the operator-controller-module. It is a structure-concept for the development of agent-based controllers (Oberschelp et al. 2002).

The basic idea is the separation of the control code into three different levels, which have different requirements. The lower level motor level implements the basic functions of the control which have direct impact on the technical process. This level is managed by the reflective operator, which is located on the next level. Control of reconfiguration within the controller, which includes emergency routines, but also the realization of parameter changes are some of the tasks of this operator.

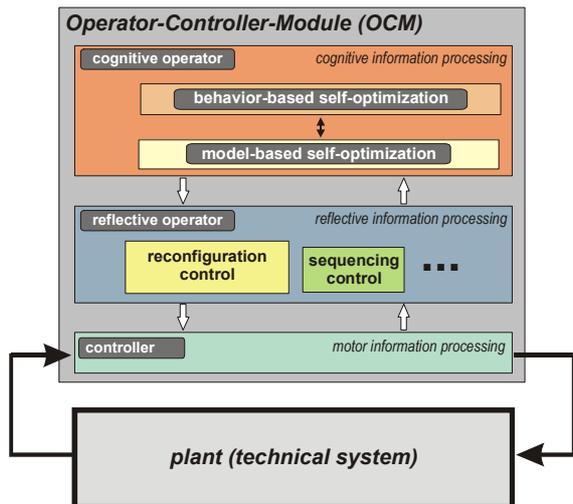


Figure 4. Scheme of the Operator-Controller-Module

On the top level we have the cognitive operator. This element is of utmost importance, since it is here, where the actual agent functions are located. The cognitive operator optimizes the system and also includes the interface to other agents. While controller and reflective operator are subject to hard real-time conditions, the cognitive operator can work asynchronously to real time. The reflective operator manages the change suggestions of the optimization in the cognitive operator and realizes them when suitable conditions (e.g. a certain controller state) arise. This ensures a save controller operation (Oberschelp et al. 2004).

ACTIVE SUSPENSION WITH TRAJECTORY TRACKING

Trajectory Tracking. Trajectory Control systems do not only serve to make a dynamic system insensitive to disturbances and to keep it within a well-defined state, but also to control a dynamic process. An example for this are machine tools: The tool must run along a given programmed trajectory despite all disturbing external forces; the controller provides a decoupling of the trajectory from the dynamics of the machine tool. In many technical applications, however, the trajectory is linked to the dynamics of the machine. Manipulators position a part at a given position. The trajectory can be chosen freely except for its endpoints, the work space of the robot and constraints like collision avoidance. In order to determine an appropriate trajectory, dynamic properties of the manipulator are considered in the optimization. With the dynamic properties, the quickest trajectory or the trajectory with the lowest energy consumption can be determined.

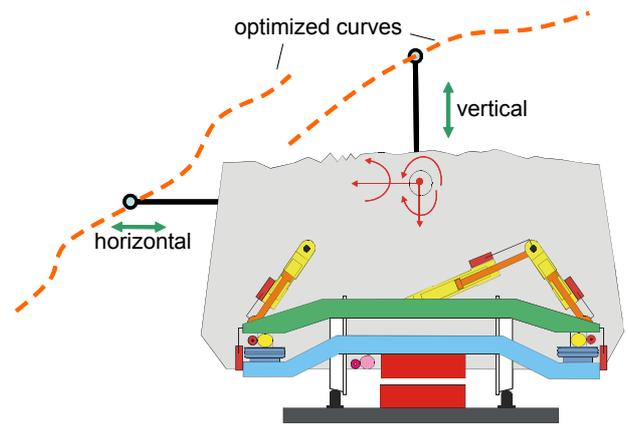


Figure 5. Trajectory tracking for the body movement

The approach can also be applied to active suspensions (see Fig. 5).

Active Suspension. Today, active suspensions can be found in upper-class cars and in some trains. In cars they are used to increase comfort while increasing safety at the same time (Streiter 1996). In trains active suspensions are before all used for tilt in order to be able to reach higher curve velocities. The NBP suspension system integrates both properties in an overall concept. The active suspension developed there allows the modification of lateral and vertical dynamics and also active tilt (RailCab 2004). It is based on air springs, which can be damped actively by base displacement of the springs. It dispenses completely with passive dampers parallel to the springs. Thus the car body is linked to the undercarriage and the train exclusively by air springs, resulting in an excellent decoupling of the car body from high-frequent excitation of drive or rails. The active control of the spring base displacement can be restricted to relatively low frequencies around the natural frequency of the car body. All in all this system can improve the ride comfort noticeably especially with higher frequencies.

The employed air spring bellows have as well horizontal as vertical stiffness so that excitations from all spatial directions can be cushioned. The active spring base displacement is done by hydraulic cylinders. Three hydraulic cylinders, arranged in a plane, move the bases of the air springs via an intermediate frame, the "suspension frame". This arrangement allows damping forces in lateral and vertical directions (see Fig. 6). In addition it is also possible to regulate the level of the car and add active tilting of the car body.

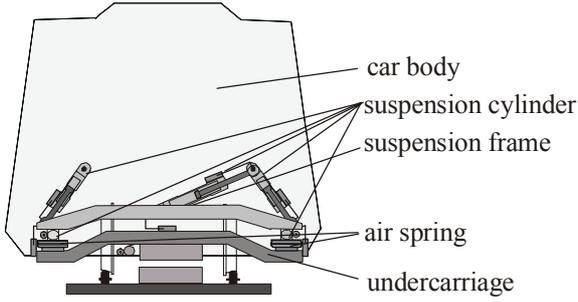


Figure 6. Active suspension system

Suspension Control. The classical control algorithm of an active suspension system uses acceleration measurements to determine the absolute velocity of the car body and displacement measurements to determine the distance between track and car body (relative position of the car body). The relative position of the car body and its first time derivative can be used to create pseudo spring and damper forces by creating additional airspring displacements proportional to the desired forces. The absolute car body velocity forms a virtual damper attached to a so called "sky hook", which suppresses the movement of the car body. For comfort reasons the sky-hook damping should be as large as possible. However, as the sky-hook damper suppresses the car body movement, a vehicle with large sky-hook-damping is not able to move up a slope or to go around a curve - after all, this requires vertical or lateral movement, respectively. If the sky-hook damper is linked to a trajectory along the optimal way across a slope or around a curve, only movement apart from the necessary movement for the desired motion would be suppressed. This technique makes it possible to have large sky-hook damping.

AGENT-BASED TRAJECTORY PLANNING

This section demonstrates how a self optimized trajectory planning can be modelled in a Multi-Agent-System.

The optimization process. The optimization is based on the measured data of the actual trajectory and the actual acceleration of the body. The aim is to maximize the comfort, which is often defined via the dying out behavior. A simple evaluation function can be obtained from the quadratic defective area against the rest position in the z -direction against the location x .

$$\varepsilon^2_{G_K} = \int_{(x=0)}^{(x=x)} \{(g_1 z_A)^2 + (g_2 \ddot{z}_A)^2\} dx \rightarrow \min \quad (1)$$

ε^2_G is the measure of comfort, z_A is the body position and \ddot{z}_A is the body acceleration. g_1 and g_2 are weighting factors.

The gain of comfort can most simply described as the reduction of the body acceleration if the relative amplitude does not have to be considered ($g_1 = 0$). What we need, for the iterative improvements, is a correlation between the trajectory and the vertical acceleration to vary the trajectory. The improvements of the trajectory can be measured through the comparison of the actual acceleration data versus the previous ones. Therefore we divide the sections, as shown in Fig. 7, in intervals (n) and calculate the quadratic deviation between the previous acceleration curve and the actual acceleration curve. The result is the measure of improvement or worsening and can be directly used for the calculation of the new interpolation points for the trajectory..

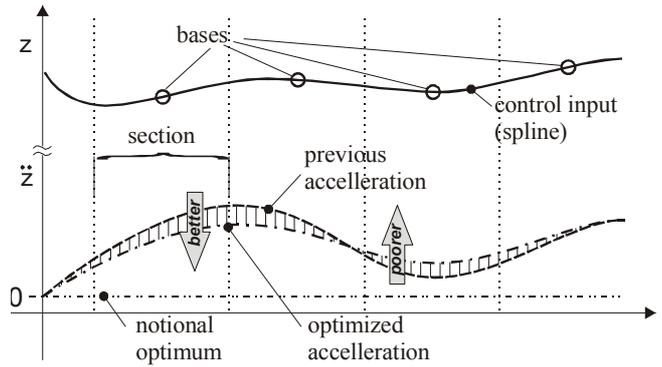


Figure 7. To optimized trajectory

The evaluation function of the variance of acceleration is:

$$\Delta \varepsilon^2_{G_K} = \int_{x=x_n}^{x=x_{n+1}} \ddot{z}_{A,P}^2 dx - \int_{x=x_n}^{x=x_{n+1}} \ddot{z}_{A,N}^2 dx \quad (2)$$

The signed result is a criterion if the last step was successful or not. A positive algebraic sign means success, a negative one failure. The amplitude of the change can be used for the calculation of the next increment. Here the step Δz is a function of the defective area and the previous increment Δz_p .

$$\Delta z_n = \text{sgn}(\varepsilon_{G_K}) \cdot q \cdot \varepsilon^2_{G_K} \cdot \Delta z_p \quad (3)$$

q is simply a weighting factor which can be varied against the number of successful steps. $\text{sgn}(\varepsilon_{G_K})$ defines the change of the direction and $\varepsilon^2_{G_K}$ the change of increment.

The search algorithm. The here used method is a modified Hill-Climbing-Search Algorithm (Russel and Norvig 2003). We don't need to start with a random trajectory but with a mathematically optimized one calculated by the track section, if there are no informations about the track characteristic available, the algorithm starts with zero-trajectory. The

shuttle follows this trajectory and sends the experienced vertical acceleration during the ride back to the track section (see Fig. 8). Based on an assessment of the actual acceleration versus the previous acceleration, the track section calculates a new trajectory for the next shuttle. The problem that the Hill-Climbing-Search Algorithm can stay in a local maximum/minimum is ignored since our only intention is to improve the offline mathematically optimized trajectory (even though we know that we could overcome this obstacle by using Simulated Annealing which combines the efficiency of Hill-Climbing with the completeness of a random walk). However we have to consider two important aspects in our scenario. The first is that our algorithm has to remember the history to decide the intensity and the direction of the search. In case it has experienced a worse state the algorithm jumps back to the last best state and adjusts its search parameters. This process is continued until a defined stop criterion is met. The second aspect is the occurrence of dependent states (interpolation points). The change of one state (interpolation point) can have significant negative and positive effects on the other ones.

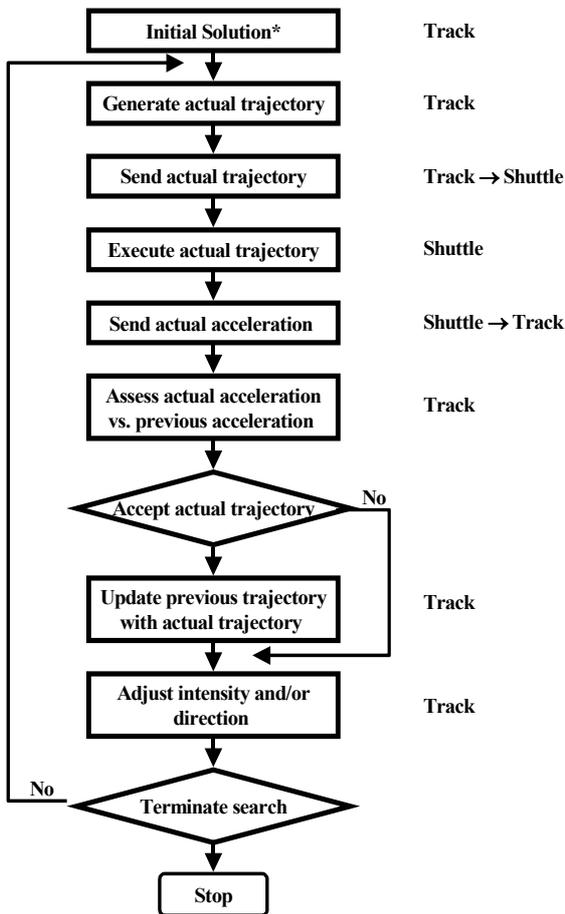


Figure 8. Structure of our algorithm

The optimization process. The trajectory parameters are the interpolation points of a spline-curve. A vehicle that approaches a track section receives the interpolation points for the default trajectory from the section administration. During the passage over the track section data about the ver-

tical acceleration of the body is measured and sent to the respective section administration.

Since for the default trajectory only continuous functions are permitted, a compact description by spline-curves is sufficient for this scenario. Here one date per interpolation point is sufficient if only equidistant interpolation points are permitted.

SIMULATION ENVIRONMENT

In the scenario described above we have to combine discrete and continuous simulation. We model the technical system by means of ordinary differential equations (ODE). The Computer-Aided Engineering (CAE) tool CAMELView (iXtronics 2003) automatically derives differential equations for the mechatronic part which is described by multibody system models .

For this purpose, nonlinear differential vector state equations are used as follows :

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}, \underline{p}, t) \quad (4)$$

$$\underline{y} = \underline{g}(\underline{x}, \underline{u}, \underline{p}, t) \quad (5)$$

with $\dot{\underline{x}}$ the state vector, \underline{y} the output vector, \underline{u} the input vector, and t the time.

We use ODE solvers which are a part of the CAE environment for the computation of the model. Since the used solver fulfills real-time requirements we are able to use the same model in offline as in online simulation (Hardware-in-the-loop).

In addition we choose the IPANEMA library as the simulation platform. IPANEMA allows the distributed real-time simulation of ODE-based models as well as finite-state machines. In addition it is possible to include external C-code (Gambuzza and Oberschelp 2003). To fulfill the requirements of an agent-based environment IPANEMA is extended by the PUB library (Paderborn University BSP-library) which allows an automatic distribution to massively parallel clusters for a realistic parallel simulation.

The PUB library simplifies the implementation of massively parallel programs according to the BSP model (bulk-synchronous parallel model). The BSP model subdivides a program sequence into so-called supersteps. At the end of a superstep a synchronisation between single parallel processes take place. During these synchronizations, messages from the previous superstep are received. For this purpose the process will be suspended. The next superstep is executed after all messages are received (Bonorden 1999).

For the simulation of mechatronic systems, modelled as time-based ODE models, the BSP model is very suitable. To

each IPANEMA model a single process is allocated. The synchronisation in time is realized by means of the synchronization mechanism of the PUB library. During a single superstep IPANEMA models are evaluated for the time interval ΔT . During synchronization messages between agents can be exchanged.

SIMULATION RESULTS

For an example, the optimization of a trajectory with track disturbance (track irregularities) is shown. Under the assumption of an ideal reference reaction of the control loop, it is sufficient to add the second derivative of the track disturbance to the acceleration of the body mass, which corresponds to the second derivative of the default curve. With the chosen distance of the spline bases, the reference reaction can be assumed to be ideal. With shorter distances a stronger curvature of the curve is possible, which requires a higher bandwidth of the actuator. The result of the addition is an incompletely corrigible disturbance of the comfort which, however, can be compensated for by the optimization.

Due to the chosen boundary conditions an analytical optimization of the curve can also be performed. In our case this is desired, since now the quality of the result can be examined more easily. In reality the disturbance is essentially unknown and the reference reaction of the body mass is not ideal.

In the following example the optimization of a section consisting of ten segments was accomplished. Fig. 9 shows the trajectory before and after optimization. The original curve displays relatively strong curvatures. By the optimization this curve can be smoothed to a high degree in approximately 65 steps:

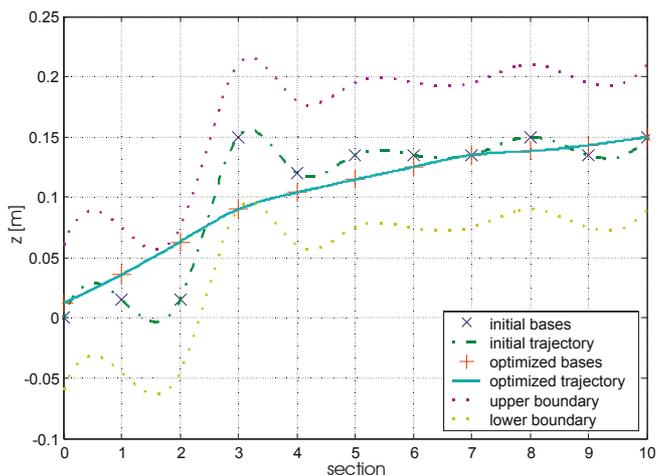


Figure 9. Initial and optimized trajectory

Fig. 10 represents the acceleration of the body mass. As can be seen, the accelerations are very strongly reduced. Since an ideal reference reaction was assumed, the acceleration of the body mass coincides precisely with the second derivative of the trajectory:

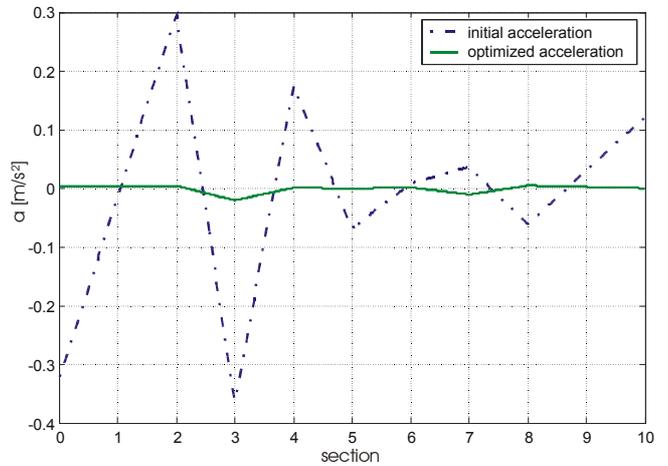


Figure 10. Initial and optimized acceleration

EXCURSUS

The active adjustable stroke of the body kinematics is approx. 12 cm; if one considers the accuracy of the track and acceleration sensors a resolution of 16 bit is more than sufficient. In the example the distance between the interpolation points is 6 metres. Therefore, at a total length of 600 metres, 100 parameters per section are needed; this corresponds to a data volume of 400 bytes per section. In our example the shuttle has a maximum speed of 200 km/h. Thus the shuttle needs 11 sec. to pass the whole length which demands a transmitting of at least 37 bytes per second. Afterwards the shuttle transmits the gauged data of the actual acceleration back to the track section. In this scenario the entire net data flow from the shuttle and back adds up to 111 bytes per second.

CONCLUSION

The paper shows how distributed learning with multi-agent systems can be used for the optimization of dynamic processes in complex technical systems. The distributed optimization is especially suitable for systems, where experience of many individual systems can be used and is thus applicable for other similar applications.

ACKNOWLEDGEMENTS

This work was developed in the course of the "Collaborative Research Center 376 - Massively Parallel Computing - Algorithms, Design Methods, Applications" - University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

REFERENCES

- Bonorden, O.; B. Juurlink; I. von Otte and I. Rieping. 1999. "The Paderborn University BSP (PUB) Library - Design, Implementation and Performance". In *Proceedings of the International Parallel Processing Symposium IPPS*, San Juan, Puerto Rico (Apr).
- Epple, U. 2000. "Agentensysteme in der Leittechnik", *Automatisierungstechnische Praxis (atp)* 42(2000) 8, Aachen.
- Ferber, J. 1999, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley.
- Föllinger, O. 1994. *Regelungstechnik*, Hüthig, Heidelberg.
- Gambuzza, A. and O. Oberschelp. 2003. "Distributed Modular Simulation of Mechatronic Systems". In *Proceedings of the European and Modelling Conference*, Naples.
- Lückel, J.; T. Hestermeyer and X. Liu-Henke. 2001. "Generalization of the Cascade Principle in View of a Structured Form of Mechatronic Systems". In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM 2001*, Como.
- Wahl, A. and E. D. Gilles. 2003. "Optimale Regelverfahren zur automatischen Bahnführung von Binnenschiffen", *Automatisierungstechnik (at)* 51(2003) 6, pp. 255-264.
- Nwana, H.; L. Lyndon and N. Jennings. 1996. "Co-ordination in software agent systems. *The British Telecom Technical Journal* 14(4), pp. 79-88.
- Oberschelp, O.; H. Hestermeyer; B. Kleinjohann and E. Kleinjohann. 2002. "Design of Self-Optimizing Agent-Based Controllers" In *Proceedings of the CFP Workshop 2002 - Agent-Based Simulation 3*, Passau.
- Oberschelp, O.; H. Hestermeyer and H. Giese. 2004. "Strukturierte Informationsverarbeitung für selbstoptimierende mechatronische Systeme" In *Proceedings of the Intelligente Mechatronische Systeme 2*, Paderborn.
- RailCab. 2004. *RailCab Project Homepage*, <http://nbp-www.upb.de/en/concept/challenge.php>, Paderborn.
- Schüssler, L. 1985. "Aktive Federungssysteme, Regelungskonzept zur Ansteuerung der aktiven Kraftstellensysteme (AKTAKON)". Technical Report Nr. 85/008, Daimler-Benz AG, R/D, Stuttgart.
- SFB376. 2004. *Collaborative Research Centre 376 Homepage*, <http://www.uni-paderborn.de/sfb376>, Paderborn.
- SFB614. 2004. *Collaborative Research Centre 614 Homepage*, <http://www.sfb614.de/eng>, Paderborn.
- Streiter, R. H. 1996. "Entwicklung und Realisierung eines analytischen Regelkonzeptes für eine aktive Federung". PhD thesis., Technical University Berlin
- Russel, S. J. and P. Norvig. 2003. *Artificial Intelligence*, Prentice Hall International.

SEMANTIC FEATURES AND FACTUAL AGENTS: A MODEL TO REPRESENT CHANGEABLE SITUATIONS

Thierry Galinho & Frédéric Serin
LIH, Computer Science Laboratory of Le Havre
University of Le Havre,
F76058 Le Havre Cedex, France

E-mail: Thierry.Galinho@univ-lehavre.fr, Frederic.Serin@univ-lehavre.fr

KEYWORDS

Factual Agent, Semantic Feature, Decision Support System, Ontology Representation.

ABSTRACT

In this paper we present a part of a Preventive Monitoring Information System. First, we describe the global architecture of the multiagent decision support system. Then, we focus on one organization of agents called *factual agents* whose aim is to represent semantic features. We also expose our ontology with its modeling. Our goal is to store and to use general and specific knowledge and information, in order to compare factual agents.

INTRODUCTION

We propose to help managers evaluate all the aspects of the current situation using data coming from different information sources like databases, managers, sensors... The goal is to build the most accurate image as possible of the significant elements of the current situation. In other words, the system must help the actors to analyse the description of the current situation in order to assist management of operational decision-making. From this viewpoint, the information system is able to inform them about the current situation and past facts related to it. This system can also be used in a situation of crisis because it provides a synthetic view of the situation and it evaluates this situation in order to anticipate its potential consequences. The goal is to improve the decision-making process of the actors by providing them, as soon as possible, with the information about what occurs and what could occur.

In this paper we present a part of the information system which we call Preventive Monitoring Information System - PMIS (Boukachour et al., 2002). We will focus on knowledge representation and the organisation of factual agents which deals with it (Boukachour et al., 2003). We shortly present the functionalities of the PMIS, its ar-

chitecture and its different components focusing on semantic features, factual agents and related ontology.

This paper breaks up into four parts: a presentation of the architecture of our multi-agent system with a detailed description of our model, a definition of the semantic features, a description of the agentification and particularly of the factual agents and, finally, a conclusion.

The structure of the factual agents uses known models and has a generic management structure for behaviour designed using an automaton. For these agents, we describe the mechanism of reinforcement and weakening of the agents which is managed at this time by the semantic features arrivals or by the interactions between agents.

MODEL & FRAMEWORK

The PMIS is based on a model using agent organisations. The agent paradigm allows to take into account the dynamic aspect produced by the evolution of the situation. The organisation of factual agents deals with the dynamic information representation process; this description includes the presentation of weighted graphs of the knowledge representation. An ontology is used to give measures to compare factual agents.

Our multiagent system (MAS) is made up various families of agents charged to recognize, to interpret, and to detect anomalies by correspondence with known or identified situations.

Global System Architecture

The system provides information about the current state of the situation. It receives facts and information coming from different sources and must: validate information, place information in relation to the situation context, evaluate the possible evolutions of the situation and give their potential consequences, dynamically increase the relevance of the situation description.

Our PMIS is made of three interfaces and a kernel based on different agent organisations (Boukachour et al., 2002) (Boukachour et al., 2003): a Human-Computer Interface for the decision-making actors, a query interface

which has access to different Information Systems. Indeed, information about the situation can be retrieved using various Distributed Information Systems. A conjectural interface which has access to a base of scenarios. The system kernel makes the connection between these three interfaces allowing them to communicate. This kernel contains the organisation of factual agents.

The situation is represented by an organisation of factual agents which is dynamic. This property allows to benefit from emergence mechanisms that lead to a dynamic configuration of the organisation representing a summary of the current state of the situation.

Graph Framework

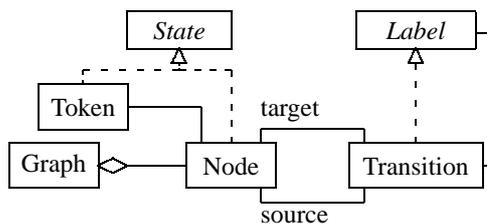


Figure 1: Graph Framework

We have created a framework called *Graph* that is made up of six classes (see figure 1). This framework is used to describe both ontologies for semantic features comparisons and automata for agent behaviours. We carry out a *labelled graph* to represent it. This graph is both itself a framework, and a part of the frameworks defining aspectual agents and ontology representation (see below).

This framework contains the classes required for the construction of labelled graphs. A graph aggregates nodes; nodes are linked by transitions. A label inputs informations into the graph coming outside from sensors or agents for example. The state notion has been created to be shared both by nodes and tokens. Token is useful to simulate the travel through a graph for example a vehicle in a road map, a token in a Petri net, several objects or agents sharing the same automaton, etc. A label defines a generic type of object for communication from a node-state to another one.

A transition has the label's signatures as behaviour, and it is able to transmit the received messages to a linked object of label type. The differentiation between transition and label is useful when informations are present in two or more transitions. A classical example is an object *way* shared by two transitions representing two possible directions.

In an ontology graph, the labels are qualifiers of the links between nodes. We have for example: links of semantic relation: "is a specialization of", "is a generalization of", "is similar to"; links of causality or dependence; links for composition or aggregation; links of action which can carry out an action.

In an agent, one automaton describes the different states that determine its behaviours. We use the token notion because the same automaton can be shared by several agents or, for the same agents, to describe different roles.

Creation of an Ontology

Agents have to communicate in a way that makes sense for them, so they must share the same language and vocabulary. This is evident according to FIPA communicative acts. However, we define our own vocabulary and semantics for the content of the semantic features. This means defining an ontology. In other words, an ontology is an explicit, partial specification of a conceptualization that is expressible as a meta-level viewpoint in a set of possible domain theories for the purpose of modular design, redesign and reuse of knowledge-intensive system components (Guarino, 1996).

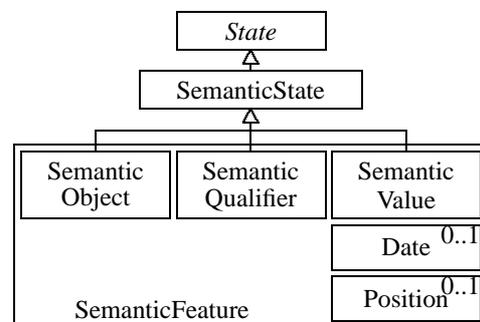


Figure 2: Categories of semantic elements

We have created different ontologies to represent the relationships between terms coming from different environments. These terms are included in the semantic features. The relationships existing among the terms are modeled with labeled transitions according to our graph framework.

SEMANTIC FEATURES

Presentation

Our system uses a model based on both a system of reception and an analysis of elementary information. These elementary informations are presented in the shape of semantic features. Each semantic feature (SF) is regarded as a fact, it is sent to the MAS which assimilates it by creating its corresponding factual agent. These agents treat these SFs (messages) and can decide whether to take into account the contained information (Wooldridge et al., 2000). The comparisons between semantic features will be done using factual agents which contain the SF. The organisation of the agents in groupings with the use of algorithms of clustering (Coma et al., 2003) will allow the comparison between the clusters of agents and scenarios.

Information translates both particular and partial aspects of an observed situation. It produces a set of SFs. Each SF represents an elementary information. The modeling of the SFs allows to obtain a homogeneous structure in the studied information system. This homogeneity is of primary importance because it allows to establish comparisons between SFs. Actually, with these comparisons, the system is able to evaluate a current situation by comparing them with referred situations (called scenarios). These situations of reference result from passed experiments, studied situations, deductions, analysis or extrapolations. We need to define the set of the observations sent to our system, that is the goal of the followed section.

Let us define a SF (semantic feature) by several properties. Each observation is an information with an index of confidence (not yet used) and a *semantic value*. A SF is an elementary information coming from the environment and the semantic value produces some semantic features.

The SF θ is an elementary information and it is part of the observation ω . It consists of five values. It integrates the event date and the location of the information. The over three informations of the SF constitute a triplet made up of an object, a qualifier of this object and a value attached to this qualifier (see figure 2). This triplet is defined by:

1. first component of a semantic feature is an object related to a certain type of object $c \in \mathcal{C}$,
2. its qualifier $q \in \mathcal{Q}$,
3. a set of possible values $\mathcal{V}_{c,q}$ which are related also to a type of object c .

Our model takes inspiration from a real case study. At the beginning of this study, no type of object was defined *a priori*. This study allows us to test our model and to define these types of objects.

The index of confidence and the origin of the information must be treated upstream (ontological treatment and determination of confidence) of the creation of the semantic feature. They will not be treated here as an indication of confidence of the observation (or rough relevance of information). The two others parameters constituting the SF are the date of the observation, and the location of the fact described by the information.

The different types of objects issuing from the study can take five identified values: *phenomenon*, *action*, *object*, *person* and *mean*. Phenomena and actions have dynamic properties, for these objects, it is necessary to associate complete temporal data: time of beginning, duration, time of observation. . . The other types of objects are regarded as descriptions of a persistent situation (at least until it is invalidated by new information). The objects, the means and the persons are called entities or persistent objects. To summary our study, we distinguish two types of objects: dynamic objects and persistent objects. Phenomena and actions are activities respectively observed (or indirectly noted) and started (or ordered). An action

is an activity with a known origin and a determined immediate goal. Phenomenon is an activity which is not an action, it has an unknown origin or it is the result of an action or another phenomenon.

We define various qualifiers and their associated values. For example:

- Qualifiers shared by the actions and the phenomena: "is-a", "state-change", "beginning-hour", "space-localization", "scale".
- Some qualifiers are specific to an action: "activity", "localization", "target-object".

The value v_{q_c} associated to the qualifier q_c can be, in some cases, identified by its type.

Persistent objects (objects, persons and means) are entities that seem persistent, objects are real entities like, for example, "vats", "valves", "vehicles". Persons are particular objects, they have an obviously significant value in the problems of management of risks and crisis where they are a goal of preventing except they are victims. They also have the characteristic to be able to have undefined behaviors. Means are sets, they join together objects and/or persons. This gathering of various entities allows specific actions. A mean may be also the way to qualify the property of particular action.

It appears, with the sight of the experiment, that each qualifier can be typed. The typing of q_c allows to define the set of the possible values v_{q_c} . This typing is significant in order to be able to establish clearly and formally rules of comparisons. The qualifiers have a representation close to the attributes' representation in a class. For example, the objects have a space localization and an identity.

The values are quantitative or qualitative, the qualifiers depend on the types of objects which they are linked up. Managing comparisons between the quantitative qualifiers is more easy than establishing a relationship between the qualitative values. However, the ontologies permit to define some proximities between qualitative values.

Proximity

The proximity of the semantic features is useful to be able to have a distance between two SFs. We aim to lay out in a formal way of one or several functions which compute the distance between two SFs or two families of SFs. Establishing distances between SF allows to reinforce or to weaken the factual agents carrying the SFs. We limit the properties of the distances to define our proximities and we adopt dissimilarities (a dissimilarity does not respect the property of the triangular inequality).

We distinguish three types of proximities: time proximity (P_t), spatial proximity (P_e), and semantic proximity (P_s). We introduce time proximity to take into account that more two events are distant, more the proximity is small. For the spatial proximity, the same reasoning is applied. We can speak about time and spatial dis-

tances. The global proximity between two SFs multiplies together these three proximities.

$$P_{SF} = P_t \times P_e \times P_s$$

Let us use Δt be the difference of time.

$$P_t = \frac{4e^{-\Delta t}}{(1 + e^{-\Delta t})^2}$$

Δe is the difference of space.

$$P_e = \frac{4e^{-\Delta e}}{(1 + e^{-\Delta e})^2}$$

These two proximities (for time and space) are a sigmoid function, it takes into account the negative values. It is written to remain on the interval $[0, 1]$. It brings the five following advantages: its continuity, its derivability, the knowledge of its primitive, its definition on \mathbb{R} entire (including negative values) and its symmetry in zero.

The definition of a semantic proximity is related to the definition of an ontology. Proximity between two semantic features θ_1 and θ_2 provides a value on $[-1, 1]$. For example, $P_s(\theta_1, \theta_2) = 0,8$ signifies that the two SFs are relatively close semantically speaking. Such measurement of proximity must relate to an ontology. This ontology graphically appears as elements in relations the ones with the others. These elements can thus be represented like nodes of a graph linked by labelled transitions. We carry out a representation using the labelled graph (see figure 1). The labels are qualifications of the links. Some of these labels can be very close such as "causes" and "can create an event". It is necessary to clearly define each field with regard to each label.

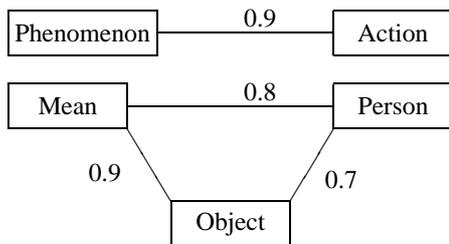


Figure 3: Graph for proximities between objects

We have created three ontological graphs. We have the graph of proximity between the five types of objects. We build this graph by taking the following values of proximity (see figure 3). The two others define proximities between qualifiers, and proximities between values.

STRUCTURE OF FA

Information is coming from different sources and we do not know if a specific datum will be important or not. So, we inject the data in the MAS to let emergence detects

some subsets of all the information (Boukachour et al., 2003).

We design factual agents for managing a semantic feature. The agent must be able:

- To represent a semantic feature.
- To compare with other SFs, i.e. to compare with other FAs in respect with the ontology.
- To try to achieve its goal (modelled by a multi-state automaton).
- To measure its own evolution and to compute its strength.

Behaviour Description

We will now describe the states and the transitions of the automaton which models the behaviour of the FA. Factual agents are both reactive and cognitive.

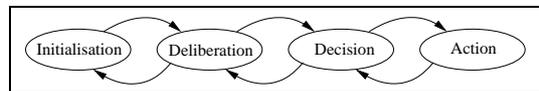


Figure 4: The inside automaton of a factual agent

As shown in figure 4, the four states of the automaton are ((Sfez, 1992)(Cardon, 1997)):

1. initialisation which is the state where an agent is created. The created agent receives messages (i.e. SF) from other agents, it updates its internal variables when it detects that some SF are close - in respect to the ontology - to its own SF. Agents stay in that state until internal variables increase enough to go over the threshold value and then change to the next state.
2. deliberation. In that state, the agent asks all the others to receive information it can use to determine if the sender is close, neutral or opposite to its SF. In that state, the agent will be more active than in its initialisation state.
3. decision is the state when the global activity of this agent rises. As in the previous states, it keeps listening to SF. But, it starts to be active by attacking one of its enemies (with opposite SF) he knows. The strategy here is to attack one of the weakest enemies.
4. action is the strongest state of the four. The goal of the agent is to try to reach that state. In that state, the other organisations of agents, as for example the clustering part, need to consider the semantic feature embedded in the agent to be of some interest to represent the current situation. If the support for that feature decreases, the agent goes back to the previous state. Here the strategy for aggression is to attack some of the strongest enemies chosen in the acquaintance network.

Going from one state to another is based on a comparison between some threshold values and the internal environment of the agent. The structure of each FA assimilates information which comes to it both in the form of SFs and in the form of interactions between FAs. These last interactions are of three types: aggression, defense and collaboration. The behaviour of each FA is given according to a state contained in an automaton.

As we explain in previous section, each semantic feature (SF) creates a factual agent (FA) in the system. The behaviour of a FA depends of its state that is a part of its knowledge. The state is a node of the automaton of the FA.

The knowledge of a FA consists of four families of fields : semantic feature, strength (force, energy and power of this agent), state (that determines the behaviour) and automaton (instance of a graph, proactiveness of this agent). The current state is a token moving in the automaton. So, implicitly, the automaton determines the behaviour of the agent.

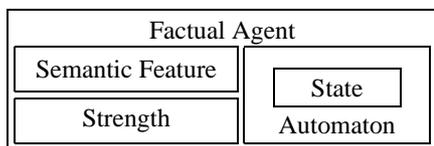


Figure 5: Fields Categories in a Factual Agent

An Agent is an atomic unit of autonomous behaviour. The core concept of Agent is independent of how the autonomous behaviour is embodied. An agent has purpose, means of receiving information from its environment, means of performing actions, mental state (knowledge and beliefs) expressed as relationships between the agents, a specification of how it responds to what it perceives in order to carry out its purpose.

Behaviour includes reasoning and decision-making processes that affect mental state, and mental state includes behaviour-related elements allowing it to perform complex, goal-directed (Caire et al., 2001). The internal behaviour of an agent can be described as a repeating three-stage cycle:

- Perception: an agent perceives its environment through Information.
- Decision: an agent decides what tasks and actions to perform on the basis of his objectives and knowledge and beliefs.
- Reaction: the agent performs the actions that it decided upon.

Scale & Thresholds

Scale represents the importance of an activity (impact on its environment) or the size of a persistent object (occupation of space or cardinality). All distances are related

to a scale. For example, if the studied problem relates to atoms, a distance of 1 m is of a gigantic size; while it is infinitesimal for an astronomical study.

For the proximity computation, the used reference value is 0 that is to say neutral action of the arriving SF. If the value is different to 0, it intervenes in the computation of the strength of the FA. Some different positive values with their signification are represented in the figure 6. The negative values mirrors the positive ones (replacing *close* by *different*).

Restricting the values in a given range allows to apply the strategy to any problem independently of the real values. The different indicated marks in the figure just help the user to fix the coefficients for the semantic distances.

The thresholds are the values used in the transitions of the FA's automaton (see above in the behaviour description).

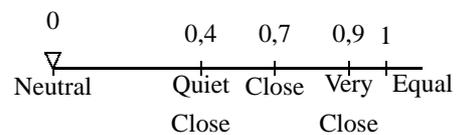


Figure 6: Scale for proximities

Strength of a FA

The reinforcement or the weakening of a factual agent is computed by the use of the interest measurement (also called P_{SF}). We use the range $[-1, 1]$ to indicate the interest of a SF for a FA viewpoint. This interest value modifies the force of the factual agent.

A factual agent (FA) has a weight m_{FA} and a speed of v_{FA} . It receives a semantic feature (SF) with the force F_{SF} . The power given by the SF to the FA is $P = F_{SF} \cdot v_{FA}$. The initial energy of the factual agent is $E_{FA} = \frac{1}{2} m v_{FA}^2$. This energy is increased with $\Delta E = T = \int P \cdot dt$ called the work of the force.

The force F_{SF} applied to the FA is function of the proximity between the SF and the encapsulated semantic feature of the FA. The new speed of the FA becomes

$$v_{FA} = \sqrt{\frac{2(E_{FA} + \Delta E)}{m}}$$

The speed, the force, the power, and the energy characterize the strength of the FA.

CONCLUSION

The introduced framework built on factual agents has been developed with an empirical validation from a real world case study. The case study we use is defined with data coming from Total's minutes book derived from a crisis training. The informations of this French petroleum company permit us to create a specific ontology, to determine objects in the ontology graph and to define the different elements of a semantic feature.

This work is part of a long term project. We are developing a prototype to test the collective behaviour of our factual agents and to correct values embedded in the ontology. One of the next steps is to connect factual agents to other organisations such as the clustering part which is developed by other members of the team (Coma et al., 2003). Clustering agents characterise the factual agent organisation in order to provide a synthetic view of the current state of the situation. They identify groups. Indeed, if a group containing factual agents in great development during the same period is identified, these agents probably contain important semantic features according to the current situation.

Among the further works we have to do, we can list: to develop automatic acquisition of new semantic features not correctly described by the ontology, to learn new scenarios to fill the base of scenarios (actually manually done). But we also have to wonder about some decision we took: is a graph the most suitable representation for the ontology, do we have to add or change the internal variables of the factual agent?

At this time we are developing a graphic system to track the behaviour of the factual agents and to improve the representation of the current situation by the organisation of the factual agents.

THIERRY GALINHO received his Ph.D. degree in Job-Shop Scheduling in 1994 from University of Rouen. Dr. Galinho is currently Associate Professor at University of Le Havre. His current application domain now concerns Agent-Based Software Engineering and Logistics.

FREDERIC SERIN received his Ph.D. degree in Object-Oriented Simulation in 1996 from University of Rouen. Dr. Serin is currently Associate Professor at University of Le Havre. His current application domain now concerns Agent-Based Software Engineering.

References

- Boukachour, H., Galinho, T., Person, P., and Serin, F. (July 2003). Towards an architecture for the representation of dynamic situations. In *Las Vegas IC-AI 2003: 659-664*.
- Boukachour, H., Simon, G., Coletta, M., Galinho, T., Person, P., and Serin, F. (July 2002). Preventive monitoring information system : a model using agent organizations. In *SCI 2002, Orlando, USA*.
- Caire, G., Coulier, W., Garijo, F. J., Gomez, J., Pavón, J., Leal, F., Chainho, P., Kearney, P. E., Stark, J., Evans, R., and Massonet, P. (2001). Agent oriented analysis using message/uml. In *AOSE2001, Montreal, Canada*, pages 119–135.
- Cardon, A. (1997). A multi-agent model for co-operative communications in crisis management system : the act of communication. In *7th European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 11–123.
- Coma, R., Simon, G., and Coletta, M. (2003). A multi-agent architecture for agents clustering. In *Agent-Based Simulation'2003, Montpellier, France*.
- Guarino, N. (1996). Understanding, building, and using ontologies. In *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada*.
- Sfez, L. (1992). *Critique de la décision*. Presse de la Fondation Nationale des Sciences Politiques.
- Wooldridge, M., Jennings, N., and Kinny, D. (2000). The gaia methodology for agent-oriented analysis and design. In *Journal of Autonomous Agents and Multi-Agent System*, volume 3, n°3, pages 285–312.

A new interaction model for agent based simulation

Flavien Balbo

Lamsade,
Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny 75775 Paris cedex 16
Inrets – Gretia
2, avenue du Général Malleret-Joinville, F-94114 Arcueil cedex.
balbo@lamsade.dauphine.fr

ABSTRACT

In this paper, we propose an interaction model which allows more realistic interactions for simulations of human societies and limits the communication cost. Interaction is a central concept when designing a multi-agent system. Classically, interaction between two agents contains two elements: communication and the action which is the result of this information exchange. This view of interaction has underpinned most research in the multi-agent domain. However, if we draw an analogy with human behavior, the notion of interaction is more complex and we show that the classical ways of interaction managing are not adapted. To reach this objective, the environment could be used to mediate interaction between agents. A toy problem shows that our proposition helps the agents to adapt their perception of communications in the light of their interest and limits the communication cost in case of complex interactions between agents. A real application stemming from the transportation domain illustrates the use of our proposition to help agents to adapt their behavior to the context.

INTRODUCTION

Interaction is one of the central concepts for the design of a multi-agent system (MAS). In the methodology called Vowel (Demazeau 1995) interaction is placed at the same level as agents, the environment and organization. That is why a multi agent based simulation has to make a choice about this component. As it is recalled in (Weiss 1999) for the multi-agent paradigm, interaction between agents contains two elements: communication and action which is the result of the information exchange. This dichotomy has underpinned most research. In the reactive agents community, communication is limited to the exchange of signals and the action result to the activation of the associated reflex. In the cognitive agents community, the communications are organized by protocols which determine the order in which the messages are exchanged. The action element is based on the analysis of high-level languages and on the analysis of the content representation language. However, this classical

view has limits. Whereas the action of a communication on the receiver is taken into account, there does not seem to be any work concerning the action of a communication on the other agents of the multi-agent system. If we define the agent context (in the broad sense) as the set of information and events of which it has knowledge, the interaction context of an agent can be defined as the constituent of its context relative to interactions. Since the impact of an interaction is limited to the protagonists and has no impact on the activity of the MAS, the classical view of interaction is a powerful limit to the interaction context of an agent. In the same way that has been identified a problem of scale when using the classical concepts of interaction in MAS for the modeling of modern human societies (Malsch and Schaeffer 1997), this paper shows that the classical view of interaction is insufficient for the modeling of highly interactional systems such as control centers (Salembier 1994). Interaction is based not only on pre-established protocols but also on the attention of the interactional activity of the participants. In multi agent based simulation and particularly in the domain of simulation of social theory (Castelfranchi et al 1992) or simulation of real human activity (Balbo 2002) (Dugdale et al. 2000), there is a need for complex interaction. The question is: what happens when agents located in the same place communicate? Indeed if the modeling of human operator is a difficult problem (Norling et al. 2000), enable them to interoperate in a realistic way is another difficulty. This paper shows that the use of classical means to organize interaction increases the communication cost. In multi-agent simulation area we propose the use of a medium (the environment) to enable the agents to participate in common conversations according to the context. For the receiving agent, the matching of the reception conditions of a message with the content improves the understanding of the interaction context.

The second section presents the issues addressed; the third section explains the modeling of Environment as Active Support of Interaction (EASI). The fourth section shows the evaluation of our proposition for a toy problem and its use for the treatment of a real problem. The last part is our conclusion and suggests future lines of research.

TOWARDS A CONTEXTUAL INTERACTION

Interactional gaps

Traditionally, the communication mode of reactive agents (non-addressed communication) is contrasted to that of cognitive agents (addressed communication). In the case of a non-addressed communication, the environment enables the agents to interact by the perception of stimuli or modifications to the environment. In this framework, the sender of the stimulus does not know the receivers of its information and it is the environment which, with physical rules such as proximity, determines the protagonists of the interaction. Each agent is potentially a participant in every interaction and its characteristics and those of its sensors determine whether or not the interaction will happen. The reception context of a stimulus is thus dependent on the application of physical rules which are beyond the control of the protagonists. Reactive agents do not exchange structured messages in compliance with protocols and have therefore mainly been used for simulations of insects on which such work was based.

Interaction by means of addressed messages allows the creation of complex protocols. For an agent, the context of an interaction is thus often dependent on the current protocol. To facilitate the matching of the sender's needs (requests) with available competences within the MAS, much research suggests specializing agents in the processing of interactions (Decker et al. 1997). But, whereas human beings working in the same environment have other sources of information (their senses) to perceive the activity of their partners, agents are unable to perceive the interactional activity of the other members of the MAS. However, it has been agreed in the definition of the agent paradigm that an agent has to perceive the world in which it evolves. If the simulation environment of an agent contains a communication part then the simulation needs a specific interaction protocol to avoid the agent isolation. The problem is that if the knowledge necessary for interactions is delocalized within specialized agents, it is not able to perceive the interactional context.

Nevertheless, an interactional act between two agents is itself a piece of information. For example, in a control center the attention of the operators is not limited to their own interactions but also to those of the others. The concepts of floating perception (Salembier and Zouinar 1998) or mutual awareness (Dugdale et al. 2000) illustrate this need and are used to express the operator's way of managing their attention to events. These events can take various forms: a message which is easily usable in MAS, an agent signaling interest for another agent. In this case, information does not concern the message itself but the perception of the exchange of messages between two agents at a particular moment. Consequently, the interest of the agents is guided by the interactional context in which they function.

Using the environment as medium

It is clear that the use of the environment by the reactive agents in order to interact does not allow an exchange of complex messages. Similarly, the use of addressed messages often supposes a more or less distributed management of the knowledge necessary for the interaction. These two interaction modes have gaps but their combined use opens a third way. This third way has to enable cognitive agents to design their interactional context. In other words, the aim is to allow an agent to specify the conditions in which it wishes to perceive interactional events. To reach this objective, cognitive agents should use the environment as a support for their communications.

The principle of an environment common to the agents is central in the reactive agents community. In this paradigm, the agents have sensors so as to perceive the environment, and effectors so as to act on it. If extended to cognitive agents, the common environment would be an interaction medium that each agent could modify by its own interventions (i.e. sending of messages) and perceive by means of sensors. An MAS is no longer designed as a sum of communications which are organized with protocols, but as an interaction medium where each message can be perceived independently of the initial needs of its sender, its receiver and their current protocol.

It must be possible to exchange complex messages (cognitive agents) and the agents must be able to perceive in the environment the messages they should receive (reactive agents). The exchanged messages can be addressed or not. The exchange of addressed messages supposes the use of pre-defined protocols. The use of the environment conditions the reception of a message to the perception capacities of the agents. In this way, each agent defines its own interactional context.

The aim of the present research is to propose a model extending the use of interactions between agents. The problem is to bind an interactional need of an agent not only to its own state but also to the state of the world. The introduction of the notion of context in the interaction protocols allows the agents to specify the conditions in which they wish to be concerned by an interaction and by extension to reconstitute more easily the context in which they are receivers of a message.

ENVIRONMENT AS ACTIVE SUPPORT OF INTERACTION: EASI

The reactive agents community has produced a modeling of the agents from their perception of the environment to their reaction. This could be extended by modeling the perception filters of the environment used by cognitive agents.

The example below illustrates the various components of the proposal. It is followed by a presentation of the

EASI model, which is a generalization of our initial work (Balbo 2002). A use of our model is presented in the last section.

Example

To illustrate our model, let us take the following toy example: simulation of interactions within a classroom. The teacher and the individual students are each represented by a different agent. The classroom constitutes the common environment where interactions (exchanges of messages) between cognitive agents are perceptible to all. The figure below contains possible interactions.

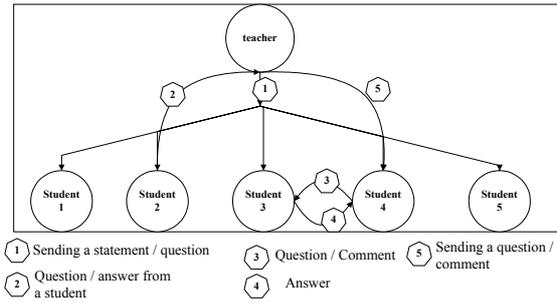


Figure 1. Representation of interactions in a classroom

The teacher gives information to the students who can ask questions (questions which the teacher answers). The students can also communicate between themselves to ask questions or to make comments. In the interactional context-building scenario, an agent chooses to perceive certain events according to its expectations. For example, a student wishing to go out of the room waits for another student to ask the teacher a question before carrying out its action. This agent is therefore interested in the interactional context of the MAS and not in the content of the exchanged message.

EASI model

In the following section, we give the minimal definitions which are necessary to the understanding of our proposition.

For the reactive agents community, a tropic agent (Ferber 1995) is defined as a tuple:

$$a = \langle P_a, \text{Percept}_a, \text{Reflex}_a \rangle$$

Where:

- P_a : set of percepts associated to an agent
- Percept_a : perception function which associates a percept to every state of the world
- Reflex_a : function which associates an action to a percept.

By extension, a cognitive agent perceiving communications in the environment is defined as:

- P_a : messages deposited in the environment and accessible to agent a . In theory every message in the environment is accessible to the agents present.

- Percept_a : set of perception filters enabling agent a to receive messages. It is the filters allowing the perception of communications which define the subset of messages perceptible to an agent.

- Reflex_a : activation of the reasoning process according to the message received (content and filter). How the agents consider the message will depend on its content and also on the filter that enabled its reception.

The environment is modeled using filters enabling agents to receive messages. The elements present in the environment will be noted:

$$\Omega = \langle \Omega_A, \Omega_M \rangle$$

Where:

- Ω_A : the agents of the MAS
- Ω_M : the messages of the MAS

Each element in the environment is recognizable by a set of properties which are accessible by the environment.

Definition 1: an entity

An entity is any element of the environment that is made up of properties:

$$\omega \in \Omega, \omega = \{p(\omega) \neq \text{null} \mid p \in P(\Omega, D_\Omega)\}$$

Where

$P(\Omega, D_\Omega)$: set of the properties of the MAS

$D_\Omega = \bigcup_p D_p$: union of the definition sets

Definition 2: an agent

Let A be an agent category and a an agent of A , then

$$\forall a \in A, a = \{p(a) \mid p \in P(A, D_\Omega)\}$$

An agent category is a set of common properties. An agent is defined as a member of a category and differs by the values of its properties. An agent has other components (knowledge or skill), but they are not useful for our interaction model.

Example: There is an unique agent category, that we called *person*. For each *person* we have three properties, the position in the classroom (*Position*), a unique identifier (*Identifier*) and the agent's role in the simulation (*Role*). The teacher agent (noted T) is defined with the tuple:

$$T: \{0, 1, \text{teacher}\}$$

Where:

- Position (T) = 0
- Identifier (T) = 1
- Role (T) = teacher

A student, for example student 3 (noted $S3$), is identified with the tuple:

$$\text{Student3}: \{3, 3, \text{student}\}$$

With:

- Position ($S3$) = 3
- Identifier ($S3$) = 3
- Role ($S3$) = student

Definition 3: a message

$$\forall m_o \in \Omega_M,$$

$$m_o = \langle \text{Sender, Receiver, Subject, } \bigcap_{l=1}^n C_{ijk}^l \rangle$$

Where:

- Sender: identifier of the sender.
- Receiver: identifier of the receiver.
- Subject: subject of the message
- $C_{ijk} = \langle p_i, f_j, v_k \rangle$
- $p_i \in P(\Omega_A, D_\Omega)$
- $f_j: D_{p_i} \times D_{p_i} \rightarrow D_{\text{Bool}}$
- $v_k \in D_{p_i}$
- D_{p_i} : domain of definition of property p_i .
- $P(\Omega_A, D_\Omega)$: set of properties of MAS.
- $\bigcap_{l=1}^n C_{ijk}^l$, The conditions under which the receiving agent has to meet to receive the message

A condition $C_{ijk}(\omega)$ is true if :

$$\omega \in \Omega \wedge f_j(p_i(\omega), v_k)$$

The definition of a message corresponds to the definition of the properties and gives us the semantics of the messages. This semantics is common to all the agents. By construction, the description of a condition becomes a property of a message.

Remark: the content of the message is not a parameter of our semantics.

Example: let M_1 be the message:

$$M_1 = \langle 1, \text{unknown}, \text{"question"}, \langle \text{role}, =, \text{student} \rangle \rangle$$

M_1 is a message sent by the teacher agent (identifier = 1) to an *unknown* receiver (the sender does not know its identifier) but with a property *role* value which is equal to student. That corresponds to the broadcast of a question (subject value) from the teacher agent to the student agents.

Definition 4: *a filter*

Let F_n be the nth filter of the environment.

$$m \in \Omega_M, a \in \Omega_A,$$

$$F_n(m, a) = \left(\bigcap_{l=1}^{l_a} C_{ijk}^l(a) \right) \cap \left(\bigcap_{l=1}^{l_m} C_{ijk}^l(m) \right) \cap \left(\bigcap_{l=1}^{l_e} C_{ijk}^l \right)$$

Where:

- $\left(\bigcap_{l=1}^{l_a} C_{ijk}^l(a) \right)$: Conditions concerning the receiving agent of the message.
- $\left(\bigcap_{l=1}^{l_m} C_{ijk}^l(m) \right)$: Conditions concerning the message processed.
- $\left(\bigcap_{l=1}^{l_e} C_{ijk}^l(e_l) \right)$: Conditions concerning the other entities identifying the interaction.

l_a, l_m, l_e : number of conditions for each entity category.

A filter links a message m to an agent a . An agent designs a filter with conditions relative to other agents and to messages e_l , thus allowing many possibilities in the management of its interactions. For example, an agent A can design a filter allowing it to receive the messages of agent B and conditioning its reception to the following interactional event: agent C sent a message to agent D. Consequently, it is not possible to represent the parameters of a filter without using sets of conditions.

Each filter enables an agent to receive messages according to conditions concerning some of the entities present in the environment. This set of conditions constitutes the interactional context in which the agent wishes to be contacted.

Example: F_1 is the filter allowing agent S3 to receive question sent by a student (identified by the variable a_l) to the teacher. The receiver will use the message for its content or as the event identifying an interactional context.

$$F_1(m, a) = \langle \langle \text{sender}(m), =, b \rangle, \langle \text{subject}(m), =, \text{"question"} \rangle, \langle \text{receiver}(m), =, 1 \rangle, \langle \text{Identifier}(a), =, 3 \rangle, \langle \text{Identifier}(b), =, i_b \rangle, \langle \text{role}(b), =, \text{student} \rangle \rangle$$

Having studied the first definitions necessary for the establishment of our model, let us now examine our example.

INTERACTION PROTOCOL COMPARAISON

Each student has a parameter called *attention*. This parameter determines the basis of the student behavior. It could take three values:

- **#0:** the student is inattentive; nothing that happens in the classroom has interest for him. Each received messages will not be treated except the teacher remarks.
- **#1:** the student listen but does not participate. The received messages are taken into account but he does not answer questions.
- **#2:** the student participates. He answers each question.

This basis attitude of a student evolves during the simulation. Each agent follows this rule: the higher is attitude, the less stable its behavior. This parameter is used to modify the interactivity level of our simulation.

In order to evaluate our proposition, we have compared it with standards of the multi-agent paradigm:

- a) **Broadcast:** when the receiver of information is unknown, the receiver broadcasts the message to all agents.
- b) **A middle-agent:** a specialized agent is the compulsory intermediary between agents. It receives and forwards each message only to agent having their

attention parameter value superior to #0. Because this parameter evolves during simulation, the agents must inform the middle-agent in case of modification of their state.

c) **Mixed**: the use of a specialized agent is required only when the number of messages may be decreased.

d) **EASI**: The value of the *attention* parameter is accessible by environment. Each agent has its own communication filter.

Scenario

In this scenario, the teacher sends questions to students and remarks to inattentive student. A student (S3 in the scenario) wants to use this event to become inattentive if its neighbor is itself attentive. This scenario supposes that S3 adapts its behavior according to two different information sources. Moreover, the aim of S3 must remain secret and that suppose that there is no particular protocol to optimize the communication cost.

To remain this intention private, each agent has to know the value of the parameter *attention* of the other. This hypothesis is necessary to simulate a student who observes the behavior of the other in the classroom. That is why the interaction protocol has to deal with the value updating of the *attention* parameter (Figure 2).

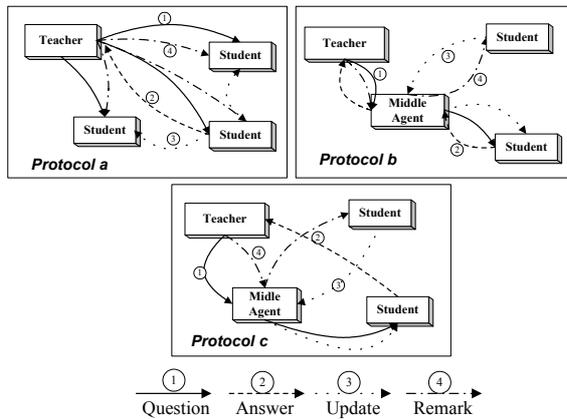


Figure 2: Interaction Protocol.

Since the two information sources are independent, only agent is able to evaluate them according to its interest. The consequence is that during the simulation each information has to be sent to each agent. That is why a mechanical increase of the simulation time (figure 3) and of useless messages (figure 4) is observed.

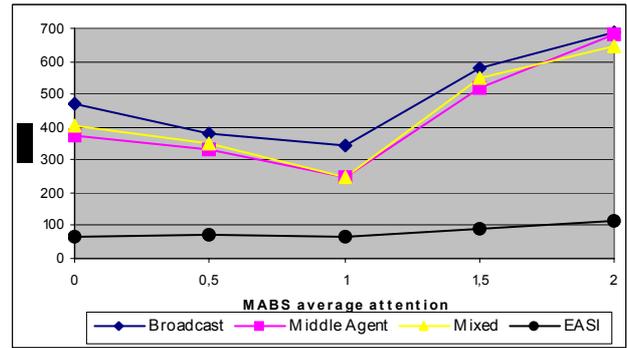


Figure 3: time evaluation

At each message a process time is associated. Because of the behavior rule and the design of the simulation, the value of the average attention determines the number of exchanged messages (update and answer message).

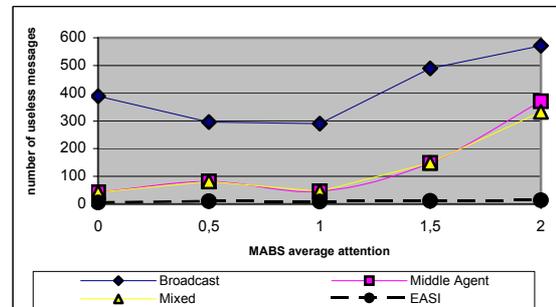


Figure 4: useless messages

For a student, a message is useless if its attention is null. For MAS, a message is useless if it is due to the protocol and not to its activity. The update messages are useless because they are the consequence of a protocol (the interaction knowledge is delocalized).

APPLICATION

The first domain in which, the proposal was applied is that of transport. Our application, called SATIR (Balbo 2002), illustrates in a very dynamic situation (regulation of an urban transport system) how an event-based interest in interactions was used to manage the inconsistency in the location of vehicles and how the reception context of a message enabled the agents to adapt the way they manages the inconsistencies.

Urban public transportation systems are naturally open systems (vehicles appear in or disappear from the network according to their activity) and distributed systems (vehicles move on a network).The multi-agent paradigm makes it possible to model and simulate those systems where the distribution of control and knowledge facilitates problem solving. Therefore, a multi-agent approach was chosen to model the system in order to 1) diagnose disturbances in the bus lines (bus delays, bus advances), 2) detect inconsistency in

positioning data sent by buses to the central regulator, 3) dynamically compute schedules, 4) monitor and process disturbances 5) simulate and choose feasible solutions. This research was part of the SATIR project done with the participation of the French Transportation Research Institute (INRETS).

Only the part related to the dynamic timetable management and management of data inconsistencies will be presented.

Timetable management involves three steps: 1) making up the theoretical timetables; 2) monitoring the network activity (modifying the timetables according to where the vehicles actually are); 3) managing the inconsistencies of the data from the sensors which locate the vehicles.

To automate these three functions, we propose two categories of agents:

1. The STOP agents, which represent the theoretical structure of the network and compute the theoretical timetables.
2. The BUS agents, which represent the dynamic part of the network. Every BUS agent is the *abstract model* of an actual vehicle running on the transportation network and reports its movements to the STOP agents.

When a vehicle passes a stop on the actual network, a warning message is sent from the BUS agent to the STOP agent concerned. The STOP agent updates its timetable by removing this vehicle from the list of vehicles due. A STOP agent which does not receive any message detects an anomaly and triggers the disturbance processing.

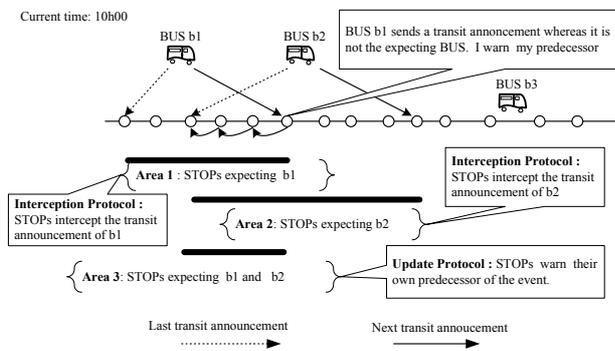


Figure 5: Inconsistencies management

One of the difficulties of timetable management concerns the management of inconsistencies which arise from the data sent by the location sensors located in built-up areas (Figure 5). Some vehicles may not be located at a significant number of stops and this may result in the triggering of false alarms. The incorrect location of a vehicle may lead to inconsistent situations with "virtual overtakings" (a vehicle is announced before the vehicle which precedes it). From the point of view of the STOP agent, there are two different problems. In the first case, there is an information defect

and in the second case, there is a problem in the reception of the information. Each STOP agent concerned must be informed as soon as the vehicle is located again (in the first case, the BUS agent sends a message to a STOP agent). In the second case, it has to recognize an inconsistent situation.

The STOP agents have a filter which intercepts a warning message concerning a transit announcement that has been sent by the BUS agent whose interceptor is waiting to a STOP agent whose position is higher than that of the intercepting STOP agent (Figure 5: Interception Protocol).

The STOP interceptor thus receives a message with a reception condition concerning the reference of another STOP and it can reconstitute the context of the reception of the message. Using the context, the STOP agent is able to deduce that the vehicle was wrongly located. In this case, the content of the message enables the STOP agent to update its state and the interactional information enables the agent to modify its role in the processing of a disturbance. In this case, it is the filter which, by its function (interception of location message), determines the processing context of the message.

In the case of "virtual overtaking", a STOP agent receives via the reception filter addressed messages (according to the identifier of the receiver) a transit announcement which the sender has sent to it. It is the comparison of the reference of the sender to the reference of the BUS agent it is waiting for that will modify its behavior (Figure 5: update protocol). In addition to taking this information into account for the update of its state, the contacted STOP agent sends a warning message to the STOP agent which precedes it in order to warn it of this event. If this preceding agent is also expecting this BUS (which is early and not correctly located), it updates its timetable and forwards the message to its own predecessor. If the STOP agent is not expecting this BUS, it will not forward the message. In this case, it is the content of the message (the reference of the sender) which allows the receiver to reconstitute the context of the interaction and to adapt its behavior.

For the same message, a transit announcement, we thus obtain an identical basic processing (the STOP agent updates its timetable) but a different reaction depending on the context of its reception:

- Message sent to the receiver by the expected sender: the local state of the network is normal.
- Message sent to the receiver by a different BUS agent: the vehicle has been "virtually" or "actually overtaken", the situation is disturbed.
- Message sent to a different receiver by the expected sender: the vehicle is wrongly located, the situation is possible "disturbed".

In summary, it is the interactional reception context of the same message which will vary the agents answer.

CONCLUSION

This paper has presented a proposal to use the environment as an interactional medium. This work represents an abstraction of our first proposal called ESAC *Environment comme Support Actif de Communication*. In the first version the problem was to define a “communication logic” enabling each agent to find its interlocutor according to the characteristics it was searching for in this agent. The new EASI proposal includes this problem and we have shown, with a toy example, how the model enables to limit the interaction cost in case of complex interaction. The proposal extends the classical view of interaction (communication organized with protocols) by allowing each agent to bind its interactional needs to the context. In MABS, our proposition enables agents to be aware of interaction with a minimal cost. This work will be extended by modulating the interest of the interaction. The filter matches the reception condition to the context. Nevertheless some contexts (or messages) may be more important than others, which is why it may be useful for agents to modulate their interest in interaction.

BIBLIOGRAPHY

- Balbo F.; Scemama G. and Tendjaoui M. 2002. “SATIR: A Multi-Agent Decision Support System for Bus Networks”. In *Proceedings ECAI Workshop on Agent Technology in Logistics*, pp 49-58, Lyon July 2002.
- Balbo F. and Pinson S. 2001. "Towards a Multi-Agent Modeling Approach for Urban Public Transportation Systems", in Omicini A., Petta P., Tolksdorf R. (Eds) *Engineering Societies in the Agents Worlds II*, LNAI 2203, Springer Verlag.
- Castelfranchi C.; Miceli M. and Cesta 1992. “A. Dependence relations among autonomous agents”. In *Proceedings of MAAMAW92*, Elsevier Science, pp. 215-227.
- Decker K.; Sycara K. and M. Williamson 1997. “Middle-Agent for the Internet”. In *Fifteenth International Joint Conference on Artificial Intelligence*, 578-583, Morgane Kaufmann.
- Demazeau Y. 1995. “From interactions to collective behaviour in agent based-systems”. In *First Conference on Cognitive Science*, Saint-Malo, France.
- Dugdale J.; Pavard B. and Soubie J.L. 2000. “A pragmatic development of a computer simulation of an emergency call center”. In *Designing Cooperative Systems. Frontiers in Artificial Intelligence and Applications*, Rose Dieng et al., IOS pres.
- Ferber J. 1995. *Reactive distributed artificial intelligence: principles and applications*. G. O’Hare, N. Jennings (Eds), Foundation of Distributed Artificial Intelligence, Wiley, New York.
- Malsch T. and Schaeffer S. 1997. “Generalized Media of Interaction and Inter-Agent Coordination”. In *socially intelligent Agents, AAAI Fall Symposium*, technical report FS-97-02, Menlo Park, CA, AAAI Press.
- Norling E.; Sonenberg L. and Rönnquist R. 2000. “Enhancing Multi-Agent Based Simulation with Human like Decision Making Strategies”. In *Proceeding Multi-agent based simulation*, pp 214-228.
- Salembier P. 1994. "Assistance coopérative aux activités complexes: l'exemple de la régulation du trafic aérien ». In *Systèmes Coopératifs: de la modélisation à la conception*, In B. Pavard (ed.), Octares, Toulouse.
- Salembier P. and Zouinar M. 1998. “A model of shared context for the analysis of cooperative activities”. In *Workshop on Mutual Knowledge and Public Information*, European Summer School on Language, Logic and Information (ESSLLI’98), Saarebrucken, August 24-28.
- Weiss G. 1999. *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*. MIT Press.

AUTHOR BIOGRAPHIES

FLAVIEN BALBO is Assistant Professor in Computer Science at University Paris 9 – Dauphine. He received a M.S. in Applied Mathematical Science and a Ph. D. in Computer Science from Paris 9 University. He is associate researcher at INRETS (National Institute of Transport and Security). His primary research focuses on Real Time Decision System. His other research interests include multi-agent system and specially interaction within multi-agent systems.

AN INSTRUMENTALIZED PARTICIPATORY APPROACH FOR COOPERATIVE KNOWLEDGE ACQUISITION TO BUILD A SOCIAL MABS

Athmane HAMEL^{1,2}

Jean-Marie ATTONATY¹

Suzanne PINSON²

¹ INRA

UMR Economie Publique, BP01

78850 Thiverval Grignon

hamel,attonaty@grignon.inra.fr

² LAMSADE, Université Paris-Dauphine

Place du Maréchal de Lattre de Tassigny

75775 PARIS Cedex 16

pinson@lamsade.dauphine.fr

Keywords

Social MABS, meeting, cooperative work, knowledge acquisition, distributed decision making, poultry chain.

ABSTRACT

This paper proposes a participatory and cooperative approach to acquire necessary knowledge to build a MultiAgent-Based Simulation (MABS). This approach is based on role-playing, meeting and Computer Supported Cooperative Work (CSCW) principles. During a meeting firm actors simulate problem solving processes by playing their usual roles. The result of several meetings is a corpus which is analyzed to provide a set of scenarios. It allows the building of a multiactor model.

In this study, we apply the approach to acquire knowledge to model and then simulate decision-making processes in poultry firms. In particular, we seek to understand the probable impacts of individual behaviors in the decision process of managing raw material.

INTRODUCTION

MultiAgent-Based Simulation (MABS) is an efficient way to understand multiactor systems, through the Multi-Agent System (MAS) modelling processes and the analysis of the simulation behaviors when run (Edmonds, 2001). This is due to the MABS's ability to cope with simple entities as well "groups" and "organizations" (A.Drogoul, Vanbergue and Meurisse, 2002; Edmonds, 2001), and interaction between entities and groups.

MAS methodologies (Burmeister, 1996; Fishwick, 1997; Wooldridge, Jennings and Kinny, 2000) and methodological proposals for computer simulation (Fishwick, 1996; Gilbert and Troitzsch, 1999) define processes and models to represent computational agents, their interactions, organizations and environments. An important step in MABS building is to identify the agents to be introduced into the model. Unfortunately, most methodologi-

cal proposals (Ferber and Gutknecht, 1998; Iglesias, Garajo, Gonzalez and Velasco, 1996; Parunak, Sauter and Clark, 1997) underestimate the difficulties faced when building computational agents. They consider that agent identification is a straightforward operation during two modelling phases: analysis and design phases. Our goal, is to propose a method for "real agents" identification and knowledge acquisition to build socio-economical simulation models.

Document Analysis or Knowledge Acquisition?

Generally, the modelling process of a MABS is based on the knowledge provided by descriptive documents. Thanks to those documents, the agents can be identified through a linguistic analysis (Parunak et al., 1997).

Multiactor systems, particularly within a non-delimited organization (see definition in Section) are often characterized by the difficulty to get documents describing the organizational functioning of such organization. In this case, a Knowledge Acquisition (KA) step is necessary to build agent-based simulation. During a KA phase, domain experts (also called *thematicians* (A.Drogoul et al., 2002)) are called up.

Actually, there are two main reasons to introduce the target system actors to the modelling process during KA process. First, those actors hold the necessary domain knowledge. Second, as shown in (A.Drogoul et al., 2002; Bousquet, Barreteau, d'Aquino, Etienne, Boissau, Aubert, Page, Babin and Castella, 2002), most of thematicians enter the design of the multiagent simulations because they are interested in understanding the global target system. Indeed, in situations where they cannot explain their contributions and the consequences of their decisions, neither deductively or analytically, the knowledge acquisition phase is necessary to explicit such knowledge.

An agent-based simulation encompassing individual and collective knowledge provides a useful tool to evaluate the consequences of individual decisions on the global target system. Our case study is understanding the im-

pacts of individual behaviors and knowledge in the decision processes of managing raw material in a poultry chain (see Section).

A new methodological framework

To deal with these two issues, namely the lack of descriptive documents and the need to identify the pertinent agents of an organization, we propose a new methodological framework. Our approach is to be used during the analysis step of modelling process. It provides a domain model called *multiactor model*, to be used by the modellers during the design phase (see Figure 1). The multiactor model is used to build defined agent models in MAS methodologies.

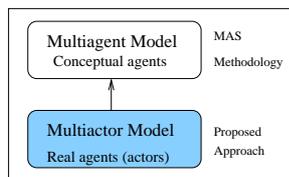


Figure 1: *The multiactor model position in MABS design process*

The main objective of the approach is to gain some understanding of the organizational patterns and decisional processes of firms, on the basis of initial model objectives. In fact, a knowledge acquisition process is essential for the building of a simulation model.

In order to collect and acquire the necessary individual and collective knowledge to build a MABS, our approach uses the meeting and cooperative work principles. It is inspired from the Computer Supported Cooperative Work -CSCW- techniques. This method is also based on role-playing during meetings. It involves several roles: thematians, expert thematians and modellers. (see section).

Concepts

A “role” is what the actor is expected to do (Yu and Mylopoulos, 1994), in accordance to his commitments and organizational rules. Moreover, an actor cooperate with other actors in accordance to their commitments. He also exchanges knowledge and coordinates his activities.

A “non-delimited organization” represents an unbounded organization, without a legal status. For example, a wheat chain or a national book market.

“Target system” characterizes the organization to be modelled and various decision processes within it. Decision-making processes are seen as a series of interactions between stakeholders (Bousquet et al., 2002).

In the next section, we present our own approach, which leads us to present, in section , a real case study. we propose an example of decision-making processes in a poultry chain. In the section , we discuss some related works

to our research area. We finish with some concluding remarks and further works.

PROPOSED APPROACH

This approach aims at: (i) understanding the structure and functioning rules of the target system through the building of MABS model; (ii) dealing with “real agents” identification (iii) and leading to a domain model, called multiactor model, to be used to design MABS.

In this approach, we postulate that we can understand multiactor systems when “real agents” are interacting. We assume that actors behave more instantaneously when reacting to some events in their environment (e.g. information requests, activity requests, etc.).

Involved roles to apply the approach

The notion of role in a modelling process has already been introduced in several works (Bousquet et al., 2002; A.Drogoul et al., 2002; Edmonds, 2001). Our approach is based on the attendance of several actors, with various roles: expert thematians, thematians, and modellers. Modellers are in charge of building the simulation models (Fishwick, 1996). The expert thematians and thematians are domain experts. The differences between the two roles are the knowledge handled, their MABS interest and objectives. Expert thematians are domain researchers, policy makers, etc. who handle general knowledge and observations of the target system. The thematians are professional experts (e.g. firm managers). They have a specialized point of view.

Simplified view of the approach

Figure 2 is a scheme of our approach. The analysis process of a non-delimited organization follows several steps:

- The process starts with an abstraction step, which depends of the model objectives. The abstraction is built after several individual expert interviews have been carried out.
- A knowledge acquisition process is necessary to understand organizational functioning and collect domain knowledge. It can be a *collective* or an *individual* knowledge acquisition. A *corpus document* results from this step.
- The analysis of the corpus provides a more precise view of handled knowledge (micro and macro knowledge), relationships between actors, interactions, etc.
- finally, a *multiactor model* is built.

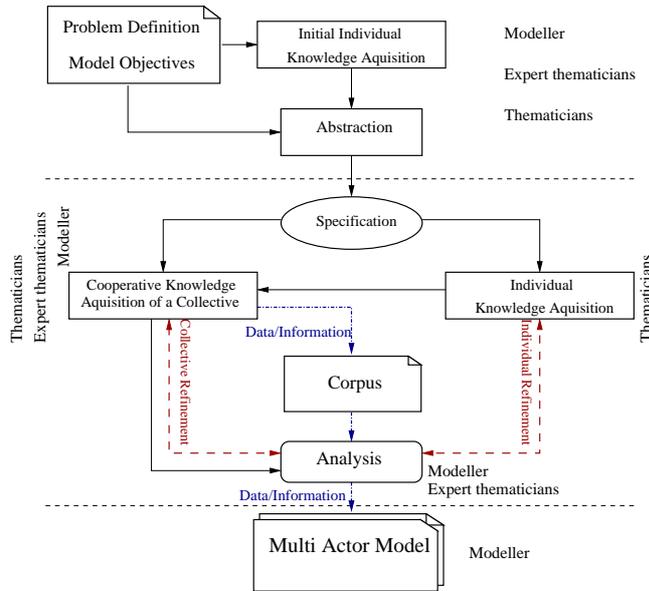


Figure 2: Simplified view of the proposed approach

Abstraction

An abstraction gives a simplified structural view of the target system. In this stage, the expert thematician and thematician roles are involved. Each expert has a subjective point of view (Edmonds, 2001) of the target system and the model. Thus, the refinement degree of our model is to be chosen.

$$\Phi = \langle F, E \rangle \quad (1)$$

where F = set of identified groups (subsystems), and E = Evolution environment of F

Example (see Section) A poultry chain consists in a set of competing firms. Each firm has its own organizational structure, strategies, etc. A firm is then considered as a group within the poultry chain.

Collective knowledge acquisition

Collective knowledge acquisition consists of role-playing meetings. A meeting (Figure 3) is characterized by a set of common objectives and a set of participant actors (thematicians). A meeting objective can be an initial model objective or be defined by the participants themselves. The thematicians interact to solve proposed objectives. The knowledge acquisition runs using:

- *a communication framework*. It is an information dialogue framework, composed of laptop computer network that can be moved to any meeting room. Each computer contains a generic application, with a very simple interface, and communicates with other applications via CORBA (Common Object Request Broker Adapter) layer. This framework also contains a *spy program* that

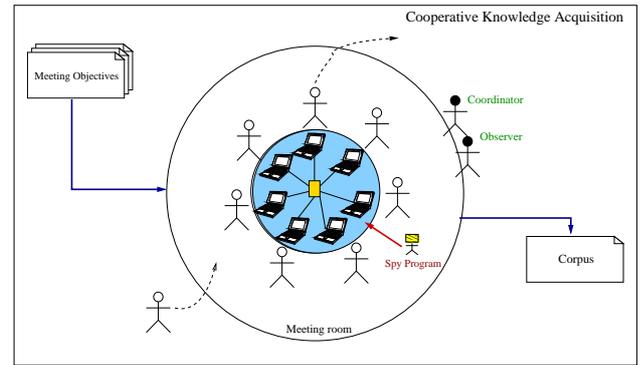


Figure 3: Cooperative Knowledge Acquisition of a Group

collects all exchanges. The stored data is called *corpus documents*. For each exchanged message, the spy program associates a sequencing identifier (including serial number within a conversation). It also stores the sender and receiver of each message, as well as the message's content.

- *meeting principles (face-to-face)*. Each meeting participant uses the application installed on his own computer. He is able to: (1) play his usual role in the firm (2) describe this role (3) communicate with actors he chooses (4) use, or define some keywords or domain concepts. Communications are done with natural language.
- *Observer and coordinator participation*. Meetings are coordinated by the modeller (called *coordinator* in this case). An expert thematician (meeting *observer*) observes the meeting development.

Important Remarks (1) For each meeting, functionalities can be used or referenced by actors, such as databases, computational resources (like linear programming -LP- optimization); (2) If an unavailable role seems to be relevant and is called up by a meeting participant, then a new role player is invited.

Analysis step

After several meetings, the corpus documents are analyzed (Figure 4) : (1) by conversation representation (UML diagram sequencing, state machines, Dooley graphs) -see Section -; (2) by role analysis, which allows to discover activities and decisions (see example below); (3) by role dependencies (Yu and Mylopoulos, 1994) analysis, which allows to discover goal and resource dependencies.

Example

- Role = *Purchaser*: carry out a new raw material purchase;
- Decision = *Buy (Yes/No)?*;

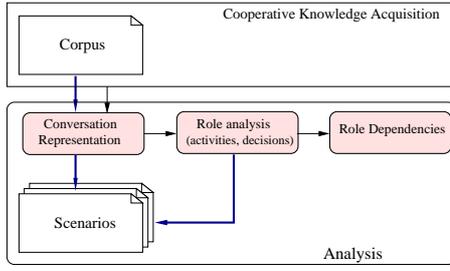


Figure 4: Analysis Step

- Activities = *Request critical information* from other actors (potential use?, global profit?), *provide requested data*.

Notion of scenario

For each discussed objective during a meeting, a *scenario* is built as follows (detailed example in section):

Scenario	Scenario name
Objective	Raw Material Purchase
Participant roles	Purchaser, etc.
Interactions	Exchange contents, conversation models
Decisions	Buy (yes/no?)
Activities/Role	Compute profit

A scenario S is a 4-tuple

$$S = \langle \langle SO \rangle, IS, CR, R \rangle; \quad (2)$$

In a scenario, a role is defined as an abstract actor. Let n be the number of identified roles.

SO =“Scenario Objective”, IS =Interaction sequences, CR =Conversation Representation, $R = \{R_i\}_{i=1..n}$ a set of pertinent roles, and

$$R_i = \langle \langle RoleName \rangle, A_i, K_i \rangle;$$

A_i =Acquaintances of R_i , K_i = R_i 's handled knowledge;

After several meetings, a **library of scenarios** is progressively built and enlarged.

Remark In order to verify or refine collected knowledge, some feedback is necessary. It can be a collective knowledge refinement (i.e. new meetings) or individual interviews.

Note. Generally, the chosen groups during the abstraction step have a heterogeneous organizational structure. This allows various scenarios for the same objective.

Individual knowledge acquisition

Individual knowledge is the results of individual interviews. A thematician can be asked to explain his sen-

tences (requests, answers too ambiguous), and also to describe an activity process, that he mentioned during meetings.

The individual knowledge acquisition is a refinement and verification step of the collected information.

Multiactor model

Let p be the number of involved participants and m be the number of scenarios after several meetings.

The multiactor model encompasses the accepted knowledge by domain experts which is:

1. a simplified view of the target system that we call the *abstract system* Φ (Equation 1);
2. a set of pertinent “real agents” A involved in the set of scenarios S . $A = \{A_k\}_{k=1..p}$;
3. for each actor a , $a \in A$, (i) his roles (ii) his knowledge base; (iii) his methods base; (iv) his acquaintances; (v) his language (words, concepts, etc.);
4. a set of role dependencies D , $D = \{D_l\}_{l=1..d}$ (d is the number of discovered dependencies);
5. a library of common objectives O , $O = \{O_l\}_{l=1..o}$ (o is the number of solved objectives during the various meetings);
6. a library of standard scenarios S , $S = \{S_i\}_{i=1..m}$.

Given the sets (1) and (2), the multiactor model Γ is defined as

$$\Gamma = \langle \Phi, A, O, S, D \rangle; \quad (3)$$

CASE STUDY: The Poultry Chain

Our research aims at designing a poultry chain model to understand the contributions of individual actors to the use of raw materials. It will be used to simulate the consequences of new regulations, price and production technique variations.

Our multiactor model was developed in four phases: The abstraction, the collective knowledge acquisition, the information analysis and finally the setting up phases.

The abstraction phase

A poultry chain is a very complex organization. To conceive its abstraction, we have carried out several *individual interviews* with domain experts (expert thematicians) and industrial actors (thematicians).

The abstract architecture (Figure 5) represents a poultry chain as composed of a set of competing firms within the agricultural raw material market and poultry market. Each firm has its own organizational structure, objectives, functioning rules, and a set of local knowledge and skills.

$$\Phi_{PoultryChain} = \langle F, E \rangle; \quad (4)$$

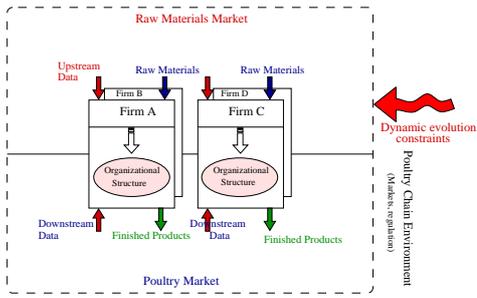


Figure 5: Poultry Chain Abstraction

$F = \{Firm A, Firm B, Firm C, Firm D, \text{etc.}\}$
 $E = \bigcup E_i$, and $E_1 = Raw Material Market$,
 $E_2 = Poultry Market$

The collective knowledge acquisition phase

Initially, we invited to a meeting actors, chosen by a firm manager (firm A, in Figure 5 and Table 1).

Thematically	Roles (as defined by actors)
Purchaser	Proposes new raw material
Formulator	Provides cheapest diet formula following nutritional and economic constraints
Manufacturer	Manufactures animal meal constrained by stock capacities, technological constraints
Nutritionist	Introduces/modifies nutritional constraints
Quality coordinator	Ensures best product quality
Manager	Proposes global strategies Assigns tasks

Table 1: Meeting participant Actors

The meeting objective is chosen by thematians: “new raw material (RM) opportunity, do we buy it?”. Then, a meeting is hold.

The interaction sequencing

To solve the proposed objective, Table 2 shows a sequence of exchanged messages between participant actors. This sequence is a part of the corpus documents.

Information Analysis

We represent the conversations (as depicted in Table 2) using *Speech Acts* (Parunak, 1996). Conversations are analyzed using three tools: UML diagram sequencing, state machines, and Dooley graphs.

Sender	Receiver	Message content
P	F	Opportunity(wheat,price, wheat characteristics) Can we use it?
F	M	planning and stock possibilities?
M	F	I can use it in 07 days.
F		OK ! I compute operation profit
F	P	Yes, we can buy it, but wheat can't be received before 07 days
P	F	OK! I negotiate the delay

F=Formulator, P=Purchaser, M=Manufacturer

Table 2: Interaction sequencing IS

Identified roles

Given the initial set of meeting participants (see Table 1), we identified three pertinent roles (Table 3) .

Actor/role R_i	Acquaintances A_i	Knowledge K_i
Purchaser Opportunity	Formulator	data=RM,Price, RM characteristics
Formulator new RM	Purchaser Manufacturer	method=Optimization program data=RM characteristics table
Manufacturer new RM	Formulator	method=stock control program data=stock capacities, production planning

RM=Raw Material

Table 3: Acquired knowledge

$R = \{Purchaser_{opportunity},$
 $Formulator_{newRM}, Manufacturer_{newRM}.$

Conversation representation: state machine

Figure 6 diagrams the state machine SM_1 of the previous conversation (see Table 2). Note that to refine the acquired knowledge, some individual interviews have been undertaken.

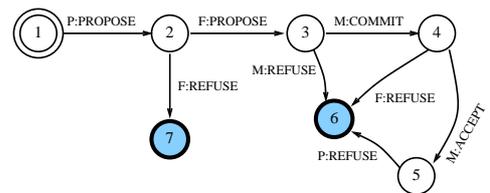


Figure 6: State machine SM_1 : case raw material purchase

Identified scenario: The “RM Opportunity” scenario

The scenario S_1 was identified. Following Formula 2 it is written as:

$$S_1 = \langle \text{“RMopportunity”}, IS, SM_1, R, A, K \rangle$$

IS =Interaction sequences, as depicted in Table 2,
 SM_1 =state machine, Figure 6,
 $R = \{R_1, R_2, R_3\}$,
 $R_1 = \langle \text{Purchaser}_{opportunity}, A_1, K_1 \rangle$,
 $A_1 = \{R_2\}$,
 $K_1 = \{RM, Price, RMcharacteristics\}$.
 $R_2 = \langle \text{Formulator}_{newRM}, A_2, K_2 \rangle$,
 $A_2 = \{R_1, R_3\}$,
 $K_2 = \{OptProgram, RMcharacteristics\}$.
 $R_3 = \langle \text{Manufacturer}_{newRM}, A_3, K_3 \rangle$,
 $A_3 = \{R_2\}$,
 $K_3 = \{stockControlProgram, stockCapacities, productionPlanning\}$.

Dependencies between roles

A number of role dependencies are identified. For example:

1. $d_1 = \text{Dependency}(\text{Purchaser}_{opportunity}, \text{Formulator}_{newRM}, \text{resource})$: a material resource dependency (optimization program, handled by the Formulator) between $\text{Purchaser}_{opportunity}$ and $\text{Formulator}_{newRM}$ roles;
2. $d_2 = \text{Dependency}(\text{Formulator}_{newRM}, \text{Manufacturer}_{newRM}, \text{resource})$: an informational resource dependency (stock capacities) between $\text{Formulator}_{newRM}$ and $\text{Manufacturer}_{newRM}$ roles.

The multiactor model setting up

Following the scenario S_1 and the Formula 3, the developed multiactor model represents only one firm (Let be $Firm A$) of the poultry chain. It is defined as:

$$\Gamma = \langle \Phi_{PoultryChain}, A, D, O, S \rangle;$$

Where

Φ = Poultry chain abstraction, defined by (4)
 $A = \{Firm A:\text{Purchaser}, Firm A:\text{Formulator}, Firm A:\text{Manufacturer}\}$
 $D = \{d_1, d_2\}$
 $O = \{\text{Raw Material Purchase}\}$
 $S = \{S_1\}$

Additional results

After several meetings, we distinguished several individual and collective decision tasks (e.g. raw material purchase). Moreover, two standard scenario classes have

clearly emerged: *strategic* and *tactic* scenarios. The former can be related to strategic objectives, such as consequences of regulation variation or the increase of firm profits. The latter can be associated with usual problem solving, such as raw material purchase.

Remark

After several meetings, a lack of interaction issue emerged. Some meeting participants did not interact to solve objectives. In such cases, these actors can be considered as not pertinent for the model. This issue is a restrictive parameter to be addressed.

RELATED WORKS

To build a MABS, two main proposal classes are taken into consideration. The first class consists of participatory approaches which are based on knowledge acquisition processes. These approaches define a domain model to represent the expertise knowledge of stakeholders (Barreateau, Bousquet and Attonaty, 2001; Bousquet et al., 2002; Bars, Attonaty and Pinson, 2002). A knowledge elicitation process integrates acquired knowledge into the expertise model (Iglesias et al., 1996; Fishwick, 1997) of the conceptual agents. The second class is MAS methodologies. These methodologies base the MABS building process on the analysis of the descriptions of the target system and model objectives. However, most MAS methodologies (Parunak et al., 1997; Burmeister, 1996; Kendall, Malkoun and C.H.Jiang, 1996) consider that agent identification is a straightforward operation. Because of this lack, such methodologies are not easily applicable to human or cognitive organizations modelling.

From our understanding of the litterature, until now the proposed knowledge acquisition approaches to build social MABS have been based on individual interviews (Barreateau et al., 2001; Bousquet et al., 2002). Our approach is based on a cooperative knowledge acquisition to avoid skews, possibly resulting from the individual KA techniques.

Very few MABS studies (Edmonds, 2001; A.Drogoul et al., 2002; Barreateau et al., 2001) explicitly describe participatory approaches as well as the roles involved in the design process. On a side, Drogoul and al. (A.Drogoul et al., 2002) introduce three roles: the mathematician, modeller and computer scientist. Each role intervenes in a model building stage. Those authors propose to define several agents that can learn the expert knowledge. To do so, experts and non-experts exchange knowledge by role-playing games to interactively define agent behaviors. On other side, F. Bousquet (Bousquet et al., 2002), O. Barreateau (Barreateau et al., 2001) and M. Lebars (Bars et al., 2002) build multiagent systems on the basis of individual knowledge acquisition and led participatory simulations on these MAS. In their studies, they simulate the impacts of actor choices during role-playing

games on shared natural resources, and then use the simulation results to resolve conflicts between the stakeholders.

MAS methodologies propose models for conceptual agents. The agents are identified following several approaches, based mainly on the analysis of the furnished descriptive data. For instance, Parunak and al. (Parunak et al., 1997) define a set of agents and agent types using a linguistic case analysis of the problem description. This activity focuses roughly on the Burmeister agent model (Burmeister, 1996). The initial set of agents is used to acquire expert knowledge and behaviors by role-playing or computer simulations. In fact, some actor behaviors are considered as strainforward, thus an expert validation step is necessary.

In role-based methodologies (Wooldridge et al., 2000; Ferber and Gutknecht, 1998), the properties of a role (Wooldridge et al., 2000) are used to create a system behavior model. Roles are then defined and mapped to various conceptual agents. To define the agent behaviors, these methodologies require the designer to have expertise in the target system, so the roles can be identified correctly.

These proposed approaches are individual-centred and neither deal with the descriptive aspects of organizations nor address the tacit knowledge acquisition issues. Our approach introduces participation principle to design socio-economical MABS. It uses cooperative processes to discover and acquire collective (macro-knowledge) as well as individual knowledge (micro-knowledge). Its main principle is to cause an effective participation of actors, so it is pictured as a fly trap.

CONCLUSION

This paper describes our proposed approach to collect necessary knowledge for MABS building. It is a participatory approach, based on role-playing, meeting and CSCW principles. It also provides a dialogue framework as a communication tool for participating actors. In order to understand organizational structure and functioning of multiactor systems, actors simulate cooperation processes within firms by role-playing games. The interaction sequences are saved and analyzed to build a domain model called *multiactor model*.

Our approach has, as a first challenge, to introduce MABS development processes within industrial organizations. To do so, we have explored some experiments in different firms. At this time, we are using the collected knowledge to develop a poultry chain multiagent based simulation.

The strongest contribution of our approach is its ability to acquire precise collective as well as individual knowledge. However, until now the acquired knowledge is hand-coded, thus it presents a weakness point to be addressed in further works.

AUTHER BIOGRAPHIES

ATHMANE HAMEL is Ph.D. Student of Computer Science at University Paris 9, Dauphine. He received a Master of Computer Science from Paris 9 University. He is now undertaking research in the multiagent systems and knowledge acquisition fields. His email is hamel@grignon.inra.fr.

JEAN-MARIE ATTONATY is senior researcher at INRA. His research area concerns small scale enterprise management. He has developed decision making support for individual enterprise and negotiation support for enterprises. These models are based on AI and DAI concepts and techniques. At this time, his main interest is knowledge acquisition from several deciders. His email address is attonaty@grignon.inra.fr

SUZANNE D. PINSON is Professor of Computer Science at University Paris 9 Dauphine. She is Head of the Research Group in Artificial Intelligence and Decision Processes at the CNRS LAMSADE laboratory. She received a M.S. and a Ph.D. in Computer Science from Paris 6 University as well as a M.S. in Computer Science and Operational Research from Northwestern University, Evanston, Illinois. She wrote a book on Cluster Analysis and has published numerous papers on Data Analysis, Knowledge Based Systems, Distributed Artificial Intelligence and Multi-Agent Systems. She is on the editorial board of several international journals. Her field of research covers Distributed Decision Making and Multi-Agent Systems, more precisely, coordination and cooperation in MAS based on a decision theory approach. Other research interests include framework for the development of management and financial distributed decision support systems. Her email is pinson@lamsade.dauphine.fr.

ACKNOWLEDGEMENTS

The case study is being carried in collaboration with Isabelle BOUVAREL (Institut Technique d'AViculture) and Michel PICARD (INRA-SRA) Tours.

References

- A.Drogoul, Vanbergue, D. and Meurisse, T.: 2002, Multi-agent based simulation: Where are the agents?, *Proceedings of MABS'02 (Multi-Agent Based Simulation, Bologna, Italy, July 2002)*, LNCS, Springer-Verlag.
- Barreteau, O., Bousquet, F. and Attonaty, J.: 2001, Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to senegal river valley irrigated systems, *Journal of Artificial Societies and Social Simulation*, Vol. 4.

- Bars, M. L., Attonaty, J. and Pinson, S.: 2002, Manga: An agent-based simulation to improve negotiation between different users in resource management. an application to the water sharing problem, *12th EURO Mini Conference, Brussels (Belgium)*.
- Bousquet, F., Barreteau, O., d'Aquino, P., Etienne, M., Boissau, S., Aubert, S., Page, C. L., Babin, D. and Castella, J.-C.: 2002, *Complexity and Ecosystem Management: The Theory and Practice of Multi-agent Approaches*, Edward Elgar Publishers, chapter Multi-agent systems and role games: collective learning processes for ecosystem management.
- Burmeister, B.: 1996, Models and methodology for agent-oriented analysis and design, *Technical report d-96-06, dfki*.
- Edmonds, B.: 2001, The use of models: Making mabs more informative, in P. D. E. L. . S. Moss (ed.), *Proceedings of MABS'00 (Multi-Agent Based Simulation, Second International Workshop, Boston, USA, July 2000)*, Springer-Verlag Heidelberg.
- Ferber, J. and Gutknecht, O.: 1998, A meta-model for the analysis and design of organizations in multi-agent systems, *Proceedings of Third International Conference on MultiAgent Systems (ICMAS'98)*, IEEE Computer Society, pp. 128–135.
- Fishwick, P.: 1996, Computer simulation: growth through extension, *Proceedings of Third International Conference on MultiAgent Systems (ICMAS'98)*, IEEE Potential, pp. 24–27.
- Fishwick, P.: 1997, The comomas approach: From conceptual models to executable code.
*[http://citeseer.nj.nec.com/\(1997\)](http://citeseer.nj.nec.com/(1997))
- Gilbert, N. and Troitzsch, K.: 1999, *Simulation for the Social Scientist*, Open University Press.
- Iglesias, C. A., Garijo, M., Gonzalez, J. C. and Velasco, J. R.: 1996, A methodological proposal for multi-agent systems, development extending commonkads, *Technical report*.
*[http://citeseer.nj.nec.com/\(1996\)](http://citeseer.nj.nec.com/(1996))
- Kendall, E., Malkoun, M. and C.H.Jiang: 1996, A methodology for developing agent based systems, in C. Zhang and D. Lukose (eds), *Distributed Intelligence Architecture and Modelling*, LNAI 1087, Springer-Verlag, pp. 85–99.
- Parunak, V.: 1996, An introduction to speech acts and dooley graphs, *Technical report*.
*<http://www.iti.orgb-van>
- Parunak, V., Sauter, J. and Clark, S.: 1997, Toward the specification and design of industrial sythetic ecosystems, *Fourth Workshop on Agents Theories, Architectures, and Languages (ATAL'97)*.
- Wooldridge, M., Jennings, N. R. and Kinny, D.: 2000, The gaia methodology for agent-oriented analysis and design, *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 3, pp. 285–312.
- Yu, E. and Mylopoulos, J.: 1994, Understanding “why” in software process modelling, analysis and design, *Tech. report dkbs-94-3, dept. comp. sci., univ. of toronto*.

MULTI AGENT BASED SIMULATION FOR DATABASE SECURITY: A FRAMEWORK

Marco Remondino
Department of Computer Science
University of Turin
10149 Turin, Italy
E-mail: remond@di.unito.it

ABSTRACT

Agent Based Simulation is successfully applied to enterprise modeling and social sciences, and is considered as a third way to represent models, alternative to the verbal argumentation and the strictly mathematical approach. The advantage over the other two is its high portability on a computer, in order to be executed, and its flexibility, which makes it the optimal tool to represent complex systems. The purpose of this paper is to discuss the possibility of using the Multi Agent paradigm to simulate a database, in regard to the security policy applied; databases are very complex objects, and thus an Agent Based approach should allow to exploit the interactions among users and the results deriving from a particular security policy. Besides, the necessity of granting a certain security level for data access often compromises the efficiency of data retrieval, and thus the optimal balance of the two is a difficult task to accomplish. The creation of an Agent Based simulation, which models a database security environment, allows what-if analysis and case study, at the variation of defined rules and parameters, without trying to change the security policy in the real environment. An operative framework with rules, to be used for the creation of a model representing a generic database, subject to a Discretionary Access Control policy, is then proposed and studied, in order to simulate the effect of security rules, through the modification of some initial parameters.

INTRODUCTION

According to (Ostrom, 1988), simulation can be considered a third way to represent social models; in particular, it can be a powerful alternative to other two symbol systems, the verbal argumentation and the mathematical one. Simulation has a great advantage over the other two, which is its high portability on a computer, through a program or a particular tool. In particular, Agent Based Simulation is optimal for modeling complex systems, which couldn't be ported to a computer in any other way. This approach allows to observe the emergence of complex behaviour, through the creation and study of models, known as Artificial Societies. Thanks to the ever increasing computational power, it has been possible to use these concepts to create software models, based on intelligent agents, which aggregate behaviour is often difficult to predict

just studying the single parts, without considering the interaction among them and with the environment.

In this paper we'll propose an operative framework for the creation of a Multi Agent Based Model of a generic database, to simulate the security rules applied to it, and to verify the various effects they have on efficiency, time and data corruption, by modifying some core parameters. A database is always a very complex object, managed by many different rules, hence the idea of simulating the security environment using a Multi Agent Based Model. Besides, the necessity of granting a certain security level for data access often compromises the efficiency of data retrieval, and thus the optimal balance of the two is a difficult task to accomplish. The creation of an Agent Based Simulation, which models a database security environment, can give answers to what-if situations, at the variation of defined rules and parameters, without trying to change the security policy in the real environment.

MULTI AGENT 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, and 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 simulation:

- Symbolic: highly structured agents, described through expressions of modal logic. This paradigm 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. Here the stress is on interaction and cooperation and not on the single entities. 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: on the single agents can thus be defined some fundamental macro-actions, which allow cooperation with the environment and with other agents. The concept of Multi Agent System for Simulation of Complex Systems 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.

DATABASE SECURITY

A Data Base Management System (DBMS) is defined as a software package, designed to store and manage databases, which are very large, integrated collections of data. A DBMS allows to reach the following objectives:

- Data independence and efficient access
- Reduced application development time
- Data integrity and security
- Uniform data administration
- Concurrent access, recovery from crashes

It is then obvious that one of the fundamental goals of a DBMS is to reach a security level which could prevent users with no specific grants to read data. It's also very important to reach a satisfying level for data integrity, by preventing users without an authorization to modify them. On the other side, it's necessary to reach an high efficiency for data retrieval, when the users have the specific rights. A security policy applied to a database must specify who is authorized to do what, and a security mechanism allows to enforce a chosen security policy. There are two main mechanisms at the DBMS level: Discretionary Access Control (DAC) and Mandatory Access Control (MAC).

The former is based on the concept of access rights or privileges for objects (i.e. tables and views), and mechanisms for giving and revoking users privileges; in this model, the creator of a table or a view automatically gets all privileges on it. The DBMS keeps track of who subsequently gains and loses privileges, and ensures that only requests from users who have the necessary privileges, at the time the request is issued, are allowed. The fundamental command, in this paradigm, is GRANT:

GRANT privileges ON object TO users [WITH GRANT OPTION]

In this way, the specified users get the privileges on the object belonging to the DB; usually, the privileges are the following ones:

- SELECT: Can read all columns (including those added later via ALTER TABLE command).
- INSERT(col-name): Can insert tuples with non-null or non-default values in this column. Similarly, UPDATE.
- INSERT means same right with respect to all columns.
- DELETE: Can delete tuples.
- REFERENCES (col-name): Can define foreign keys (in other tables) that refer to this column.

If a user has a privilege with the GRANT OPTION, he can pass it on to other users, in turn with or without passing also the GRANT OPTION. Privileges can of course be lost, through the REVOKE command; if a user loses his privileges on an object, also the ones who had them from him will lose them. A user can receive the same privileges from different subjects and, in this case, he would lose them only if all these users lose those privileges on the object.

While in SQL-92, privileges are assigned to authorization ids, which can denote a single user or a group of users, in SQL:1999, and in many current systems, privileges are assigned to roles, that can then be granted to users and to other roles. This approach reflects how real organizations work and illustrates how standards often catch up with de facto standards embodied in popular systems.

Differently from the model described above, the MAC is based on system-wide policies that cannot be changed by individual users. Each object in the database is assigned a security class and each subject, user or user program, is assigned a specific clearance for a security class. The rules based on security classes and clearances govern who can read or write which objects. The MAC was born to overcome a typical flaw of the discretionary system, known as Trojan Horse. In fact, user A could create a table, on which he has all the privileges, and then can grant to user B the INSERT privileges on it. User B has privileges on, and thus can access, another table, containing secret data, which are forbidden to user A; then, user A modifies the code of an application program used by user B to additionally write those secret data to the newly created table, and so user A can now access these secret data. Bell-LaPadula model defines the main rules for the management of MAC. In this model we find:

- Objects (e.g., tables, views, tuples)

- Subjects (e.g., users, user programs)
- Security classes: Top secret (TS), secret (S), confidential (C), unclassified (U)
- An order for the classes: TS > S > C > U
- Each object and subject is assigned a class:
 - Subject S can read object O only if class(S) > class(O) (Simple Security Property)
 - Subject S can write object O only if class(S) = class(O) (*-Property)

The main idea is to ensure that information can never flow from a higher to a lower security level. This, obviously, avoids the Trojan Horse problem. The MAC rules are usually applied in addition to any discretionary controls that are in effect.

AN AGENT BASED MODEL FOR DB SECURITY

In the following description of the model, we will represent a database organized according to the DAC, which is simpler and easier to port to a programming language. The use of an Object Oriented Language (OO) is assumed; this kind of languages (C++, Java) allows the creation of many independent objects, without having to write specific code for each of them. Besides, in an OO language, there are proprieties such as inheritance and polymorphism, useful for this model.

We can think of a set of agents, which are the users of a database, organized into a hierarchy; in general, a community of agents which can access data according to specific rules. When the single agent needs a datum, he first looks for it and, if he can't access it directly, he asks other agents, who have the specific permission on it. Hierarchy and proximity relations are defined among the agents: there are n levels, organized into a pyramid. Level 1 is the upper one, while Level n is the bottom. Besides, on the same level, proximity relations can be defined: a list could also exist, called Project Colleagues, containing the IDs of the agents working on the same project, so that if one of those creates a table or an object, the other agents will automatically have the access to them. Data inside the database are modelled as very simple objects, which have: a unique number, so that they can be called and retrieved; an identifier, signalling if the datum is corrupt or fine; the ID of their creator. Besides, there is a variable, associated with the datum, which signals whom accessed it for the last time; this is used to keep track of whom damaged it, if the datum is not fine anymore. Each user agent has his own unique ID, representing the name of the subject; a number, identifying the level to which he belongs and a list of the privileges on data; each element of the list is an array with the following elements:

- 1) datum number (code)
- 2) read privilege (yes/no)
- 3) ID of the subject that granted the privilege at point 2
- 4) write privilege (yes/no)

- 5) ID of the subject that granted the privilege at point 4
- 6) delete privilege (yes/no)
- 7) ID of the subject that granted the privilege at point 6

Besides, each agent has another list, containing the privileges he granted to others. Again, each element of the list is an array, with these elements:

- 1) datum number (code)
- 2) IDs of the subjects to whom read permission has been granted
- 3) IDs of the subjects to whom write permission has been granted
- 4) IDs of the subjects to whom delete permission has been granted

Obviously, an agent can't grant a privilege on a datum if he hasn't got it himself. When a subject creates an object, an array is automatically inserted in his list, with the new datum number (code) and all the privileges on it. Besides, Colleagues List described above could be implemented and in this case, when an object is created, all the subjects in this list will automatically have the privileges on it. Each user has also an unreliability index, which is increased each time he damages data, after a write operation. When an agent must complete an operation on a certain datum, he tries to access it directly, by looking in its list if he has the needed privileges on it. If he has the privileges, he access the datum in a single time unit, t. This process is shown in Figure 2: the user B, belonging to the third level of the pyramidal hierarchy, wants to access datum_8, in the central database. When he contacts the DBMS (1), the datum attributes (2a) are compared to the list of the privileges of the user (2b). If the user has the needed privileges, he can immediately access the datum (3), and the operation is finished in a time t.

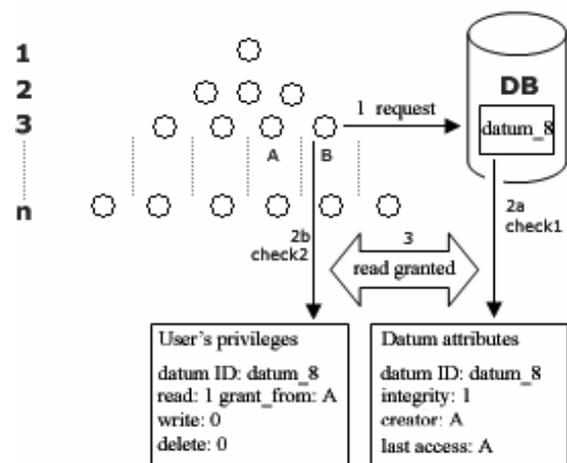


Figure 1 – access attempt, with privileges

If he hasn't got the needed privileges, he asks the datum who its creator is; this requires a time equal to that of retrieving the datum directly, that is t ; as a reply, the creator ID is returned. According to the adopted security policy, which in the simulation can be changed by the user, the agent will ask for the grant directly to the creator or, in the most inflexible case, he will have to move up in the hierarchy, asking for the grant to an user at the level which is immediately upper, and so on, till when he meets one that has the needed privileges. In the worst case he will need to go all the way up to the creator; obviously, each request will consume some time, which can be considered equal to $t/2$. When the user meets an agent with the required privileges, he asks him to pass them to himself, consuming again a time $t/2$.

According to the inflexibility of the security policy, selectable before the simulation starts, and according to the unreliability index of the subject, the privileges will or won't be granted. If they are granted, the user will be able to access the datum, in a time t ; again, according to the security policy, the privileges can be kept by the user or can be immediately revoked after the operation has been completed. If the privileges are not granted, the user which owns them will access the datum on behalf of the requesting agent in a time $2t$. Of course, if the same user has to access the datum again, he will have to pass again through all these steps. In Figure 3, we show an example of the described case: user B must access datum_8 (1), but after verification (2a and 2b), he realizes he can't do that directly. The datum then returns the ID of its creator, that is user A (3), who is on the same level as B can thus be contacted directly (4) by B.

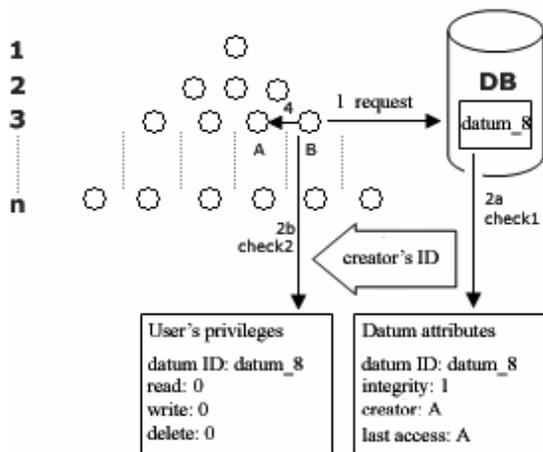


Figure 2 – access attempt, without privileges

With some simple calculations, we see that in the most inflexible situation, if the agent accessing the datum is not its creator, the access time, for each operation, is:

$$3t + t \left(\frac{\Delta + 1}{2} \right)$$

where Δ is the distance between the level of the user requesting privileges and the creator of the object that is being accessed. This time must be compared to time $3t$, needed for data access in the most flexible situation, in which privileges are always granted. When an agent has the write privilege on a datum, there is a probability function which determines the possibility that this operation compromises its integrity. During the write operation, the user could modify the flag variable of the datum from 1 to 0.

The next time this datum will be necessary, the problem will emerge: this will increase the index of system unreliability and will waste a time $2t$, to restore the datum. The user who damaged it, whose ID is stored in the Last Access variable of the datum, will have its personal unreliability index increased and, according to the security policy adopted, will lose or not the privileges on that datum. When a user loses his privileges on a certain datum, also the agents who had the privileges from him will lose them. It could be possible that an agent had received the same privileges from more than one user: in this case, he will keep the privileges, unless all the granting agents lose them. The need of the users to access data is controlled by random functions, and so it the probability function for data corruption.

This probability increases with the growth of the delta between the level of the creator and that of the user accessing the data. During the execution of the simulation, two real-time graphs will be created: one will represent the average time for data retrieval; the other one will represent the general unreliability index of the system, derived from the average of corrupted data. By varying the security policy, through the initial parameters, it will be possible to compare different situations, after the same number of simulation steps and with the same random seed. The security policy affects the probability for an agent to grant the privileges to another user, on certain data. A probability equal to 0 means that no agent will receive the privileges, so that only the owners can access the data they created. In this case, the general unreliability index will be very low, but the time for data retrieval will reasonably be very high. A probability equal to 1 means that the privileges are always granted, no matter who asks for them: of course this will bring to an opposite situation. The intermediate cases, i.e. a probability between 0 and 1, are the most interesting and difficult to predict, but also the most useful to model real situations.

Also the creation of new data is managed in a random way, and at the beginning of the simulation some steps

will be dedicated to this activity. We can think of a number of data with an inverse proportion in respect of the level or, in alternative, we can put in the simulation the exact situation that we observe in the organization we want to model.

The last case that needs to be considered is the request of privileges on certain data by a user who is at an upper level than their creator; there could be an automatic grant or, if we want to be more realistic, the request could be addressed directly to the creator, without needing to move down in the hierarchy, but using the rules defined in the security policy, that consider the personal unreliability of the requesting agent.

A BASIC IMPLEMENTATION

While the research is still in progress, we created a working example of the “worst possible case”, to show that the simulation is feasible and that Agent Based Technology can be successfully applied to database security simulation. In this first implementation, we had to simplify several rules, when compared to the ones described in the previous paragraphs; these will be added in future implementations of the model.

The implementation is an agent based simulation realized with Java Agent-Based Simulation library (<http://jaslibrary.sourceforge.net>), by Alessandro Cappellini (cappellini@econ.unito.it). This simulation can represent an ordinary un-secure database, or a normal data warehouse, (e.g. a “soho” fileservers and its directories); in fact we haven’t got any restriction, and everybody can access everyone’s data. This is not the only simplification introduced in this first implementation: there is just a single level of agents (no hierarchy); just two basic operations are defined on data (read and write); no new data are created; damage data are not repaired; no access policy is defined.

The results are quite straightforward, since even if the probability to corrupt data is very low, sooner or later the database will collapse (100% failures, as shown in Figure 4).

Though, this simple model is just the basis to implement all the other rules described above. At each “tick” the agents access a random datum inside the db; they can access the datum to read it (probability 70%) or to modify it (probability 30%). They check the owner of the datum and its integrity; if the datum is corrupt, the access fails. When they access the datum to modify it, there’s a probability (10%) that they can corrupt it.

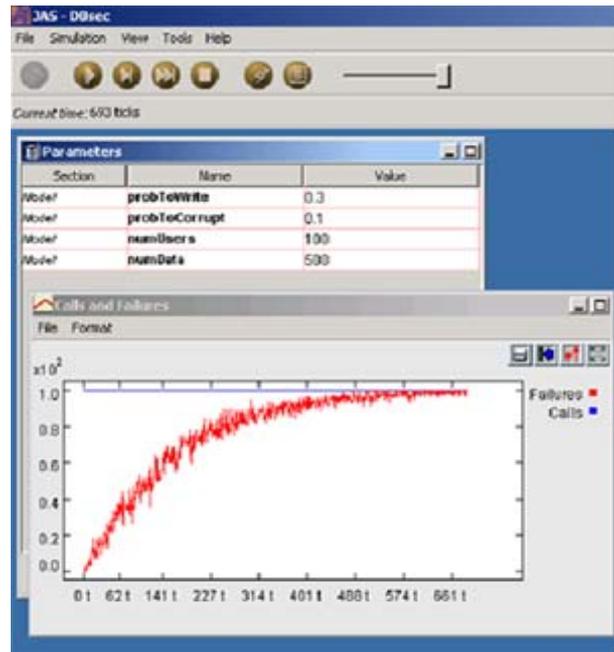


Figure 3 – The Basic Case being Simulated

CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have described a complete framework with rules, for the creation of a software Agent Based Model, to be used to simulate the behaviour of a DBMS, with regards to database security under a DAC. Three are the fundamental features, considered for the simulation: data integrity, time for retrieval and flexibility of the security policy used. While the first two are the dependant variables, the last one is the independent variable, which is the one that can be set by the user, through some parameters, before the simulation starts. The main purpose of this work is to demonstrate the feasibility of an agent based software simulation of database security, which would allow a what-if analysis for designers and users. A simplified working model is then shown, in which many rules described in the theoretical framework are not yet implemented. This is already interesting, since it shows that an agent based model of a database is indeed feasible. Since the research is still in progress, we intend to further develop the simulation tool, by adding all the features described in the theoretical framework, thus converting it into an operative model of a real database.

REFERENCES

- Bertino E. et al. 2001. “Intelligent Database Systems”, Addison Wesley Professional
- Huhns, M. and Singh, M. 1997. “Readings in Agents”, Morgan Kaufmann
- Gilbert, N. and Terna, P. 2000. “How to build and use agent-based models in social science”, *Mind & Society* 1, 57-72

- Gilbert, N. and Troitzsch K.G. 1999. "Simulation for the Social Scientist", Open University Press
- Khalil E.L. 1996. "Social Theory and Naturalism" in Khalil E.L. and Boulding K.E. [Eds.] 1996. "Evolution, Order and Complexity". Routledge Publ., London
- Ostrom T. M. 1988. "Computer simulation: The third symbol system", *Journal of Experimental Social Psychology*, 24:381-392.
- Ramakrishnan R. and Gehrke J. 2000. "Database Management Systems" (2nd Edition), McGraw-Hill

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.

ALESSANDRO CAPPELLINI, author of the basic implementation in JAS, is a PhD Student in Simulation, University of Turin, Italy.

AGENT- BASED MODELLING OF HUMAN ACTING, DECIDING AND BEHAVIOUR - THE REFERENCE MODEL PECS

Bernd Schmidt

Bernhard Schneider

University of Passau

Innstr. 33, D-94032 Passau, Germany

E-Mail: schmidtb@fmi.uni-passau.de, schneidb@fmi.uni-passau.de

KEYWORDS

Agent- based simulation, reference model, PECS, human behaviour, emotional intelligence.

ABSTRACT

Agent- based modelling and simulation is a powerful tool in studying complex systems. For the purpose of designing related simulation models, it is helpful to utilize already existing problem- related design patterns: reference models.

The PECS reference model provides capabilities for object- oriented model specification. Its application area is settled in the field of agent- based simulation, especially for models deriving from studies in sociology, economy and psychology. Major aspect is the modelling of human acting and deciding, interaction of human beings with each other and with their environment, where a human being is considered a psychosomatic unit with cognitive capacities embedded in a social environment. Accordingly, the PECS architecture takes into account physical conditions, emotional states, cognitive capabilities and social status that are represented each by specialised components: Physis, Emotion, Cognition and Social Status from which the name of the PECS reference model derive.

The presented case study is supposed to explain the interplay between the PECS components in order to control and to coordinate the available reactive, deliberative and reflective behavioural patterns that enable a PECS agent to show the most complex form of human behaviour: emotional intelligent behaviour.

INTRODUCTION

Agent- based methods are suitable for constructing simulation models in which human behaviour is of importance. Human behaviour must not be reduced only to cognitive capabilities. More complex theories, among them the ψ - Theory by Dörner (Dörner 1999), additionally take into consideration physical and emotional circumstances as well as human interplay with a social background. In order to follow that

approach, it gets necessary to model all known modes for the control of behaviour: reactive, deliberative and reflective behaviour. A description of these behavioural patterns is provided in (Schmidt 2000).

The PECS reference model described in (Urban 2000) and (Urban 2004) provides capabilities to model these forms of human behaviour. Taking into account a close interrelation between the components Emotion and Cognition, it is possible to model furthermore the most complex form of human behaviour called emotional intelligent behaviour, specified in (Mayer & Salovey 1997).

As a domain independent methodology-founded scheme of construction as proposed in (Klinger 1999), the PECS reference model acts as a pattern for the agent Adam. The agent itself is implemented as a Simplex3- simulation model. A detailed description of the model Adam can be found in (Schmidt 2000). Its implementation realises the ideas and principles introduced in this article and enables the agent to show emotional intelligent behaviour.

1 REFERENCE MODELS AND ARCHITECTURAL PATTERNS

A reference model describes a standard solution for an entire class of problems and serves as a blueprint for a class of real systems having a common deep structure and that do differ only in superficial qualities. Major aim in using reference models is to reduce the complexity of design tasks and thereby reducing the effort in time and work concerning the simulation model development. A reference models capacity depends on the size of its set of solvable problems.

With the PECS reference model, an architecture is proposed that applies to a wide range of systems where human behaviour plays a part. The principal architecture proposed here claims to be applicable for more than just special ad hoc cases. Adopting the general reference model to individual peculiarities of a real system is possible by filling in the empty spaces provided by the architecture. This means, for example, that the number and the type of state variables, the

dependent variables as well as the structure of the transfer function F , the algebraic function H and the output function G can be modified without difficulty. Similarly, the agent can be endowed with a diverse repertoire of actions that indicate the internal and external actions that the agent is capable of. As a result, very diverse agents and communities of agents can be described with the same reference model.

1.1 Agents As Representatives For Humans

The application area for human-like agents is very wide. It comprises among others:

- figures in games,
- actors in movies,
- robots who interact with humans,
- software assistants meant to help users,
- tutors in teaching and learning systems,
- software agents meant to provide information to their clients,
- delegates for customers in e-commerce,
- humans in simulation models including human factors such as workers,
- human beings in social psychological theories.

In order to answer the question of whether human beings can really be modelled and whether human like agents can be designed at all, it must be pointed out that a model is never a direct replica of real facts. A model is always an abridged, reduced version of its original.

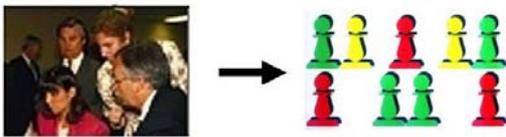


Figure 1: Real human beings and agents as their models

Agents as model-representatives for humans are constructed using the methodology of abstraction and idealisation. This filtering process of idealisation and abstraction reduces real human beings to agents. Figure 1 shows that the rich and colourful real situation is modelled by agents that look very simple. The model does not contain all the qualities that distinguish human beings as human beings. Nevertheless, agents can still have a purpose in science, technology and theory

1.2 System-Theoretical Principles

The basic methodology of the PECS architecture is based on system theory.

A system in terms of system theory is characterised by a set of state variables. These state variables can change their value on the basis of their own dynamics or on the basis of an external input. In addition to the state variables, there can be so called dependent variables, which can be calculated by means of the state variables. The modified internal system state consisting of the new

state variables and the new dependent variables will then lead to an output that can, in some cases, take on the form of an action.

The transfer function F describes how the system state variable $z(t_n)$ turns into the subsequent state $z(t_{n+1})$,

in the time-discrete case:

$$z(t_{n+1}) = F(t_n, z(t_n), w(t_n), x(t_n)) \quad (1)$$

in the time-continuous case:

$$\dot{z}(t) = F(t, z(t), w(t), x(t)) \quad (2)$$

The algebraic function H describes the relation between the state variable $z(t_{n+1})$ and the dependent variable $w(t_{n+1})$:

$$w(t_{n+1}) = H(t_{n+1}, z(t_{n+1})) \quad (3)$$

The output function G determines the manner in which the new internal state, which came about as a result of the input, shows itself as output $y(t_{n+1})$ to the outside:

$$y(t_{n+1}) = G(t_{n+1}, z(t_{n+1}), w(t_{n+1}), x(t_{n+1})) \quad (4)$$

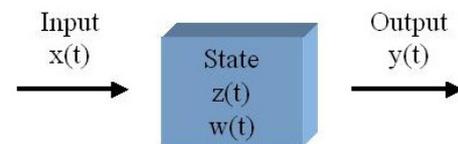


Figure 2: System-theoretical basis for the PECS architecture

An example could be the state variable Fear. It can change by itself or through input from the outside. Figure 6 shows how Fear decreases when nothing happens. A sudden frightening experience as input leads to a sudden increase. It is the function F that describes both these changes in time.

The dependent variable FearM is closely correlated to the state variable Fear. FearM is the corresponding strength of the motive to reduce that fear. The stronger the fear, the higher the value for the motive strength FearM. The algebraic function H determines the dependency of FearM on Fear. (See Figure 7)

2 THE PECS AGENT

The internal structure of a PECS agent is based on a system-theoretic approach and on the usual architecture in robotics. Accordingly, the structure can be divided into

- an input layer consisting of the components Sensor and Perception,
- an internal layer describing the state of the agent and including the components Physis, Emotion, Cognition and Social Status,
- an output layer comprising the components Behaviour and Actor.

Figure 3 illustrates the overall structure of a PECS agent.

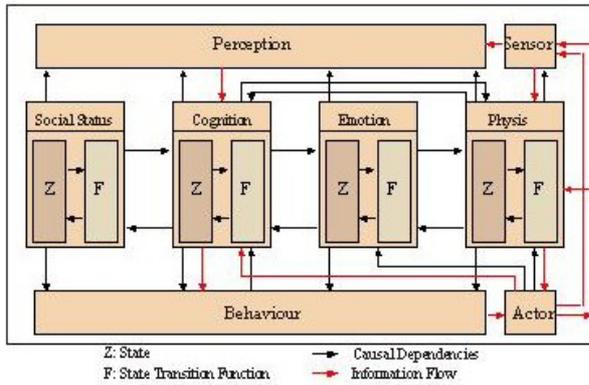


Figure 3: The internal structure of a PECS Agent

2.1 Structure Of A PECS-Agent

The component Sensor is responsible for the reception of sensory input data from the environment of the agent. This sensory information is pre-processed in the component Perception, where information-filtering mechanisms or perceptual processes may be realised.

The components Social Status, Cognition, Emotion and Physis describe the internal state of the agent and contain the state variables of the agent and the associated state transition functions. The component Cognition, in particular, provides space to model the knowledge base as well as the deliberative and reflective behaviour of the agent.

The component Behaviour determines the order in which actions are executed. It contains a set of condition-action rules on the basis of which an execution order is issued. This set of rules is used to model the reactive behaviour of the agent and to co-ordinate the interaction of reactive, deliberative and reflective behaviour. The execution orders are passed on to the Actor, which is responsible for their execution.

The Actor contains the repertoire of actions that the agent is capable of. These actions can be divided into external and internal actions. External actions have an impact on the environment. Internal actions have an effect on an internal component of the agent itself.

2.2 Component Cognition

The PECS reference model is based on the component-oriented, hierarchical modelling principle. Accordingly, complex components can be functionally decomposed into a set of specialised, interconnected sub-components. Following this maxim, the component Cognition of the PECS reference model will be subdivided into the following five components:

- Self Model
- Environment Model
- Protocol Memory
- Planning
- Reflection

Each of these sub-components contains its own state variables and its own state transition function. Of particular interest is the component Reflection, which acts as a supervisor or manager within cognition. It is necessary if the agent should possess reflective capabilities.

The component Self Model contains the agent's knowledge about its own internal state and related operations.

The component Environment Model stores a mental representation of the agent's environment and the processes designed to manipulate and extend this representation such as learning or reasoning.

The idea for a component Protocol Memory was inspired by the approach taken by Dörner (Dörner 1999). The component Protocol Memory gathers information about executed action sequences, formerly pursued plans and methods used to analyse them.

Within the component Planning, planning knowledge and the planning process are modelled. The planning process is responsible for the generation of a plan to reach the agent's intended goal, whereas a plan is a sequence of actions to be performed one after the other. To construct a plan, the component Planning can retrieve information from the components Self Model, Environment Model and Protocol Memory. In PECS planning, algorithms known from Artificial Intelligence are used, such as A* search.

The basic idea of having a component Reflection was taken from Sloman, who proposed a three-layered architecture for human-like agents including a Meta-Management-Layer (Sloman 2000). The function of the component Reflection is to monitor, evaluate and improve internal processes. In order to perform this task, reflective processes can exchange information with the components Self Model, Environment Model, Protocol Memory and Planning.

A complete description of the PECS reference model is provided in (Urban 2004).

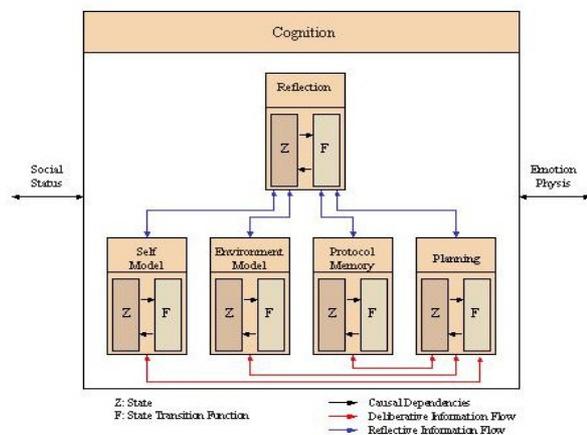


Figure 4: The interior of component cognition

3 MOTIVES AND MOTIVE SELECTION

An important part of the PECS agents is the mechanism of motives and motive-selection.

Motive is defined as a psychological force that drives the organism to execute an action. Motives can be

- Drives
- Emotions
- Acts of will

What all motives have in common is that they appear with a certain intensity and that they may also be in competition with each other. The motive with the highest motive-intensity will determine action.

Since drives, emotions and acts of will together are regarded as motives and as each of these motives has a corresponding intensity, these motives are comparable. It is possible to establish which motive is strongest at a given time and so determine the action to be executed. It is conceivable that an agent experiences hunger and fear simultaneously while also following the goal of deliberately tidying the house.

The three different selection criteria that determine action, namely drive intensity, emotional intensity and intensity of will, are not constant but change with time. This means that different motives may be action-determining at different times. Thus it is possible that at first the act of will *tidy the house* displays the highest intensity of motive. However, after a while, hunger becomes stronger and stronger. It will then overtake the intensity of will that led to the action of tidying. The action of tidying is interrupted. A new motive takes over control. The agent will go to the fridge.

For the modelling of human behaviour there will have to be a motive selector that monitors the development of various motive intensities and ensures that it is always the strongest motive that determines action. A particular example is shown in Figure 5.

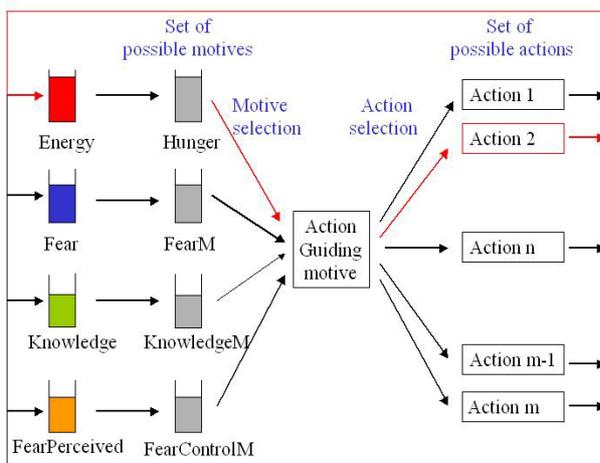


Figure 5: Motives and motive selection

On the left side are internal states that could be the causes of motives.

- *Energy*
Every organism needs energy. An energy deficit will result in hunger as a drive.
- *Fear*
The encounter with a dangerous situation may induce the emotion fear. This fear is correlated to a motive with the motive strength FearM.
- *Knowledge*
An act of will is presupposed that has consciously set itself the goal of acquiring knowledge about an unknown environment. The intensity of will that operates as a motive will be determined, e.g. by the amount of knowledge an agent already has and the amount he still wants to acquire.
- *Fear Perceived*
The state variable is FearPerceived coming from the self-model. Dependent on this consciously known emotion, the strength of will, FearControlM, can be activated, with which the agent is motivated to control his fear.

The motive intensities of hunger as a drive, fear as an emotion-motive and knowledge acquisition as acts of will are in competition with each other. The motive with the highest motive intensity is chosen as the action-guiding motive via motive intensity. From the set of possible actions it determines the action or action sequence that is actually carried out.

In Figure 5, hunger has the highest motive intensity. It will determine the action to be executed. This could, for example, be food search and food intake. These actions will improve the energy level and thus reduce hunger. It follows from this that instead of hunger, an act of will, for example knowledge acquisition, will become the strongest and thus the action-guiding motive. The action of exploration of the environment could then come into play.

The central idea of motive intensity and of the motive selector has been taken from (Dörner 1999), and developed and adapted to the basic concepts of the PECS reference model.

4 MODELLING OF EMOTIONAL INTELLIGENCE

The following investigation is intended to present the methodology that the PECS-reference model follows to specify and model the reflective control of emotion and cognition. It shows that the PECS reference model can even deal with very complicated and difficult processes in a well structured and clear manner.

4.1 Emotion And Cognition

The case study is based on aspects of the psychological concept introduced in 1990 by J. D. Mayer & P. Salovey in *Emotional Intelligence* (Mayer, Salovey 1990) and (Mayer, Salovey 1997). This concept was then popularised by D. Goleman, author of the best-

seller *Emotional Intelligence. Why it can matter more than IQ* (Goleman 1995). Mayer & Salovey defined emotional intelligence as “the ability to monitor one’s own and others’ feelings and emotions, to discriminate among them, and to use this information to guide one’s thinking and action”.

This short definition makes clear that the following aspects are part of the process:

- The emotions and their dynamics,
- The capability to observe and to monitor the actual emotions,
- The cognitive capability to recognise and to categorise the emotions,
- The act of will to influence the emotions and to replace the emotion-induced actions by others more sensible ones.

The following is a short example of the procedure:

An agent encounters a dangerous situation that causes a strong emotion of fear. Emotional intelligence would take into account the following procedure:

- Emotions and their dynamics
The agent becomes afraid and feels fear. Under normal circumstances, the increase in A’s emotion of Fear would lead to an increase in the corresponding motive FearM, that as a consequence would cause the action of fleeing headlong and without consideration.
- The capacity to observe and to monitor the actual emotions
If the agent has a high degree of emotional intelligence he will be able to avoid being overwhelmed by his emotions. He is able to observe what is going on inside. He notices that he has become afraid.
- The cognitive capability to recognise and to categorise the emotions
The agent realises that it is the increased fear that troubles him and that motivates him to flee.
- The act of will to influence the emotions and to replace the emotion-induced actions by others more sensible ones.
If the agent’s emotions stay within manageable limits and the agent’s will is strong enough, he can consciously decide not to yield to his emotions but to control them and replace the original action of fleeing by a more sensible one.

The presented concepts are in accordance with (Canamero 1997), (Damasio 1994), (Moffat et al. 1995), (Ortony et al. 1988) and (Velásquez 1997).

4.2 The State Variable Fear And The Dependent Variable FearM For The Motive Strength

Fear is considered a state variable.

Without an input, that means that without a triggering event from outside, the state variable Fear decreases

continuously over time. This process can be modelled by the following equation:

$$\text{Fear} := \text{FearMax} * e^{\text{FearDecrease} * \text{Fear}} \quad (5)$$

An event from outside leads to a sudden, discrete increase of Fear. Figure 6 shows this discrete increase and the continuous decay of the state variable Fear.

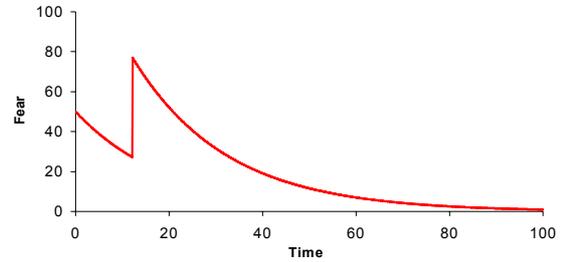


Figure 6: Dynamics of the state variable Fear

The emotional states connection to the corresponding motive FearM is determined by the adoption, that motives can be modelled by following the regularities of a generalized logistic curve of expansion.

$$\text{FearM} := \frac{\text{FearMMax}}{(1 + e^{-\text{FearMIncrease} * (\text{Fear} - \text{FearMTurn})})} \quad (6)$$

Figure 7 shows the dependency of motive FearM on the state Fear.

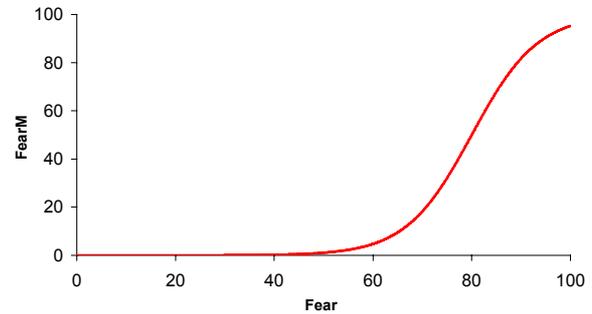


Figure 7: The intensity of the motive FearM as dependent on the state variable Fear

Two points should be kept in mind:

- It is not the state variable Fear but the motive FearM that leads to an action.
- The motive FearM does not always cause an action. FearM has to compete with other motives. If FearM is the strongest motive and dominates all others, only then does FearM determine what the agent does (see Figure 5).

4.3 The Emotional Intelligence Quotient EQ

The state variable EQ is introduced as a measure of the degree of an agent’s emotional intelligence. It is assumed that the agent has an innate EQ that can increase throughout its life span. The agent may gain emotional experience and enhance its capabilities of

emotional intelligence (Mayer & Salovey 1997). To simplify, it is assumed that the process of learning depends on the time taken by the agent to reflect upon its emotional state and that this can be described by the following differential equation:

$$EQ' = EQ_{\text{Increase}} * \frac{EQ_{\text{Max}} - EQ}{EQ_{\text{Max}}} * EQ \quad (7)$$

The EQ has an influence on the agent's ability to perceive, monitor and regulate its emotional state.

4.4 Self-Perception

Self-perception enables an agent to observe and to monitor its own state. The information self-perception provides, is stored in the component self-model.

All reflective considerations are based on the state of the self-model. This means that an agent does not have access to its original state variables but only to the state variables it observes and which entered into its self-model. This accounts for the obvious fact that the motives and the actions of an agent are not determined by the reality of the case, but by what the agent thinks or believes is the reality of the case.

Before an emotion is observed and before the value is transferred to the self-model, the emotion must have a definite strength. This self-perception threshold, $SP_{\text{Threshold}}$, from which onwards an emotion is observed depends on the EQ describes this matter of fact.

$$SP_{\text{Threshold}} := SP_{\text{ThresholdMax}} * e^{-SP_{\text{ThresholdDecrease}} * EQ} \quad (8)$$

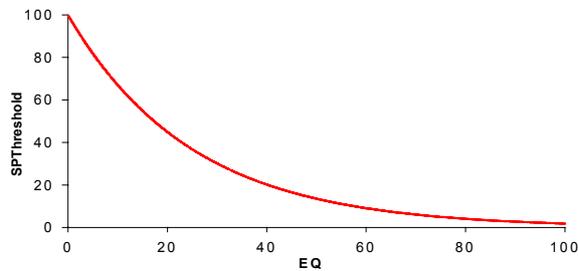


Figure 8: The mutual dependency of the self-perception threshold and the EQ

Figure 5 shows that an agent with a high EQ is already able to observe and monitor slight emotions whereas an agent with a low EQ needs very strong emotions before it is able to perceive them and incorporate them in its self-mode

4.5 The Motive FearControlM

The motive FearControlM is a motive that indicates willpower (see Figure 9). The value of FearControlM depends on the EQ and on the observed value of the emotion FearPerceived.

FearControlM :=

$$\frac{FearControlM_{\text{Max}} * b * EQ}{(1 + e^{-FearControlM_{\text{Increase}} * (FearPerceived - FearControlM_{\text{Turn}})})} \quad (9)$$

Figure 9 shows the dependency of FearControlM on the EQ. One sees that the strength of the motive increases as the EQ increases. This means that with the same observed value for FearPerceived, the higher the EQ the higher the will power to do something against that fear.

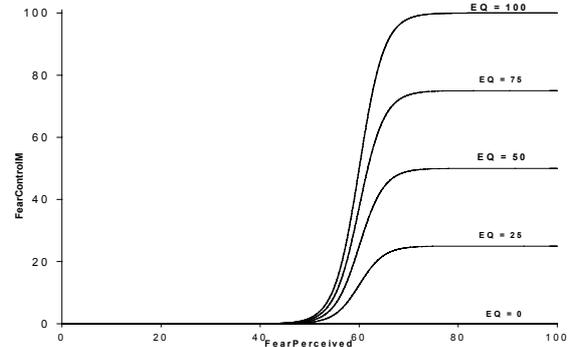


Figure 9: Intensity of the motive FearControlM dependent on the state variable FearPerceived

5 ADAMS MENTAL DYNAMICS

The investigation is conducted within the simulation model Adam. This model Adam is supposed to show and explain the intricate interplay between the various PECS-components.

The model Adam has a broader application than is described here. Emotional intelligence is only one aspect of it. A detailed description of the complete model Adam can be found in (Schmidt 2000).

5.1 Initial Situation

At the beginning, Adam stands on field (3,3). His present state of the state variables and the corresponding motive-strengths indicate that the motive hunger has the highest value and is therefore action-guiding (Figure 10).

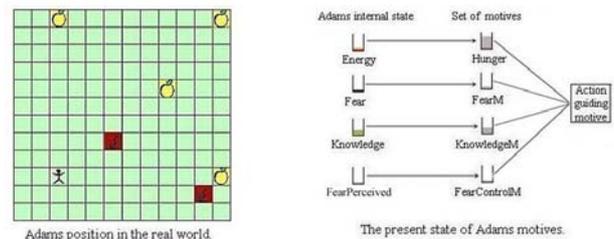


Figure 10: Present state of the agent Adam

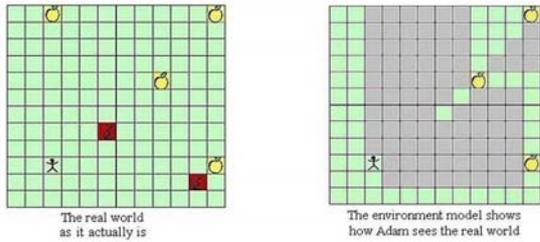


Figure 11: The real world and Adam's environment model

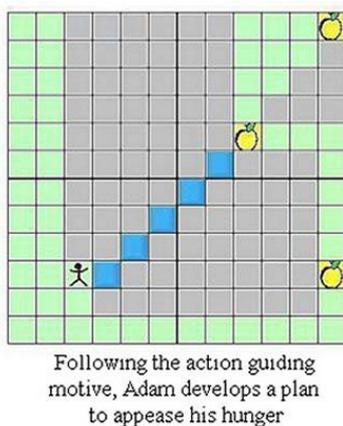
As a consequence, Adam will deliberately develop a plan to reduce his hunger. For this purpose, Adam uses the information that he possesses and that he has integrated in his environment model. Figure 11 shows the environment as it really is on the left side and on the right side the model Adam has of it.

The environment model Adam has is the basis for his planning.

Adam realises that a food source exists on field (9,8). He develops a plan to get there. For his planning he uses the usual algorithms known in Artificial Intelligence. In the present case, an A* algorithm is applied. Figure 12 shows the intended path. One notices that Adam does not know that the field (6,5) is a danger point. Therefore, he cannot take this into account.

Now Adam follows his plan by executing one action after another. He slowly proceeds towards the food source. This goes on until he hits the danger point on field (6,5). At this point, Adam's original plan of action will be interrupted. A new situation arises.

In the following, the very complex and intricate procedures that will take place and that will help Adam to cope with the situation by controlling his emotion of fear will be thoroughly described. This leads to an increased chance of solving the problem.



Following the action guiding motive, Adam develops a plan to appease his hunger

Figure 12: Intended path to the food source

5.2 Extension Of The Environment Model

In a step-by-step fashion, the various actions are explained that take place within the agent Adam:

- (1) Adam encountered a new situation. He performs the action Explore. He notices that he has landed in a trap. This information is taken up by the Sensor.
- (2) This information will be handed on to the component Perception. The component Perception transfers it, in this case unmodified, to the component Cognition.
- (3) The information, that the field (6,5) is a danger point, will be incorporated into Adam's environment model. Adam has extended his knowledge about the real world.

Figure 13 shows these steps.

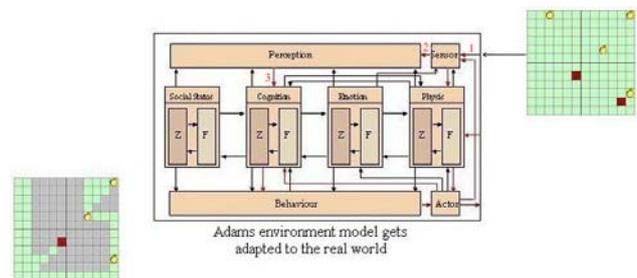


Figure 13: Extension of the environment model

5.3 Emergence Of The Emotion Fear

- (4) The component Cognition evaluates the new facts. It realises that there is a dangerous, fear-inducing situation. This insight leads to a sudden increase in the value for the state variable fear in the component Emotion to a new value of 95 [FU]. This increase is based on the cognitive appraisal theory of emotions.
- (5) With the state variable Fear there is a dependent variable which indicates the motive strength that belongs to the corresponding state variable (see Figure 6 and Figure 7). Figure 7 shows that the value of 95 for Fear leads to a motive strength of 94,2. That would mean that the motive FearM has the highest value and therefore would become action guiding.

Figure 14 shows these steps.

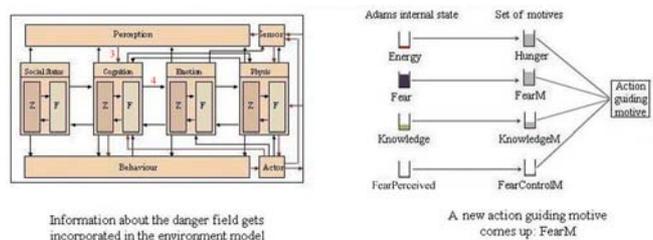
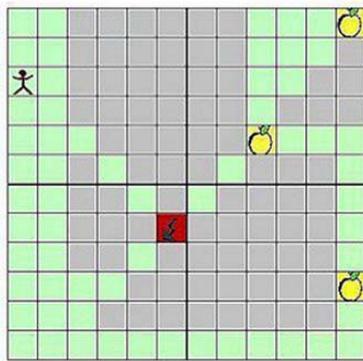


Figure 14: Emergence of the emotion fear



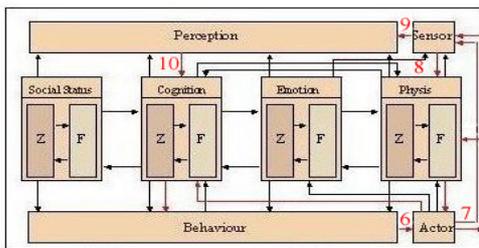
After his imprudent flight, Adam finds himself far away from the next known foot field

Figure 15: The action Flee without a reflective architecture

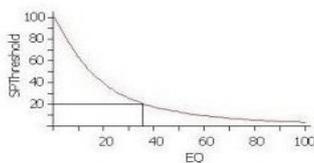
If Adam did not possess the capabilities for reflective behaviour, he would exert the action Flee (see Figure 15). However, if Adam has an architecture which makes reflective behaviour possible, Adam will be in a position to control his fear and to cope with the situation more effectively.

5.4 Realisation Of The Emotion Fear And Its Transfer Into The Self Model

- (6) Because of the new situation, Adam with his reflective capabilities will call the action SelfObserve. This action is supposed to register Adam's own fear and transfer it to the self model
- (7) The component Actor transfers the action SelfObserve to the component Sensor.



The steps, Adam takes to establish his own fear



Information has to pass the self-perception threshold to get handed to the self model

Figure 16: Realisation of the emotion fear and its movement into the self model

- (8) The component Sensor establishes the value for the emotion Fear.

- (9) This information is handed on to the component Perception in the usual way. At this point, the value of the EQ is introduced for the first time. To be able to establish his own fear, the Fear must have a fixed value. The value of Fear must lie over the so-called self-perception threshold. This threshold depends on the EQ in a way that is shown by Figure 8. The higher the EQ, the more Adam is able to observe his own emotion – in this case his fear. With a high value of 90 for EQ [EQU] the threshold is very low. It is only 2,7, meaning that Adam is aware of his fear.
- (10) After the information passes the self-perception threshold, the perceived fear will be transferred to Cognition, which will hand it on to the self model. Adam now consciously “knows” how fearful he is. Using this knowledge he can now try to cope with his fear.

Figure 16 shows these steps.

5.5 The Motivation To Control The Emotion

- (11) The action SelfObserve should cause the component Reflection to control the fear. This SelfControl is, however, not possible in all cases. It depends on Adam's level of general arousal. The higher the general arousal within Adam, the less he will be able to deliberately control himself. Furthermore, the capability for self-control is influenced by the EQ. The higher the EQ, the more Adam will be able to control himself. The so-called arousal threshold is introduced in order to describe this fact. If the actual arousal lies below this threshold, the execution of the action SelfObserve is possible. Adam is not excited enough to be carried away by his emotions.
- (12) The component Actor transfers the internal action SelfObserve to the component Cognition.
- (13) Within the component Cognition, the component Reflection checks the self-model and reads the value of the perceived Fear. In the next step, the value for the motive FearControlM will be determined. The strength of this motive depends on the perceived Fear and on the EQ. The higher the Fear, the more Adam will attempt to reduce his fear. At the same time, Adam's capability to control his fear depends on the EQ. A high EQ value will lead to a high motivation to control the emotions. In the present case, the perceived Fear has the value 90 [FU] and the EQ has the value 90 [EQU]. This leads to a value for the motive FearControlM of nearly 100 (see Figure 9).

Figure 17 shows these steps in detail.

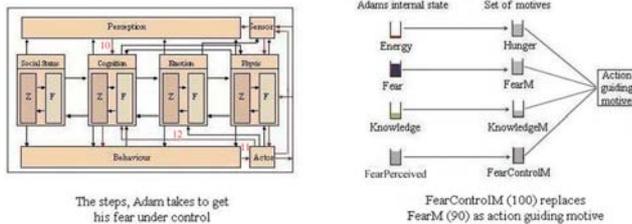


Figure 17: Motivation to control the emotion

5.6 Control Of The Emotion

- (14) The strength of the motive FearControlM is transferred to the component Behaviour. In this component, the various motives and their strengths are compared and the motive with the highest value will be selected and made action guiding. Figure 17 shows the new situation. The motive with the highest value is the motive FearControlM. Therefore, the action Calming will be chosen.
- (15) The action Calming is transferred to the Actor.
- (16) The Actor instructs the Reflection in the component Cognition to execute the action Calming.
- (17) The Reflection interferes with the component Emotion and changes the value for Fear from the value of 90 [FU] to the value of 30 [FU]. The agent succeeded in controlling his behaviour by reducing his emotion of fear.

5.7 The Continuation Of The Actions

Figure 18 shows the values of the motives after the action Calming. The strengths of the motives FearM as well as FearControlM are now noticeably reduced. The original motive Hunger again becomes action guiding. On the basis of his reflective capabilities, Adam was able to control his emotions. He can now deliberately devise a new plan with a sequence of actions that will lead him, like before, to the food source (see Figure 18).

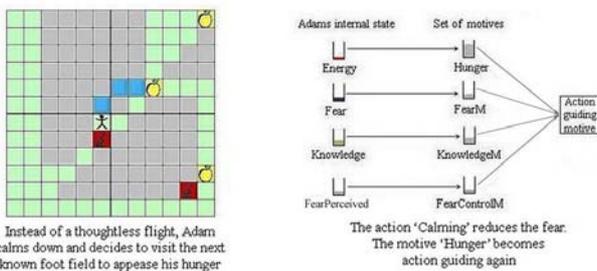


Figure 18: Deliberately devised new plan to the food source

6 CONCLUSIONS

The PECS reference model provides a structured framework for the design of agent systems with a special focus on modelling human-like agents. With PECS

it is possible to construct a wide range of models for agents whose dynamics are determined by physical, emotional, cognitive and social factors and their interactions. Of particular value is the possibility to specify the following three modes of behaviour control:

- Reactive behaviour
- Deliberative behaviour
- Reflective behaviour.

The case study of emotional intelligence has shown in principle how the reflective layer of control can supervise and organise the complex interaction between cognitive and emotional processes.

REFERENCES

Cañamero, D. (1997). "Modeling Motivations and Emotions as a Basis for Intelligent Behaviour", in: Johnson, W. L. (ed.), *Proceedings of the First International Symposium on Autonomous Agents (Agents '97)*, The ACM Press, New York, 148-155.

Damasio, A. R. (1994). *Descartes' Error, Emotion Reason and the Human Brain*, Grosset/Putnam Books, New York.

Dörner, D. (1999). *Bauplan für eine Seele*, Rowohlt Verlag, Reinbeck bei Hamburg.

Goleman, D. (1995). *Emotional Intelligence. Why it can matter more than IQ*, Bantam Books, New York.

Klinger, A. (1999). *Referenzmodelle für die Abbildung von Personalsteuerung in der Simulation*, SCS- European Publishing House, Delft, Erlangen, Ghent, San Diego.

Mayer, J.D., Salovey, P. (1990). "Emotional Intelligence", in: *Imagination, Cognition and Personality*, 9, 185-211.

Mayer, J.D., Salovey, P. (1997). "What Is Emotional Intelligence?", in: *Emotional development and emotional intelligence*, BasicBooks, New York, 3-32.

Moffat, D., Frijda, N. H., P. (1995). "Where there's a Will there's an Agent", in: Wooldridge, M. J., Jennings, N. R. (eds), *Intelligent Agents, Lecture Notes in Artificial Intelligence 890*, Springer Verlag, Berlin, 245-260.

Ortony, A., Clore, G. L., Collins, A. (1988); *The Cognitive Structure of Emotions*, Cambridge University Press, Cambridge, UK.

Schmidt, B. (2000). *The Modelling of Human Behaviour*, SCS-Europe BVBA, Ghent .
or:
[http://www.or.uni-passau.de/english/2/mod behav.php3](http://www.or.uni-passau.de/english/2/mod%20behav.php3)

Seif El-Nasr, M., Yen, J., Ioerger, T.R. (2000). *FLAME - Fuzzy Logic Adaptive Model of Emotions*, International Journal of Autonomous Agents and Multi-Agent Systems, Vol.3., No.3, 1-39.

Sloman, A. (2000). "Architectural requirements for human-like agents both natural and artificial. (What sorts of machines can love?)", in: *Human Cognition and Social Agent Technology, Advances in Consciousness Research*, John Benjamins, Amsterdam, 163-195.

Urban, C. (2000). "PECS – A Reference Model for the Simulation of Multi-Agent Systems", in: Suleiman, R., Troitzsch, K. G., Gilbert, G. N. (eds), *Tools and Techniques for Social Science Simulation*, Physica Verlag, Heidelberg.

Urban, C. (2004). *Das Referenzmodell PECS: Agentenbasierte Modellierung menschlichen Handelns, Entscheidens und Verhaltens*; Dissertation, Faculty for the Department for Mathematics and Informatics, University of Passau, to be published, 75- 129.

Velásquez, J. (1997). "Modeling Emotions and Other Motivations in Synthetic Agents", in: *Proceedings of the National conference on Artificial Intelligence (AAAI-97)*, MIT/AAAI Press.

AUTHOR BIOGRAPHIES



BERND SCHMIDT, Professor at the University of Passau, Department for Operations Research and Systems Theory. For further information please see <http://www.or.uni-passau.de/english/1/schmidt/schmidt.php3>



BERNHARD SCHNEIDER, Student of computer science at the Faculty for the Department for Mathematics and Informatics, University of Passau

VIRTUAL CRIME SCENE RECONSTRUCTION WITH INTEGRATED ANIMATED CHARACTERS

Nicholas Davies, Qasim Mehdi, and Norman Gough

Games Simulation and AI Centre
The University of Wolverhampton
School of Computing and Information Technology
Lichfield Street, Wolverhampton, WV1 1SB West Midlands, UK
N.P.Davies2@wlv.ac.uk

KEYWORDS

3D Scene Construction, DirectX, Skinned Meshes, Computer Graphics.

ABSTRACT

This paper proposes some new techniques for the development of virtual 3D environments for crime scene reconstruction. Current techniques were investigated, which concentrate on the production of photo-realistic scenes. However, these were considered to be too specialist and expensive. The Crime Scene Creator developed here provides a cheaper and quicker alternative that allows quasi-accurate scenes to be constructed and populated with animated characters to act out crimes events, with some very encouraging results. The paper also describes some initial investigations for implementing voice recognition and natural language processing as a basis for further development.

INTRODUCTION

This paper describes research on developing an application for constructing a 3D virtual scene that can be manipulated and viewed from any angle in real time without having to be pre-rendered. This enables investigators to evaluate and eliminate different hypotheses much more rapidly. Current research has developed techniques for the creation of high definition photo-realistic models (Howard *et al* 2000, Little *et al* 2001). However, these techniques are very process intensive and highly technical. The primary goal of the present research was to develop faster methods for creation of quasi-accurate scenes that can be produced with very little training or expensive hardware. Other techniques investigated were the introduction of animated intelligent characters that could act out scenes described to them via Natural Language (NL). This would enable witnesses and victims of crime to develop scenes from descriptive text rather than complicated user interfaces which may require the involvement of a system administrator who could inadvertently influence the final scenes. The paper is structured into 6 sections: Introduction: this section; Background: Current crime scene reconstruction systems; Requirements and Methodology; Implementation; Example: An example of how the system was used to create a scene; Conclusions and Further Work.

BACKGROUND

It is now possible to develop photorealistic 3D models of a scene that can be viewed from multiple angles. The first method investigated was created by Howard and Murta

(2000) in collaboration with the Greater Manchester Police, UK. Data was collected via a set of forensic photographs. These were taken from four corners of a room to give as much coverage of the scene as possible. Photographs were also taken of individual objects with police rulers placed next to them to give an indication of scale. An initial floor plan was produced from measurements taken at the scene and entered into a scene builder in the form of line segments. These were then extruded to produce the 3D room. The height was judged visually rather than using exact measurements. Once the room had been built, objects were placed within the scene depending upon their position gathered from the forensic photographs and measurements. Existing libraries of models that matched objects from the scene were used. These were objects the police had already developed for other applications, as well as some public domain objects. Although some of these items did not match the objects within the scene exactly, as the objects were not of critical importance it was decided that this was of little significance. The completed model allows the viewer to navigate around the scene and view objects from any angle. However, the scene may not have been completely accurate due to estimation of scales, and the inclusion of third party models.

Little *et al* (2001) developed a portable automated system for gathering information from crime scenes. Their technique uses a robotic system with a combination of cameras and laser range finders. The machine automatically moves around the room to get comprehensive coverage. The information gathered by the laser produces a set of points in 3D space. These are combined to form a mesh, and images taken from the cameras are mapped onto it. A model is available for preview as the machine makes its way around the room so a human operative can check for errors. For instance, due to laser technology, the system can have problems with mirrors or other reflective objects. Once the room has been accurately reconstructed, data can be exported to a number of different platforms for rendering, including PC, Unix and Silicon Graphics Machines. The results obtained were far more accurate than scenes developed from photographs; however, the resolution of the models was not as high as that obtained from Howard's technique, and as such was not as visually impressive. Sequeria *et al* (2001) outline a method very similar to that of Little *et al*. It also involves a robotic system that uses camera and laser range finders to produce a 3D reconstruction of the environment. Its approach is slightly more methodical as it plans its next capture point automatically to guarantee complete coverage. Again, the results generated were highly accurate, although no animation was implemented.

Hu and Brown (2002) investigate a method of recreating scenes using mosaic photographs. A mosaic is a single picture of a scene created by taking multiple pictures at regular intervals. This is particularly useful where there is not enough space in a location to get a complete picture from one angle. A camera was placed in three corners of a room and rotated by regular amounts. An algorithm was then employed to determine the 3D structure of features within the room. The resulting scene was of a high quality, although in the conclusion of the report it was noted that the technique had only been tested on simple environments. It is unclear how useful this technique would be in a larger complex environment.

Dyer (2001) outlined a method of scene recreation that uses a volume-based approach. Typically, scene recreation attempts to model shapes within an environment by making a triangular representation of its surfaces. This has some disadvantages. It can be difficult to reconstruct a surface that has sections obscured from certain viewpoints. There must be correspondence across multiple views that indicate that surfaces are linked. Surface patches must be fused together to form a single consistent model. Dyer's technique differs in that it starts with a volume of 3D space and splits it into regular sized cells known as voxels. It then uses multiple views to indicate whether a space is filled or not. When the room has been completely generated, the voxels can either be transparent, indicating empty space, or opaque indicating occupied space. The filled voxels are then coloured to match the surface colour of the scanned object. The advantages of using this technique are that complete coverage of the scene is not required and it is not necessary for the camera to be calibrated exactly. The results of this method were accurate but the images did appear to be quite pixelated. A reason for this may be that the volume was not broken down into sufficiently small voxels to produce high definition results, although it can be assumed that the smaller the voxel size, the greater would be the processing power required to render the image.

In addition to investigating methods for environment recreation, work based on modeling artificially intelligent characters was also examined, and several potential agent architectures were identified. Prendergast and Ishizuka (2004) outline the creation of life-like characters for uses such as synthetic actors, teammates in games and tutors. Particular attention is paid to the benefits of implementing a model of the internal emotional state of synthetic entities and applying social constraints, which it is hoped will make the characters appear more believable allowing viewers to gain a feeling of empathy towards them. They go on to describe two tools they developed that can be used to control and model the internal emotional state 'SCREAM' (SCRipting Emotion-based Agent Minds) which contains the emotional model, emotion generation, emotion regulation and importantly emotion expression. Giles et al (2003) explain the importance of expressing the emotional state. Failure to do so can make complicated processes seem like a sequence of arbitrary actions, which fails to produce a sense of empathy from the viewer. The second tool MPML (Multimodal Presentation Markup Language) is an XML

style markup language that allows characters actions to be scripted using a combination of tags. This is an important feature, as it will allow the freer scripting of characters without having to rely on pre-render graphics or set animation sequences.

Going one step further Anastassakis et al (2002) describe the mVITAL Intelligent Agent System that can be used to model an intelligent environment populated with multiple agents that interact with each other. An overall world view is maintained, and agents work on a sense-decide-act cycle. An agent can request a view of the current world state, and update the world state by performing actions. Agents do not directly communicate with each other; instead they send a special speech action to the world, which other agents pick up. To decide whether to act upon the speech action, each agent has reasoning capacity based on Belief-Desire-Intention (BDI) architecture, and an agent personality module called Virtual Agent Language (VAL). It is hoped that this method will eventually merge intelligent multi-agent systems with intelligent virtual environments to produce an integrated platform for areas such as education, presentations and entertainment.

REQUIREMENTS AND METHODOLOGY

While the scene creation techniques investigated produced some impressive results, it was considered that the amount of technical skill required to produce the scenes was so complex it would require a team of specialists, and such resources are rarely available. We preferred to have a system that would be fast, efficient and relatively cheap to use. Therefore an alternative had to be found. This was achieved with the use of a pre-created DirectX graphics engine for scene rendering, custom graphics developed in 3D Studio Max, and the development of a graphical application to orientate the objects when creating the scene. While this method is not as accurate as other techniques, it does offer some advantages. It is not always possible, or necessary, to create a photorealistic scene to evaluate and document the circumstances of a crime, and it would be very expensive to produce a virtual scene for every crime committed. In certain situations a method is required to quickly and cheaply create and render scenes. Examples of this are where the scene has been destroyed *e.g.* in the case of fire, or in areas that are in constant public use. It also appeared that the systems created previously had been over reliant on producing static photo-realistic scenes and had neglected the introduction of virtual animated characters. This is a drawback when the actions of the individuals are more important than the environment in which the incident occurred. Vital information and hypotheses can be evaluated when there are virtual characters acting out the crimes in front of the investigator rather than just motionless images. Therefore the following requirements for the system were identified:

A simple to use graphical system was required that would allow information to be gathered from various data sources to create a virtual crime scene. The system was required to render the scenes that would accept input generated from the scene creator and render it to screen. It should be possible to explore and view the scene from different angles in real

time. To show the committed crime, the rendering system should be able to display animations. A set of graphic objects that could be used within the generated scene needed to be developed. For example these should include floor, wall and ceiling tiles used to create the room as well as objects for use within the scene such as furniture, lights and plants. A set of realistic animated characters and a set of complete scenes were also needed.

The methodology used to complete the design and development tasks was based around the waterfall method. After the initial requirements were investigated, and a design introduced, the system was implemented. However, the design was not rigid, and with the use of waterfall method, the requirements and implemented system could be updated constantly, with the result affecting subsequent sections. In this way it was possible to add a degree of flexibility to the design. This methodology was implemented due to the learning curve required to investigate all the systems and applications needed for this system. It was not possible at all stages to know whether an implementation would be achievable. The waterfall method allowed the freedom to go back and update the design based on new information.

IMPLEMENTATION

The crime scene creation system implemented (Fig. 1) was developed in Visual Basic 6 and allows users to develop scenes in a graphical environment. The scene was split into several layers each containing a grid of cells that could be populated with a library of pre-created objects. There are levels for wall, floor, object and characters. Each item can be scaled and orientated in any direction. When a scene is produced, first wall panels are drawn out and scaled, and then a floor and ceiling panels are selected and positioned. Finally objects are placed within the created room. The scene creator produces a set of files containing coordinate data and a list of objects, which are loaded by the graphics engine for the scene to be rendered. Once the scene has been viewed in the graphics engine, it can be modified instantly and re-rendered as required until the user is satisfied with the results.

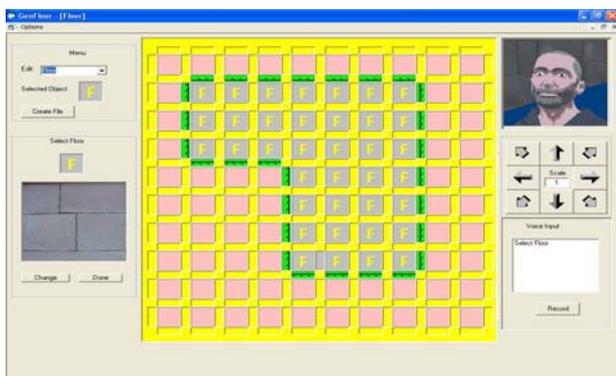


Figure 1 Crime Scene Creator

Before the implementation of intelligent agents is commenced, as an interim process, high definition characters with pre-created animation sequences were produced. According to Herda *et al* (2000), one of the greatest

challenges in computer graphics is the realistic modeling of the human body. This is because of the brain's familiarity with common objects. If a character is not animated and reconstructed accurately, the viewer will pick up on it, even if they cannot tell exactly what is wrong. Hence, the first step of the process was to develop an accurate graphical representation of a human figure.

The introduction of data collection methods such as laser scanning have made it possible to create realistic human characters to a high resolution. However, this is just a surface representation. The human body is a very complex structure, which is made up of a skeletal frame, fat and muscle and covered with skin, which stretches and folds over this volume. Algorithms that allow realistic body movement including the fluid motion of skin have been developed to add a sense of realism to models. The current standard technique based on a moving hierarchical skeleton is known as a skinned or skeletal mesh, which is the technique implemented in 3D Studio Max. Skinned meshes are based on the idea that an object's shape can be controlled by an underlying bone structure. The bones positions can be changed to alter the stance of the object, and animations can be produced via the manipulation of the position and orientation of the bone structure over a regulated time period. The surrounding layers react in an appropriate manner simulating muscle contraction and tension.

Again due to budget and time constraints it was not possible to develop a character via laser scanning techniques, although it was possible to develop a full skinned mesh with animation in 3D Studio Max following a method outlined by Steed (2002). This also has some advantages. It is possible to control the resolution of the created character. This is important, as smooth animation of high definition objects is still a demanding task on today's graphics hardware. With the use of 3D Studio Max, the created model could be optimised to remove unneeded faces to aid faster real-time rendering. Another advantage of 3D Studio Max is the ability to export models as various file types. In this case models were exported as .x files which are a native file type used in DirectX. This aided development of the graphics engine, as within the DirectX development framework there exists many libraries specific to the manipulation and animation of .x files.

The first stage of the process of creating the animated characters was to gather photographs of a real-life person that were used as a guide when modeling. The photographs were also used to make materials and textures to colour the model. Two initial sets of photographs were taken. One set of the head in isolation, and the other for the full body. Both sets were taken from head on, and from the side. Unneeded and distracting information was then removed from the bitmapped image, and the images scaled so the proportions were equal. The images were then imported into 3D Studio Max, and set as a background image. Lines were drawn over prominent features to produce a set of guide lines for the front and side of the head. The side view was then rotated through 90° and moved to the centre of the image to produce a 3D shape. With the guidelines in place, one side of the head was modeled with the use of a geodesic sphere (a

spherical object made up of 128 faces), which was placed over the guidelines and cut down the middle to produce a semi sphere. This was done as it is more efficient to only model half the head as once it is completed it can be mirrored to produce the complete head. The exosphere was manipulated to the shape of the head by moving, adding, and dividing faces. Making complex elements such as the mouth, eyes and ears proved to be very time consuming. Because of the 3D nature of the object being modeled, and only having a 2 dimensional image on the screen it can difficult to know exactly where a point is being moved. It is possible to use a multiple panel view that shows the object from four different perspectives; however, with the limited screen size it is a trade off between scale and perspective (Burford & Blake 2001).

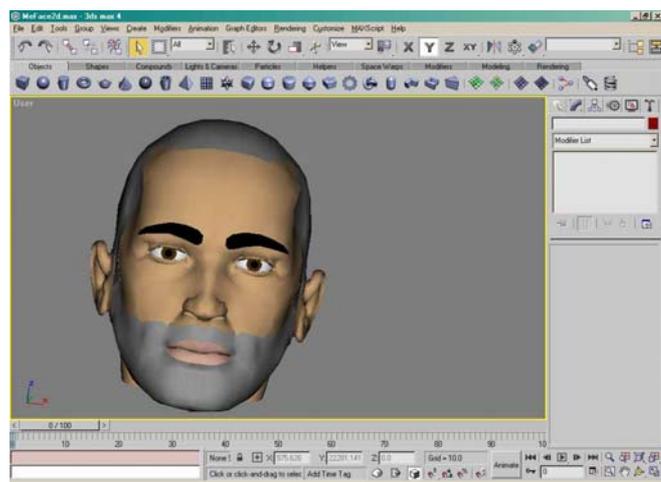


Figure 2 Character Developments

Next, the face was split into separate areas, namely the mouth, beard, eyes and eyebrows and hair. Textures generated from the original photographs were then added to appropriate areas. Finally, the left head segment was copied and mirrored, and both halves joined together to form the complete head shown in Figure 2.

Once the shell of the character was completed, a biped object was developed to act as the internal bone structure, and attached. Each point on the biped had to be positioned exactly to the corresponding bone on the original object for a successful attachment, which again is a very labour intensive task. When the character was completed, animation sets could be imported and used to animate the model by moving the internal bone structure to create fluid motion (Aubel & Thalmann, 2000).

After some initial experimentation and investigation it was decided that the rendering system used to display the scene would be developed using the graphics library DirectX 8.1. However, this is an arbitrary choice as in many ways as OpenGL is just as powerful. Even VRML would be sufficient for the rendering of the scenes. The choice was made easier with the availability of a pre-developed graphics engine created by Adams & La Mothe (2002) that was, after some modification, suitable for this project's purposes. The graphics engine works by loading all mesh and position data generated in the crime scene creator, and the graphics

created in 3D Studio Max. The resulting scene is then rendered to the screen in real time. It is possible to view the scene from any angle by moving the point of view via the use of the keyboard and mouse. It was found the most convenient method of doing this was to use the keyboard to change position (up, down, left, right, forward, back) and to change the viewing angle via the mouse. Further research is required to indicate whether this is the optimal method, but it is a control system used in many first person computer games so users may be familiar with it.

EXAMPLE

Figure 3 shows two views of a scene created in the crime scene generator, and rendered by the graphics engine. The room was composed of 8 * 8 floor and ceiling panels and 24 wall panels. 5 objects including 3 chairs, 1 wall cabinet, and a sofa. 1-point light is located in the scene to illuminate it. Finally the scene shows an animated character performing a kick.

The example shows how a scene could be developed with the use of a witness and a system expert who would utilise the software. First, measurements and photos of the scene were taken. The textures used on the floor, wall and ceiling tiles were created directly from these photos and incorporated into the system. Next, the chair and wall cabinets were modelled in 3D Studio Max. Finally, the character was developed. In this case they were made from photographs taken of a real person, although it is not always possible to get this amount of detail about suspects. It could be feasible to develop the characters in a similar way to current photo fit images, but this was beyond the scope of this project. Once the basic scene has been created, the witness of the crime could sit in with the software user and describe the witnessed crime, in this case one person kicking another person. Animations are developed until the user is happy with the sequence of events and the characters are incorporated into the scene. Finally, the scene is rendered and viewed from multiple angles – including the point of view from the witness.



Figure 3 Created Scenes

CONCLUSION AND FUTURE WORK

After researching current crime scene recreation techniques it appeared that the main aim was the creation of photo-realistic models. However, the limitations they all face are

that the scenes are static; once the scene has been developed it is difficult to alter it by adding objects without generating an entirely new model. Also, the initial process of creating the scene is a very expensive and labor-intensive task. The technology is only suitable for crime scenes that are still in existence, so that if the location has been destroyed through a crime such as arson, the current techniques would not allow the recreation of these scenes. Also, sometimes it may not be necessary to produce a completely photo-realistic model when the actions that took place there are more important than the location. Finally, and most importantly, the major disadvantage with these techniques is that the actual crimes themselves cannot be recreated and rendered. The main advantages of using the system described in this paper are that scenes can be developed very quickly and effectively, without the need for expensive and technical equipment. Using the Crime Scene Creator, it is now possible to alter the created scene rapidly by changing the components and re-rendering the scene. The original scene does not even have to exist and a scene could easily be developed from blueprints, or even the memory of witnesses. Animated characters can be incorporated into the scene, which adds an important dimension to static scenes created previously. Finally, the scenes are rendered in real time and therefore there is no need to wait while an animation sequence is pre-rendered.

The system does have some limitations. The creation of characters is a major drawback, as it requires an extensive knowledge of 3D studio max to develop them, and a level of artistic ability of the creator. An improvement would be to create the characters from within the crime scene generator. This may be achieved with the use of a photo-fit type system that allows users to select appropriate interchangeable body parts. The animation sequences are limited to a few sample actions, and it may be impossible to create a character that is able to perform any potential action described by individuals by the simple use of animation sets. No collision detection has been introduced. At present, in the interests of simplicity, the characters can wander round the scene, walking through objects, walls and other characters. The animation of characters could be improved with the use of a collision detection algorithm and some form of AI that allows characters to interact with each other. Work is currently being undertaken to produce a system similar to Prendinger et al (2002) and Anastassakis (2002) that will allow characters to become emotional, socially aware, and have the ability to reason about tasks. In this way, actions could be ascribed and acted upon, and with the use of realistic emotion expression a very realistic scene reconstruction could be produced.

In addition to character modelling, the level of detail that can be achieved in the crime scene generator is low. Objects can only be placed in a grid area of 500 * 500 pixels; also the generator only allows one item to appear in one grid area. This limits the ability to place multiple small items close together, which may be an important aspect of some crime scenes. Also, the scene creator is incapable of generating uneven terrain *i.e.* steps or slopping surfaces. A better, if more processor-intensive solution may have been to use dynamic triangular points rather than a 2D grid. This would

allow a degree of freedom in the creation of the scene; however, it would add a level of complexity to the process.

The most important enhancement that could be adopted in the future would be a Natural Language Processing (NLP) system that allows the users to explain crimes to the system in their own words to increase speed and accuracy of the scene development process. This would involve the use of a speech recognition engine, NLP, and a text-to-speech (TTS) system. The first stage would be to gain a textual representation of the spoken input to the system as Gauvain *et al* (1994) describe. However such systems are still in development. It is still not possible to implement a system that can achieve perfect recognition levels that accepts input from multiple speakers. Therefore, recognition applications have been split into two types: speaker dependent and non-speaker dependent (Lea 1980; Rowden 1992). The former achieve very high accuracy results over large grammars, but they have to be trained by the user before use, and once trained, will only work at acceptable levels for the trainer. Non-speaker dependent systems do not require users to train the system, but have the disadvantage of only supporting limited grammars and vocabulary. An attempt was made to implement this type of system into this application, with mixed results. The recognition engine used was only capable of accepting American English pronunciation and as a result, recognition levels were low and the system was not fully implemented. However, this type of system has been implemented before. Pausch & Leatherby (1990) created a system that accepted a combination of voice and mouse input to develop technical drawings, which achieved an average of 21.23% speed increase over mouse and keyboard input. The system developed would have to be more sophisticated than a simple command entry system, as a level of understanding would be required. This would be in the form of an NLP application, which would attempt to gain an understanding of naturally spoken text. As the users would enter data in their own words, a level of complexity is added. Spoken text differs in many ways from written text, as it does not always comply fully with grammatical rules. Methods need to be investigated as to which NLP system would be most appropriate. Finally, a TTS system such as that described by IBM (2000) could be introduced to take speech in a textual format and output a synthesised natural sounding voice. To make the speech sound as natural as possible IBM uses a system called Concatenative TTS, that has a dictionary of unique phonemes. When the speech is generated in the synthesiser, the appropriate phonemes are chosen and combined (concatenated) according to linguistic rules generated by a text pre-processor. Theoretically, it is possible to produce any phrase using only these sounds and rules. In this way, the system could be adapted so the characters performing the crimes could also have a voice

REFERENCES

Adams, J. and La Mothe, A. 2002 *Programming Role Playing Games with DirectX*, USA: Premier Press.

- Anastassakis, G. Panayiotopoulos, T. Ritchings, T. 2002. *Virtual Agent Societies with the mVital Intelligent Agent System* Knowledge Engineering Lab, Department of Informatics, University of Piraeus, Piraeus, Greece
- Aubel, A. and Thalmann, D. 2000 Realistic Deformation of Human Body Shapes, *Proc. 11th Eurographics Workshop on Computer Animation and Simulation*, Interlaken, Switzerland.
- Burford, D. and Blake, E. 2001 *Real-Time Facial Animation for Virtual Characters*, Collaborative Visual Computing Laboratory, Department of Computer Science, University of Cape Town.
- Dyer, C. 2001 *Volumetric Scene Reconstruction from Multiple Views*, University of Wisconsin, 469-489.
- Gauvain, J., Lamel, L., Adda, G. & Adda-Decker, M. 1994 *Speaker-Independent Continuous Speech Dictation* cedex, France.
- Giles, M., Ballin, D., Dodsgon, N., 2003 *Intergrating internal behavioural models with external expression* University College London in Partnership with BTexact Technologies, Ipswich
- Herda, L., Fua, P., Plankers, R., Boulic, R. & Thalmann, D. 2000 *Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion*, Computer Graphics Lab (LIG) In: *Computer Animation*, Philadelphia, PA. <http://citeseer.ist.psu.edu/article/herda00skeletonbased.html>.
- Howard, T. and Murta, S. 2000 *Virtual Environments for Scene of Crime Reconstruction and analysis* Advanced Interfaces Group, Manchester.
- Hu, B. and Brown, C. 2002 *Interactive Indoor Scene Reconstruction From Image Mosaics Using Cuboid Structure* The University of Rochester Computer Science Department.
- IBM. 2000 *Introduction to TTS*.
- Lea, W. 1980 *Trends In Speech Recognition* 1st ed. New Jersey: Prentice-Hall, Inc.
- Little Q., Small, E. and Ralph, R. 2000 *Forensic 3D scene Reconstruction*, Sandia National Laboratories.
- Murta, A., Gibson, S., Howard, T & Hubbard, W. 2000 *Modelling and Rendering for Scene of Crime Reconstruction: A Case Study*, Advanced Interfaces Group, University of Manchester, Department of Computer Science.
- Pausch, R. and Leatherby, J.H. 1990. A Study Comparing Mouse-Only Input vs. Mouse-Plus Voice Input for a Graphical Editor, *Proc. AVIOS '90 Voice I/O Systems Applications Conf.*, 227-231
- Prendinger, H. and Ishizuka, M. 2004. *Introducing the Cast for Social Computing: Life-Like Characters*, Department of Information and Communication Engineering, Graduate School of Information Science and Technology, University of Tokyo
- Prendinger, H. Descamps, S. Ishizuka, M. 2002. *Scripting the Bodies and Minds of Life-like Characters*, Department of Information and Communication Engineering, Graduate School of Information Science and Technology, University of Tokyo
- Rose, C., Bodenheimer, B. & Cohen, M.F. 1998 Verbs and Adverbs: Multidimensional Motion Interpolation Using Radial Basis Functions, *IEEE Trans. Computer Graphics and Applications*, 32-40.
- Rowden, C. 1992 *Speech Processing* 1st ed. Maidenhead: McGraw-Hill.
- Sequeira, V., Gonçalves, J. & Ribeiro. I. 2001 *3D Reconstruction of Indoor Environments* European Commission, Joint Research Centre, Ispra (VA), Italy.
- Steed, P. 2002. *Modeling a Character in 3DS Max*: Texas: WordWare Publishing.

**COMPUTATIONAL
MODELLING
AND
SIMULATION
IN
SCIENCE
AND
ENGINEERING**

LEACHABLE GEOMETRY

Christian Körber, Dietmar P.F. Möller
Fachbereich Informatik
Universität Hamburg
Vogt-Kölln-Straße 30, 22527 Hamburg,
Germany
E-Mail: koerber@informatik.uni-hamburg.de

Caroline Hanusch
Institut für Markscheidewesen
Technische Universität Clausthal
Erzstraße 18, 38678 Clausthal-Zellerfeld,
Germany
E-Mail: Caroline.Hanusch@tu-clausthal.de

KEYWORDS

Computational Geometry, Salt Leaching, Dynamic Geometry, Hydro-Geochemistry

ABSTRACT

Most geoscientific problems have a geometrical component respective representation. Depending on the requirements of the problem which need to be modeled, an optimal geometrical model needs to be chosen. Then, several properties have to be taken into account: differentiability, regularity, modifiability, controllability, extendibility and the possibility to attach a physical model. A geometrical model which supports all of these properties is yet to be invented. While one model for instance might support differential geometry very well, it may lack the ability to interpolate complex geometry, which another model does, while lacking in turn differentiability. This paper summarizes the properties of the most common geometrical models for solid geometry in 3D in the context of the modeling of the hydro-geochemical process “salt leaching in flooded potassium mines”. Hereby emphasis is placed on model topology and model dynamics. Additionally, consequences for geometrical modeling due to the fundamental differences between geochemical and physical based modeling are pointed out.

INTRODUCTION

The subject of this paper is related to the interdisciplinary (comprising Geo-Technology and Computer Science) project cluster *Development and Application of ICT-based Methods for the Impact Analysis, Prognosis and Control of anthropogenic influenced Processes in Geosystems* supported by the DFG (German Research Foundation).

Geoscientific background for the described investigations is an already for decades lasting underground salt leaching process in the area of Stassfurt/Germany. There, potassium bearing salts have been mined since the 19th century, resulting in numerous underground cavities, which have been filled with water since then (Schwandt und Seifert 1999), inducing a still ongoing leaching process.

The salt deposit has a layered structure (figure 1) where alternating more or less potassium bearing salt rock

layers appear (Knak 1958). Since salt rocks of different composition shows different leaching characteristics, they necessarily have to be distinguished in a corresponding geometrical model.

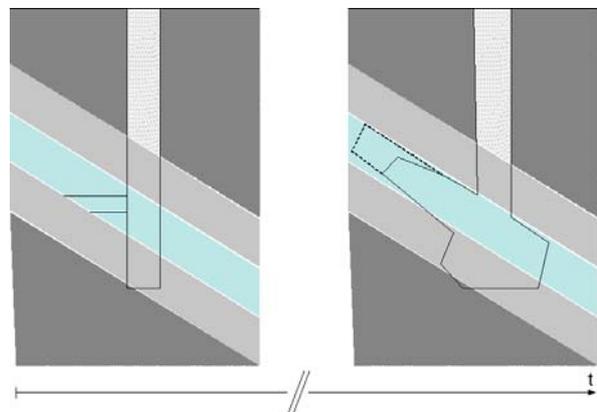


Figure 1 Salt leaching, Stassfurt, Germany

The left subfigure shows three different salt rock layer and the mining shaft, the right subfigure shows additionally the growing brine body.

Characteristic for potassium bearing salt is that not just salt is leached resulting in some kind of salty water (brine). In fact a circulation process occurs, while certain components become leached, others drop out (Sander 1988) and accumulate at a lower level, actually masking the leaching process in that area. The composition of the brine constantly changes over time while interactions constantly take place between salt rock and solution.

These dynamic interactions can be localized along the reaction surface between brine (fluid) and rock (solid), more basically between objects with different geochemical attributes. The direction and velocity of the solution process can be described by vectors, determined by an underlying process model, which integrates the relevant parameters of all involved objects (rock, fluid, reaction surface).

Thus, basic requirements for a geometrical model (the term model refers to the two- and three-dimensional geometrical models depicted in this paper) being

capable to represent the described features are as follows:

- many complex bodies
- dynamically altering objects
- differential geometry on the reaction surface
- interactions/interdependencies between objects
- topology preservation (no self penetration etc.)

The Geo Information System BAGIS and its successors (Kesper and Möller 1999), developed at the chair for Computer Engineering at the University of Hamburg, so far employed static parametric surfaces and solids (Körber et al. 2003), which are optimal for visualization and differential geometry. Since fundamentally altering parametric models is hard, as is preserving topology, the project group started to re-evaluate alternative dynamic solid models. This is recaptured on the following section. After that, the commonly used approach for dynamic geometric processes, physical based modelling, is considered with respect to the salt leaching process. This will be followed by a description of easily occurring, but hard to remedy topological entrapments. Finally an approach using a combination of voxel representation and parametric linear interpolation is suggested.

TYPES OF GEOMETRICAL MODELS

Geometrical Models can be partitioned into 5 broad classes (McDonnell 2000):

1. implicit Geometry
2. Constructive Solid Geometry (CSG)
3. parametric geometry
4. subdivision models
5. cell decomposition

Implicit Geometry defines geometry as the solution to an equation like $x^2 + y^2 - 1 = 0$ which defines the unit circle. This is, while mathematically exact, computational expensive and requires supportive algorithms such as the *marching cube* (Lorensen and Cline 1987) to be handled halfway efficiently. It is also very hard to find an implicit expression for complex geometry defined by samples. Finally, differential analysis is generally not possible on implicit functions unless converted to a parametric or an explicit form.

Constructive Solid Geometry (CSG) builds solids by applying set (Boolean) operations on primitive forms. This approach is strongly related to machines for metal processing for which it was invented. It is obviously unsuitable for geometry which is hard to combine from primitives and also inappropriate for direct differential analysis.

Parametric models are vector valued functions defining their shape by a regular grid of control points (Piegl and Tiller 1997). Initially developed for construction and

especially for design (automotive engineering, aircraft construction and shipbuilding) Bézier-, B-Spline- and NURBS-models are meanwhile well elaborated tools. Beside shape modification, parametric models stand out by their differentiability due to the closed form basis functions (polynomials) of whom they consist. Nevertheless, we found it very hard to reconstruct real world geometry from samples, especially for solids, as known algorithms work only for surfaces since they require a projection onto a plane (Hormann and Greiner 2000) which is impossible for solids. Additionally, modifying the control point grid, e.g. to adapt it to increasing model complexity, is complicated and, especially in 3D, inefficient, since the regular grid structure must be maintained. Reconstruction is also hampered by the regular grid structure, which often does not fit well to asymmetric and complex geometry.

While the former three classes have a mathematical form, the remaining two are based on algorithms.

Subdivision models (Catmull and Clark 1978) are based on the successive refinement of an initial arbitrary grid. They converge, dependent on their subtype, to biquadratic resp. bicubic B-Spline representations. Because they lack basis functions, they don't share the mathematical abilities of parametric models. Thus, computing differential properties such as the derivatives, requires an indirection. Unlike parametric models, subdivision models don't depend on a regular control point structure, which makes them suitable for complex and asymmetric geometry. Because of the structure of the algorithm they also have built-in level of detail (LOD) abilities (Hoppe 1998).

Cell subdivision finally completely discretizes the space into small regular units. Most common subtypes are voxel and octrees. Voxel decompose space into regular units, usually cubes. They obviously require n^3 space which is their greatest drawback, limiting model size and complexity more than any other model to the amount of available memory. Another suboptimal property is rendering which usually requires determining the boundary of the modelled object. Both can be overcome by employing another cell decomposition subtype, octrees, who decompose space hierarchically and uses fine resolution only for the objects boundary, allowing fast access to the boundary and decreasing memory consumption. Cell decomposition is commonly used in medical science, e.g. for magnetic resonance tomography (MRT) when the inside of an object is at least as interesting as its boundary. Both voxel and octrees suffer necessarily from their discrete structure, which can be very obvious, depending on the scale.

PHYSICAL BASED MODELING

The motivation of the attachment of a physical model to a geometrical model and thus deforming geometry dynamically, was derived from two reasons. One is to make models easier to handle, controlling them by forces the user experiences in everyday life. The other reason is to actually model a physical process on spatial objects.

The several degrees of freedom (control points, knot vectors, weights) parametric models provide, complicate controlled model modification, which is especially significant for NURBS. This resulted in the demand to apply real physical forces, e.g. pressure, on the model in order to achieve the desired deformation. (Terzopoulos 1987). Another demand came e.g. from medical sciences, asking for realistically behaving models which could be employed in real time for surgery training (Wu et al. 2001).

This behaviour can be achieved by employing Finite Element Methods (FEM) on parametric and subdivision models. This approach is called physical based modelling and is quite self-evident, since the geometric structure (triangular resp. quadrilateral grid) of the models corresponds well with a FEM polygon mesh generation.

However, the underlying physical laws are based on mechanics, i.e. the model is deformed based on e.g. pressure and tractive forces and using material properties like viscosity and elasticity.

Hence models can obviously be deformed by physical based modelling, but not altered in a more fundamental manner.

The salt leaching process is rather taking away something (the salt) from an object (salt rock) and adds it to another object (brine), than deforming objects. The salt leaching process could perhaps be mimicked by one object (the brine) applying pressure on another (salt), but this would require successively increasing forces in order to increasingly compress the salt body and feels generally inappropriate.

PRESERVING TOPOLOGY

If the model of an object consists in fact of several smaller models, like the salt leaching area, topology problems may arise once that the carefully constructed model becomes subject of forces whose exact impact on the model is either unknown or not efficiently pre-computable. That means that preserving the topology of a robot arm, composed from several parts, is controllable, because all parameters are known and user or computer controlled, which makes it easy to detect e.g. self penetrations or even better, to prevent them in the first place. Figure 2 on the other hand illustrates exemplary cases which may arise if even simple 2D polygon models start to change their shape according to an underlying complex process. The figure shows schematically the reaction front between two salt layers and the brine body. One of the salt layers is hard and one is easy to leach which results in different leaching

rates, which prevents to treat the layers as one. The two top subfigures show how the top layer penetrates the lower layer because the movement of the reaction front, induced by the leaching process, is modelled as the compression of a polygon model, see the preceding section. The lower subfigures show how insufficient area coverage may result from a geometry which becomes increasingly complex. The brine body in the lower right subfigure does not have enough vertices and edges to fill the space which results from the leaching.

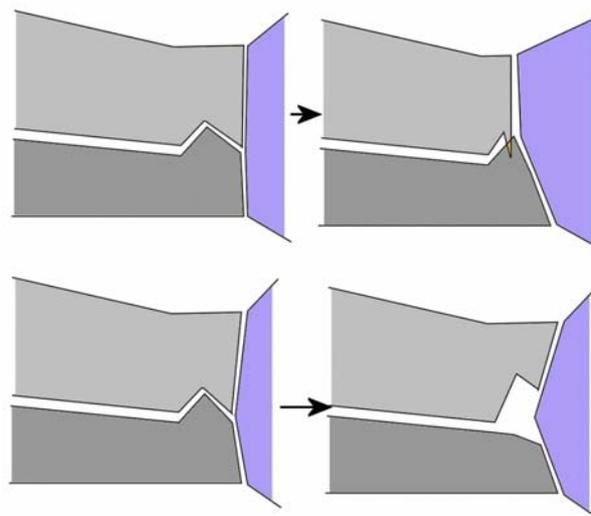


Figure 2 Illegal Topology due to Dynamic Geometry top and bottom left: initial position, top right: self-penetration, bottom right: insufficient area coverage

Such issues could be partially overcome, though often for the price of inducing constraints and limitations. In the first example e.g. vertices could be moved along edges. This would require a regular grid to work, which is quite a constraint as described in the preceding sections.

CONCLUSION AND PERSPECTIVE

None of the evaluated geometrical models optimally meet the requirements of the salt leaching process. While implicit geometry and CSG never really were candidates, also subdivision and parametric models come at a high price. It appears questionable whether the easily differentiable structure of parametric models or the arbitrary grid structure of subdivision models, justify the hassle expected from maintaining legal topology due to dynamic topology.

That brings cell decomposition into the focus, which was originally declined because this approach doesn't fit to BAGIS' Data Model (Kesper 2001), which would have to be extended.

Nevertheless, cell decomposition fits well to the hydro-geochemical process as one cell can simply switch attributes from salt to brine without bringing topology into any trouble. One issue which had to be dealt with is

that the reaction surface moves very slow, perhaps 1cm per cycle of the underlying process model, which would then be the required resolution for e.g. voxel. We currently favor a model which combines cell decomposition and parametric properties by linking attributes not to voxel but to a regular grid of control points between whom we linearly interpolate. This allows a finer transition between control points / voxel without requiring more memory. Formally this is a linear solid B-Spline but since the control points lie on a regular grid, and the geometry thus implicit, the similarities to voxel are obvious. First test in 2D seem to confirm our expectations. Figure 3 shows a mimicking (no process model is used) of the salt leaching process, which does not show the hard edges which are typical for voxel.

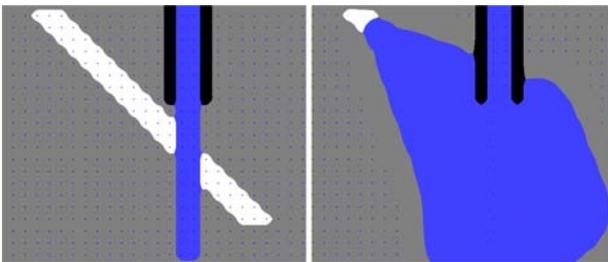


Figure 3 bilinear interpolating 2D cell decomposition of the investigation area

Some issues, like embedding several objects in one geometrical model, identifying the reaction surface and deriving its differential properties still need to be handled, but are considered easier to be handled than the mentioned topological and process related disadvantageous properties other models imply.

REFERENCES

- Catmull, E. and Clark, J. 1978. "Recursively generated B-spline surfaces on arbitrary topological meshes". In *Computer Aided Design*, September 1978, 350-355
- Hoppe, H. 1998. "Smooth view-dependent level of detail control and its application to terrain rendering". In *IEEE Visualization '98*. D. Ebert, H. Hagen and H. Rushmeier (Eds.). 35-42
- Hormann, K. and Greiner, G. 2000. „Quadrilateral Remeshing“. In *Proceedings of Vision, Modeling and Visualization 2000*. Saarbrücken, 153-162
- Kesper, B. and Möller, D.P.F., 1999. "BAGIS – Baugeologisch-Geotechnisches Informationssystem: Informationstechnische Realisierung". In *Raumbezogene Informationssysteme für geologische, bau- und geotechnische Aufgaben 1999a, 1. Clausthaler FIS-Forum*. G. Reik and W. Paege (Eds.). Papierflieger, Clausthal-Zellerfeld, 57-66
- Kesper, B. 2001. "Konzeption eines Geo-Datenmodells unter Verwendung von Freiformkörpern auf Basis von Volume Non Uniform Rational B-Splines", *Dissertation*, University of Hamburg
- Körber, C.; Möller, D.P.F.; Kesper, B. and Hansmann W. 2003. „A 4D Modeling Concept using NURBS Applied to Geoscience“. In *4th Mathmod Proceedings, Volume 2*, I.Troch (Ed.). ARGESIM, Vienna, 583-591
- Knak, I. 1958: *Feinstratigraphische Aufnahme des Kalilagers Staßfurt auf den Schachtanlagen Neustaßfurt VI/VII, Berlepsch-Maybach und Ludwig II*, Diplomarbeit Bergakademie Freiberg.
- Lorenson, W.E. and Cline, H.E. 1987. "Marching cubes: A high resolution 3D surface construction algorithm". In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques Computer Graphics*. ACM Press, New York, 163-169.
- McDonnell, K.T. 2000. "Dynamic Subdivision-based Solid Modeling". Research Proficiency Examination. Department of Computer Science, Stony Brook University, New York
- Piegl, T. and Tiller, W. 1997. "The NURBS Book, 2nd Edition". Springer Verlag, Berlin und Heidelberg
- Sander, W. 1988: *Quantitative Beschreibung der Lösungs-metamorphose beim Eindringen von Wasser in ein Bergwerk im Zechsteinsalinar.- Kali und Steinsalz*, Band 10, Heft 2, 54-61.
- Schwandt, A. and Seifert, G.: *Natürliche und gelenkte Flutung von Salzbergwerken in Mitteldeutschland*. In: *Kali-, Steinsalz und Kupferschiefer in Mitteldeutschland: Exkurs. f. Veröf. GGW 205: 61-72*; Berlin.
- Terzopoulos, D.; Platt, J.; Barr, A. and Fleischer, K. 1987. "Elastically deformable models". In *SIGGRAPH 87 Proceedings, Vol. 21 No. 4*
- Wu, X.; Downes, M.S.; Goktekin, T. and Tendick, F. 2001. "Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes" In *Eurographics 2001, Vol. 20, No. 3*,

AUTHOR BIOGRAPHIES



CHRISTIAN KÖRBER was born in Hamburg, Germany and enrolled at the University of Hamburg studying Computer Science. He obtained his MSC degree in 2002. He had been hired as research assistant from the chair of Computer Engineering and worked on the topics Geo Information Systems, <http://www.verve-gis.com>, with emphasis placed on computer graphics and geometric modeling. His e-mail address is koerber@informatik.uni-hamburg.de and his Web-page can be found at



CAROLINE HANUSCH was born in Bad Harzburg, Germany and went to the Technical University of Clausthal, where she studied Geology and obtained her degree in 1999. In 2001 she started working at the Institute of Geotechnical Engineering and Mine Surveying at the Technical University of Clausthal (<http://www.igmc.tu-clausthal.de>) where she since then works on a Geo Information System (<http://www.verve-gis.com>), emphasizing Geochemistry and Process Modeling.



DIETMAR P.F. MÖLLER was born in Preetz, Germany and studied at the Universities of Mainz, Bonn and Bremen, where he studied Electrical and Control Engineering. He obtained his doctoral degree in 1980 and in 1985 a degree in Medicine. From 1985 to 1991 Dr. Möller lead the anaesthesia division of Dräger AG in Lübeck, Germany. 1991 he became Full Professor for Computer Science at the Technical University of Clausthal. In 1998 he was elected from the University of Hamburg as Full Professor and Chair of Computer Engineering. Since 1998 he also holds the position as Adjunct Professor at the California State University Chico. His E-Mail address is dmoeller@informatik.uni-hamburg.de and his Web-page can be found at <http://www.informatik.uni-hamburg.de/TIS/>

Evaluation of Methods for the Process Modeling of Salt Leaching Processes

Christian Zemke, D.P.F. Möller
Universität Hamburg – AB TIS
Vogt-Kölln-Str. 30
D-22527 Hamburg, Germany

E-mail: ABTIS@informatik.uni-hamburg.de

Klaus Maas
TU Clausthal - IGMC
Erzstr. 18
D-38678 Clausthal-Zellerfeld,
Germany
E-mail: IGMC@tu-clausthal.de

KEYWORDS

Rule based system, fuzzy reasoning, neural networks, salt leaching process.

ABSTRACT

This paper introduces different methods in order to support the understanding and prognosis of dynamic salt leaching processes in a flooded potash shaft. Here geochemical computations as well as kinetic boundary conditions became part of the regarded data base. The process model has to consider the partly unknown geometrical conditions as well as the complexity of the geochemical processes. The complex intercorrelations of effects in the regarded geosystem make it necessary to include all available information. This requires the discussion of several methods of pre-processing and analysis. This survey will contribute to determine sustainable treatments of possible reclamation in the Stassfurt mining area.

INTRODUCTION

The subject of this paper is embedded in a framework which combines GIS (Geographic Information System) specific characteristics with a non numeric simulation approach. This framework is based on the interdisciplinary project cluster *Development and Application of ICT-based Methods for the Impact Analysis, Prognosis and Control of anthropogenic influenced Processes in Geosystems* supported by the DFG (German Research Foundation).



Figure 1: Location of the investigation area

The investigation area is an old mining shaft within the worldwide oldest potash mining district, near to the city of Stassfurt, Germany (fig. 1).

MINING AREA STASSFURT

The area of Stassfurt had profited from saline mining for hundreds of years. Since 1861 potash was extracted in deep mining systematically whereupon a part of the underground plant was situated underneath the urban area. A part of the cavities area was drowned unintentionally in 1878 to 1922 by fresh water, other parts were still active until 1972. The shaft VI, which is in focus of our investigations, was flooded with brine from 1975 to 1979 (fig. 2). The flooding and leaching process led to a brine filled cavity volume of approx. 20 million m³. The surface is affected till today by:

- sudden fallen sinkholes,
- a slightly active subsidence,
- a daily drain of approx. 1000 m³ to avoid a partial flooding of the city by groundwater,
- endangerment from partial highly toxic neglected deposits within the wetting area.

In consequence of the subsidence processes over 850 buildings in the urban area were torn off. The situation for the city of Stassfurt and the region is hardly controllable. The urgency of sustainable geotechnical measures to break the self stimulating cycle of drainage, fresh water inflow and the resulting leaching process is evident (BUSCH ET AL. 2003).

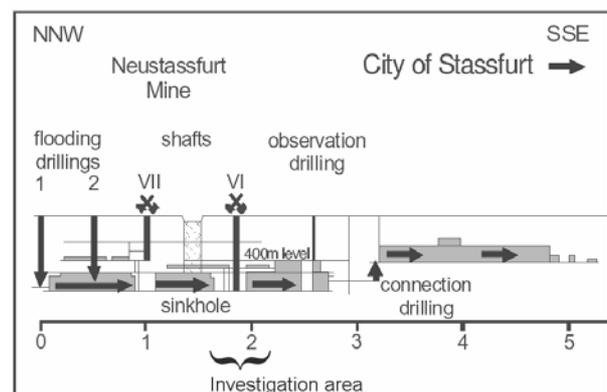


Figure 2: Details of the investigation area

A special feature in the Stassfurt district was the mining of carnallite, which could be used directly for agriculture purposes. Therefore, numerous excavations exist, whose pillars have a high grade of carnallite, which stimulates the leaching process by high leaching ability. Carnallite is leachable, even by saturated NaCl brine. In order to be able to reconstruct the leaching process accurately, a model under consideration of the mining situation and all mechanical and hydrologic parameters is necessary.

The local mining authority considered several approaches of analytical and numerical calculations (LUX ET AL. 1999). They took into account parameters like chemistry, density, direction of flow, discharge position, water elevation places, convection cells, convergence, as well as natural leaching processes. On the basis of these parameters, hydrological and geochemical monitoring measures were assigned. The results of these measurements refined the model of the leaching process, but still no evaluation was possible, in which way the leaching process will be affected by stopping the drainage. Until now, no plausible prognosis can be given about the behaviour of the induced leaching process (FISCHER UND MAAS 2003).

Motivated by these circumstances we considered to make a new approach of a spatial and time related model of the leaching process, which should be able to provide numerical calculations of stress conditions leading to subsidence prognosis with more reliable boundary conditions. This approach is based on hydrogeochemical computations with stochastic preprocessing strategies combined with a rule based system and a 3 dimensional representation of which certain surface properties can be derived and included in the thermodynamic balances.

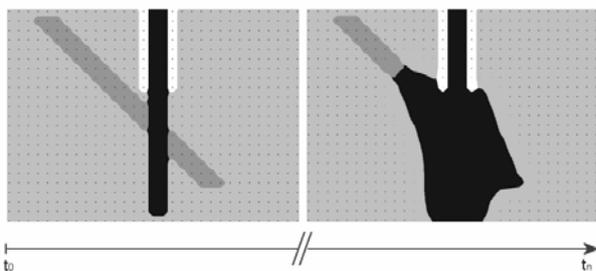


Figure 3: Sketch of the leaching process in a carnallite layer.

THE ENVIRONMENT

This chapter gives a short overview about the environmental settings. This will lead to the necessity of a brief parameter discussion with a main focus on preprocessing strategies.

Geological Settings

The Stassfurt mining district is part of the subhercynian axial depression, which contains the Zechstein

formation up to Kainozoic sediments. The saline Zechstein formation consists of six saline cycles (z1 to z6). Within the investigation area the mineable potash beds belong to the Stassfurt cycle (k2/z2). This cycle is developed in layers of carnallite, showing a thickness up to 45 to 55m. Near the axes of the NNW-SSW striking Stassfurt-Oschersleben anticline, the layers dip in an angle of ore than 80° NE. The anticline is covered discordantly by anhydrite and only a few meter of quaternary sediments.

Hydrogeological Settings

Within the flooded underground mine sites and cavities, quantity and quality of the circulating brine is widely unknown. Furthermore a sinkhole, fallen in 1975, caused a hydraulic contact between the underground mining cavities and surface- /groundwater. Therefore, the hydraulic system is an open system and any calculation of soluble and transported compounds is based on assumptions that are difficult to determine.

Hydrogeochemical Settings

Thermodynamic Balances

The solution processes of salinar rocks can be reconstructed by chemical-thermodynamic equilibrium computations (WOLERY 1992) on the assumption of certain boundary conditions. The assumption of a closed system is indispensable regarding the transferred chemical species (masses of soluble) as well as the quantity of the brine (law of continuity). In an open system appropriate boundary conditions have to be estimated and/or replaced by heuristics.

Mechanical Effects

Moreover mechanical aspects affect the underground solution process. Continuous dissolution and recrystallization cause a changing structure of the salt rock surface. While crystallization processes lead to a partial sealing (masking) of the surface, the sedimentation of insoluble components withdraw parts of the salt rock surface from the solution progress (blocking). In consequence of these processes the proportion of brine volume and salt rock surface varies and therefore crucially influences the development of the entire solution process.

Kinetic Effects

Kinetic parameters (as well as mechanical) are so far barely considered in available geochemical computer models or they are only integrated indirectly. Kinetic parameters could be derived from the pitch of the contact surface between the solid and the liquid phases. This angle affects local convection cells. These cells should be determined as solution cells, which act homogeneous in the chemical-thermodynamic equilibrium computations. Knowing these cells it becomes also possible to assign a certain expanse of the contact area that is necessary in a balance computation.

NECESSITY FOR NEW APPROACHES

The necessity for sustainable geotechnical measures in the Stassfurt district is obvious. The expected ground movements and sinkholes as well as the associated flooded areas do not only threaten buildings and infrastructure. The numerous dump locations in the potential flooded areas endanger also the health of the inhabitants. The determination on a remediation concept is difficult, because of the lack of knowledge and the complexity of the in situ mine damage. Conventional approaches for the modeling of the causative leaching process fail due to fuzzy parameters and undetermined boundary conditions. The main dependencies, that are not included in common geochemical computer models, are the reciprocations between metric, topologic and thematic issues. With the interdisciplinary research project described above, the feasibility of new approaches is evaluated. This takes place first for a limited investigation area. The methodical approach could be subsequently extended to wider areas of the excavation.

SUGGESTED MODELLING TECHNIQUES

In general the salt leaching process can be reconstructed by chemical-thermodynamic equilibrium computations. Those models provide mass and volume balances between the fluid (brine) and the solid (rock) phases with a certain compound.

The salt deposit in the investigation area has a layered structure (fig. 4) where alternating more or less potassium bearing salt appears. Since salt rocks of different composition shows different leaching characteristics, they necessarily have to be distinguished in a corresponding process model. Especially layers with a high grade of carnallite, which stimulates the leaching process by high leaching ability, could not be described as a homogeneous area while that layer itself has varying attributes that influences the leaching process (MAAS 2001).

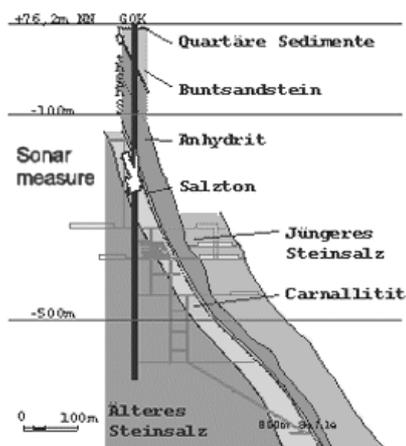


Figure 4: Profile of the salt deposit in the investigation area

Pre-processing

Those mentioned attributes that affects the leaching process can be characterize as

- amount of insoluble components
- fraction of chemically combined water
- porosity, permeability
- rock temperature, brine temperature
- rock stress
- viscosity of salt

Chemical components of the carnallite layer are known from detailed lithological profiles, that can be assigned to the investigation area (fig. 6).

Data pre-processing should lead to spatial clusters that act homogeneous in chemical-thermodynamic equilibrium computations. Different techniques of cluster analysis are well known as a fundamental operating principle in statistics. Thereby it is imperative that the given dataset suites to certain pre-requisites.

A neural based cluster analysis can be performed by unsupervised network topologies such as self organizing feature maps (SOM). The main advantage of this method is that in case of non-linear relations between parameters no standards of equation systems must be assumed or determined by iterative loops. Furthermore no limitations in reference to the amount of given samples or their distribution must be considered. The basic algorithm follows certain learn and optimization strategies that approximate separation functions autonomously. The result of a neural based cluster analysis by SOM can be represented as patterns of activity on a 2 dimensional maps of interconnected neurons (fig. 5). Correlating parameter will show a similar activity, whose significance must be determined by experts knowledge (KOHONEN 1997).

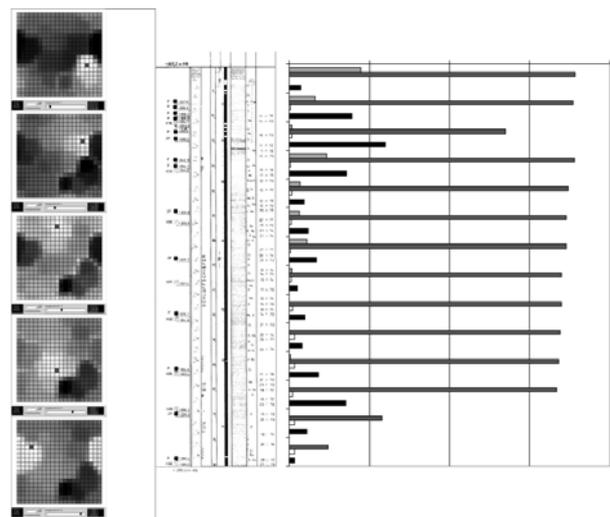


Figure 5: SOFM based cluster analysis of borehole sections (ZEMKE 2003)

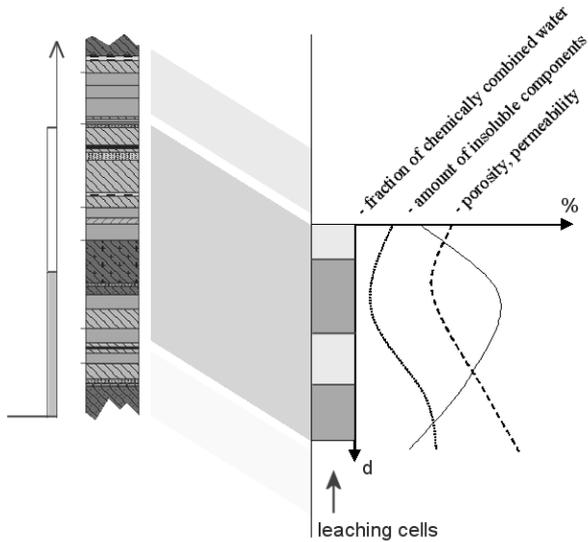


Figure 6: Pre- processing of data, detection of leaching cells (schematic sketch)

Rule Based Systems

Assuming that we have determined the leaching cells by cluster analysis, we can compute the mass-volume exchange between fluids and solids by chemical-thermodynamic equilibrium balances. But still no kinetic or mechanic influences are involved in the leaching simulation. Our approach tries to combine a rule based systems with geochemical computer models in a GIS architecture.

Fuzzy Based Approach

The first association that occurs when you think of rule based systems is a fuzzy based approach. Fuzzy systems provide simple strategies to rule complex data structures by linguistic statements (MÖLLER 2003). The regulator consists of a specification of the situation (IF – part) and its resulting reaction (THEN – part). The IF – part is composed by one or more conditions, which are combined logical by operators (AND / OR). The evaluation of the definition of the situation is realised by an inference operator that checks for the degree of performance of the defined fuzzy rules. Finally combining operators lead to conclusions about the system behaviour by evaluating the respective conditions (fig. 6).

The specific Impact analysis is defined by an adequate appending or deleting of rules. To support this complex incident, the content of truth of each rule can be modified by an association factor. This value associates a factor to each rule, which can be used as an individual weighting for certain criteria. It can also be used for the adjustment of rules while calibrating the process model. Several possibilities exist to link self learning algorithm to this manipulating factor (GRIMA 2000).

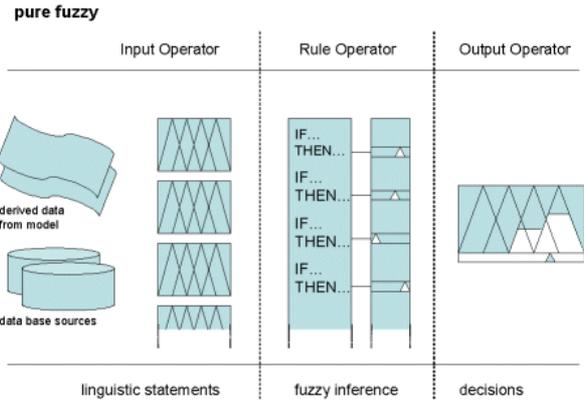


Figure 7: Simplified structure of the fuzzy rule based system.

Alternative Approaches

As an alternative to a fuzzy based rule system our current research is evaluating neural network based algorithms. *Causal Networks* or their increments *Bayes Networks* describe miscellaneous applicable structures for the compact processing of complex data. They are often used in context of expert systems (JENSEN 1996). Finally neural networks with a supervised learning algorithm such as *Backpropagation (BPN) Networks* could be adopted, if the impact structure of a system is generally known (fig. 8). Those algorithm have been successfully tested in other environments (ZEMKE ET AL. 2003).

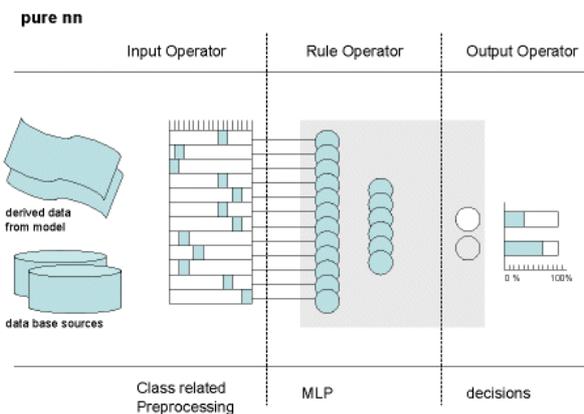


Figure 8: Simplified structure of the BPN rule based system.

EXPECTED RESULTS

This paper can not reach a conclusion while the mentioned research topics aren't sufficiently compiled yet. But the combination of the suggested ICT methods should lead to results, that help to detect sustainable geotechnical measures in the Stassfurt district.

REFERENCES

- BUSCH, W., HANUSCH, C., KNOSPE, S., MAAS, K.: (2003a): *A time and space continuous model of salt leaching processes in a flooded potash mine.* – In: I. Troch, F. Breitenegger (Hrsg.), Proceedings 4th MATHMOD Vienna, Feb. 2003, ARGESIM-Report 24, Vol. 1, S. 94 und Vol. 2 (CD), pp. 592-597, ARGESIM-Verlag, Wien.
- FISCHER, C., MAAS, K. (2003): *Modellierungsansätze für die räumlich-zeitliche Abbildung geokinematischer Prozesse.* – In: Sroka, A.; Wittenburg, R. (Hrsg.): 4. Geokinematischer Tag, 15.-16. Mai 2003, Schriftenreihe des Institutes für Markscheidewesen und Geodäsie an der TU Bergakademie Freiberg, H. 2003-1, S. 183-191, ISBN 3-7739-1504-7, Verlag Glückauf, Essen.
- GRIMA, M. A. (2000): *Neuro-Fuzzy Modelling in Engineering Geology.*- 244; Rooterdam (Balkema).
- JENSEN, F. (1996): *An introduction to Bayesian Networks.*- Springer (Berlin).
- KOHONEN, T. (1997): *Self-Organizing Maps.*- 551; Berlin (Springer).
- LUX, K.-H., HOU, Z., DÜSTERLOH, U. (1999): *Neue Aspekte zum Tragverhalten von Salzkavernen und zu ihrem geotechnischen Sicherheitsnachweis, Teil1: Theoretische Ansätze, Erdöl, Erdgas, Kohle,* 115. Jg., Heft 3, S. 122-127, 1999. Teil 2: Beispielrechnungen mit dem neuen Stoffmodell Hou/Lux, Erdöl, Erdgas, Kohle, 115. Jg., Heft 4, S. 198-206.
- MAAS, K. (2001c): *Analysis, Modelling and Simulation of recent Subrosion in a flooded Potash Shaft.* – Paper presented at Fall 2001 Meeting of the Solution Mining Research Institute (SMRI), published under www.solutionmining.org. Albuquerque, New Mexico U.S.A.
- MÖLLER, D.P.F. (2003): *Mathematical and Computational Modeling and Simulation.*- 478, Heidelberg (Springer).
- WOLERY, T. J. (1992a): EQ3/6, *A Software Package for Geochemical Modeling of Aqueous Systems: Package Overview and Installation Guide (Version 7.0)*, UCRL-MA-110662-PT-I, Lawrence Livermore National Laboratory, Livermore, California.
- ZEMKE, C. (2003): *Methodik und Anwendung neuronal-basierter Klassifikatoren in der Analyse komplex wechselwirkender Prozesse in Geosystemen.*- Diss., TU Clausthal, Clausthal-Zellerfeld, 132 S.
- ZEMKE, C., REIK, G., MÖLLER, D.P.F. (2003): *Neural based classifiers applied to complex geosystem analysis.*- Proceedings 4th MATHMOD Vienna, Feb. 2003, ARGESIM-Report 24, Vol. 1, S. 99 und Vol. 2 (CD), pp. 604-609, ARGESIM-Verlag, Wien.

AUTHOR BIOGRAPHIES

CHRISTIAN ZEMKE was born in Northeim, Germany and studied Geology at the Technical University of Berlin and the Georg-August Universität in Göttingen, where he obtained his MSc degree in 1995. After he worked at a coal mining facility near Görlitz (Sachsen) he had been hired by the Technical University of Clausthal (1997) as research assistant where he finished his PhD Thesis in 2002. He is now a member of the interdisciplinary research group in the field of simulation for geosystems analysis at the University of Hamburg Dept. Computer Science. His e-mail address is: zemke@informatik.uni-hamburg.de and his Web-page can be found at www.verve-gis.com



KLAUS W. MAAS was born in Dortmund, Germany and went to the Universities of Aachen and Berlin where he studied mine surveying from 1985 to 1991. He finished his PhD thesis in 1999 and presently works at the Institute of Geotechnical Engineering and Mine Surveying at the Technical University of Clausthal (<http://www.igmc.tu-clausthal.de>) as project coordinator (<http://www.verve-gis.com>). His e-mail address is klaus.maas@tu-clausthal.de.



DIETMAR P.F. MÖLLER was born in Preetz, Germany and went to the Universities of Mainz, Bonn and Bremen, where she studied Electrical Engineering, and obtained his doctoral degree in 1980. In 1985 he obtained a degree in Medicine. From 1985 to 1991 Dr. Möller lead the anaesthesia division of Dräger AG in Lübeck, Germany. 1991 he became Professor for computer Science at the Technical University of Clausthal. He went to the University of Hamburg in 1998 where he holds the chair for Computer Engineering. Since 1998 he is also adjunct Professor at the California State University Chico. His E-Mail address is dmoeller@informatik.uni-hamburg.de and his Web-page can be found at <http://www.informatik.uni-hamburg.de/TIS/>



SIMULATION OF MICROMAGNETIC PHENOMENA

Markus-A. B.W. Bolte
Prof. Dr.-Ing. Dietmar P.F. Möller
Fachbereich Informatik,
Arbeitsbereich Technische Informatiksysteme
Universität Hamburg,
Vogt-Kölln-Strasse 30, D-22527 Hamburg, Germany
E-Mail: mbolte@physik.uni-hamburg.de

Dr. Guido D. Meier
Alexander Thieme
Institut für Angewandte Physik und Zentrum für
Mikrostrukturforschung
Universität Hamburg,
Jungiusstrasse 11, D-20355 Hamburg, Germany
E-Mail: meier@physik.uni-hamburg.de

KEYWORDS

Simulation, Micromagnetic Model

ABSTRACT

The study of micromagnetism is a very active field of research with prospective applications in microelectronics, harddisk data storage as well as non-volatile random access memory. In recent years, the use of simulation tools has become one predominant factor in the rapid development of new devices as well as in the deepened understanding of underlying physical principles. To bridge the gap between theoretical predictions and experimental findings we extended an existing simulation package to allow directly comparison of experimental results with the theoretical background that is implemented in the simulation software. In general the comparison results in excellent fits. Moreover we increased the performance of the existing code which will enable further extensions and applications.

INTRODUCTION

A growing understanding of the ferromagnetic properties of micro- and nanometer sized structures have been the prerequisite for much of the technological advances that occurred in the last decades in the field of data storage. It is also likely a crucial factor in future developments of microelectronic and data storage devices such as the magnetic random access memory (MRAM) or the anticipated spin transistor (Datta and Das, 1990, Meier and Matsuyama 2000) which is one possible way to further decrease the size and increase the speed of computer componentware. The microscopic and thermodynamic theory of magnetism has been established well over 70 years ago with fundamental work from Landau, Lifshitz, Gilbert, and others (Landau and Lifshitz 1935; Gilbert 1955). The real challenge is to conclusively compare theory with experiments. Hence, the principles of micromagnetic simulation have become accepted as the main bridge over the last three decades. The importance of numerical and simulation techniques to micromagnetism has several reasons: First, due to the extremely large number of free variables ($3N$, with N being the number of electrons or cells in a specimen) there are only a limited number of special scenarios in which there exists an analytical solution, e.g., the rotational ellipsoid (Stoner Wohlfarth

1948; Aharoni 1998). Secondly, the existing theory offers solutions to a wide range of physical problems from magnetoresistance to domain wall motion and spin wave excitation. Thirdly, a few parameters suffice to provide a good approximation of real systems, and finally, its algorithms have become very fast and able to efficiently simulate large-scale samples while others, e.g., ab-initio approaches, are extremely time-consuming.

In the following paragraphs we give a brief introduction into the principles of micromagnetism, as well as the most important simulation aspects. Thereafter we discuss the structure and performance analysis of an implementation used in micromagnetism and our adaptations and extensions. Moreover we compare our simulation results with measurements done by several techniques, which show excellent fits.

Micromagnetic principles

The micromagnetic theory describes magnetic phenomena such as hysteresis effects and domain walls and is based on the approximation to use a continuous magnetization vector instead of discrete magnetic moments located on the sites of the crystal lattice, i.e., on the atomic length scale. It also sets the scale in which this assumption is valid, i.e., the nanometer scale is the lower limit. In principle no upper limit exists apart from computational limitations. Initial works by (Landau and Lifshitz 1935; Gilbert 1955; Brown 1963) have given the theory a foundation that has become widely accepted and that holds true for a wide variety of applications. The foundation to micromagnetic theory is the equation of motion established from thermodynamic consideration (minimization of the Gibbs' free energy) by Landau, Lifshitz and Gilbert. It describes the change of a magnetization vector over time as a function of the effective magnetic field at the location of the vector and some parameters inherent in a magnetic system, i.e., the saturation magnetization M_s , which is the strength of the magnetization vector, the gyromagnetic constant γ and the damping constant α . Similar to the motion of a gyroscope where a gravitational field causes a precessional movement around the axis of the gravitational pull, the magnetization precesses around the direction of a local effective field H_{eff} . A sketch is shown in Figure 1.

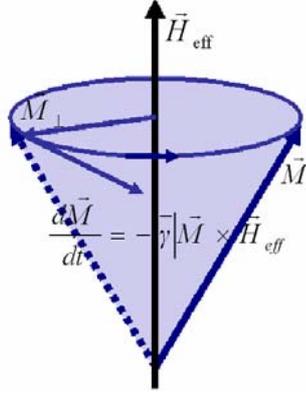


Figure 1: The Magnetization Vector Precedes around the Direction of the Effective Field. In a Real Sample the Damping Term of the LLG Equation Causes the Magnetization to Align with the Effective Field after a Characteristic Time.

To this precession the damping term is added that takes into account the fact that energy is exchanged with the surrounding system, thus the magnetization loses energy over time and tends to align with the field vector of the effective magnetic field at the location of the magnetization. Thus the equation of motion for the micromagnetic model is written in the form of the Landau-Lifshitz-Gilbert (LLG)-equation

$$\frac{d\vec{M}}{dt} = -\gamma \vec{M} \times \vec{H}_{\text{eff}} - \frac{\gamma\alpha}{M_S} \vec{M} \times (\vec{M} \times \vec{H}_{\text{eff}}). \quad (1)$$

The effective field originates from the superposition of four different interactions between the spins of magnetic material (in the absence of mechanical forces we can neglect magnetostrictive effects). The sum of their energies is the total energy of the system which becomes minimal in stable spin configurations

$$E_{\text{tot}} = E_{\text{exch}} + E_{\text{aniso}} + E_{\text{Zeeman}} + E_{\text{demag}}. \quad (2)$$

These four contributing terms are exchange energy, anisotropic energy, Zeeman energy, and de- or selfmagnetization energy. They have their origin in the quantum mechanical nature of magnetism and can not satisfactorily be understood without a comprehension of its principles. In quantum mechanics, electrons are described as wave functions, and the location of electrons can be seen as the square of the amplitude of such a function. The location cannot be determined with infinite accuracy, but rather has an uncertainty that depends on the electron's velocity (uncertainty principle). The integral of the wave function is normalized to one; the wave function can therefore be seen as the probability function of the location of the electron. From these basic considerations the electronic orbitals of atoms can be calculated; they determine the probability of electrons being at a certain location in relation to the atom's nucleus. The three space coordinates are transformed into integer quantum numbers that describe their energetic properties. Essential to magnetism is the electrons'

fourth intrinsic attribute, its spin. It is directly proportional to the magnetic moment of the electron, the motion of which is described by the LLG equation. The electronic and magnetic properties of charged particles are inseparably connected by Maxwell's equations. More details can be taken from introductory physics books (Cohen-Tannoudji et al. 1996; and Jackson, 1998).

Energy terms

The energy of a solid state system can be calculated by taking into account all possible interactions of the electrons with each other and with outer electric and magnetic fields. As stated in the introduction, this is very costly concerning both time and computational resources. Within the theory of micromagnetism the electric and magnetic properties of many individual atoms are averaged to material constants such as the strength of the interaction of the spins of electrons in close proximity due to an overlap of their wave functions. This is an extremely strong but only close-range interaction known as the exchange coupling. The change in energy from this effect can be calculated as

$$E_{\text{exch}} = -J \sum_{i \neq j} S_i \cdot S_j, \quad (3)$$

where J , also called exchange integral, is the shared volume of the wave functions of electrons i and j and S_i and S_j their spins. Here the indices can also stand for neighboring simulation cells. What the equation says is that when considering only the exchange coupling, the spins (magnetization vectors) of neighboring atoms (simulation cells) want to align parallel to make the exchange energy minimal. The exchange energy is normally determined by considering only nearest-neighbor atoms/ cells, but it does not take into account the spatially uneven (anisotropic) distribution of electrons according to their quantum numbers as mentioned above. This is done by introducing a term known as the anisotropy energy, which considers the shape of the electronic orbitals and the location of neighboring atoms in the lattice of a solid. For uniformly magnetized ferromagnets, the spins of the electrons tend to align to so-called easy axes as described by the Stoner-Wohlfarth-model for rotational ellipsoids (Stoner and Wohlfarth 1948). For example, for materials organized in hexagonal closed packed lattices, there is one principle easy axis, and its energy term is written as

$$E_{\text{aniso}} = K_{u1} \int_V \sin^2 \phi dV + K_{u2} \int_V \sin^4 \phi dV, \quad (4)$$

where ϕ is the angle between magnetization and the easy axis, and K_{u1} and K_{u2} are uniaxial anisotropy constants, derived from Taylor series expansion. Typically the second order term is already much smaller than the first order anisotropy.

A third contribution is the Zeeman term that describes the influence of external magnetic fields on the spins. Basic electrodynamic calculations result in an added energy

$$E_{Zeeman} = - \int_V \vec{M} \vec{H} dV. \quad (5)$$

Here M is the magnetization of one simulation cell, which is the averaged unbalanced spin of all atoms represented by the cell, and H the external magnetic field. The scalar product of the two is then integrated over the cell volume V to derive the field energy. Again, the magnetization will seek to align itself to the direction of the outer field as described by the LLG-equation. Exchange coupling and Zeeman interaction are competing against a fourth term that depends on the shape of the magnetic system. This demagnetization energy depends on the interaction of the magnetization with the stray field H_s generated by the magnetization itself, and is written as

$$E_{demag} = - \frac{1}{2} \int_V M H_s dV. \quad (6)$$

The magnetization can also be seen as magnetic polarization charges which are unbalanced at the surfaces of a structure. Just as with their electric equivalents, this creates a dipole field that is the stronger the more charges are unbalanced at the surface. This means, that under the influence of this interaction the magnetization vectors will try to be arranged parallel to a sample's surface, because then there would be no polarization charges. The stray field can be approximated as the field of a dipole that occurs between two charges with a relative displacement between them. The stray field is a long-range interaction between magnetizations and thus often becomes the dominating part of the effective field.

Domain walls

With the four competing energy terms described above the magnetization of ferromagnetic materials in zero-field environments generally tend to be arranged parallel to each other and preferably along easy axes in large volumes called Weiss areas (Weiss 1907) or domains, thus minimizing exchange and anisotropy energies, while on the boundaries the stray field (also called demagnetization field) is prevalent. This means that on the interfaces of these domains a change in the direction of the magnetization vectors occurs; this can be viewed as a wall between the different domains. Since their first discovery and investigations by (Bloch 1923, Néel 1955) and subsequent studies done by (Aharoni 1996) and others research on domain walls has become of major interest, because the electronic and magnetic properties of materials depend largely on the behavior of domain walls (Hertel and Kronmüller 1999). One has to differentiate between several types of walls: In the Bloch wall the magnetization vector rotates around an axis normal to the domain wall, leading to surface charges at the interface of the wall with other media. Another type, the Néel wall, is prevalent in thin films; here the vector rotates in plane of the structure. Other types, such as the cross-tie or the asymmetric Bloch

wall are combinations of the fundamental types. Depending on parameters such as spatial extension and energy coefficients which configuration will appear (Ramstöck, Hartung and Hubert 1995).

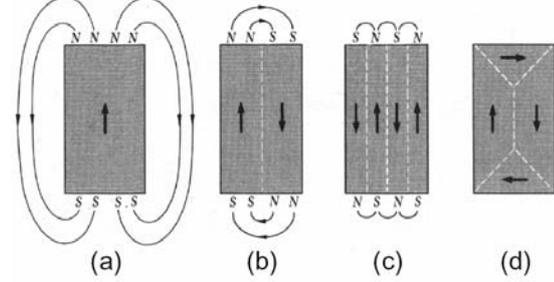


Figure 2: Formation of domains reduces the stray field. From left to right, the demagnetization energy is reduced by the formation of domains, especially by closure domains. Modified from (Hubert and Schäfer 1998)

Consider as a simple example the rectangular particle of Figure 2. Let us assume that the external field is zero, i.e., the Zeeman energy vanishes. We further assume that the particle possesses first order uniaxial anisotropy along the long axis. The remaining energy contributions are the demagnetization and the exchange energy, the latter of which tries to align all magnetic moments parallel. If exchange is the leading energy contribution, as it is true for small particles the configuration of Figure 2(a) is preferred. However, in this case the stray field of the particle is maximized which can be easily seen by the large number of free poles at the surface, equivalent with a high demagnetization energy being proportional to the stray field times the magnetization of the particle. The configuration of Figure 2(b) reduces the stray field energy at the cost of the generation of a domain wall in the middle of the rectangle. Within the wall magnetic moments are pointing in hard directions and cause anisotropy energy and also an exchange energy contribution. Depending on the material parameters and the size of the rectangle configuration (c) might further reduce the total energy. The arrangement of domains shown in Figure 2(d) is known as the Landau pattern and often occurs in nature, because it couples a major part of the magnetic flux inside the particle, thus minimizing the stray field, i.e., the demagnetization energy. Most of a multi-domain sample's magnetic energy is therefore contained in the domain walls.

COMPUTATION OF MICROMAGNETIC PROBLEMS

Science is concerned with understanding and controlling the materials and forces of nature for the benefit of humankind. Therefore it is necessary for scientists to analyze and improve the performance of systems, when the components of which originate from different domains. Examples include either adapting existing systems to new demands and/or conditions, or designing new applications such as those in mechatronics, automotive, avionics, aerospace, micromagnetics, etc. Such systems in-

clude components derived from many different science domains. In many cases, solutions to problems have been found by applying appropriate mathematical models and computer simulation to them. A recent White House report identifies computer modeling and simulation as one of the key enabling technologies of the 21st century. Its applications are virtually universal.

There are several excellent micromagnetic simulation packages available today, with different modeling schemes. We have decided on OOMMF – the object oriented micromagnetic framework, a project in the National Institute of Standard and Technology (NIST, <http://math.nist.gov/oommf>) for several reasons. First its code is freely available, making it possible to compare techniques and algorithms with established publications, a vital criterion for scientific research. With its modular architectural design it is also extensible and very flexible. It has also proven to be a very fast application for the dynamic 3D simulations of the rectangular magnetic samples that we mainly investigate.

OOMMF is implemented as a client/server-architecture in which the different components request and offer services via a service directory. These components include user interfaces for the input of the problem to be simulated, so-called solvers that progress the simulation time by stepwise integration of the above LLG ordinary differential equation (ODE), and display components for scalar and vector values. A script file gives a description of the problem; the exchange of data is performed via TCP ports. The code can be modified at three levels; first one can choose different display components and settings to vary the output format. Secondly, one can adjust material constants, the OOMMF components used and their input parameters to control sample and simulation attributes; and finally, a full and arbitrary access on the C++ code itself allows debugging, modification and extension of any component of the package. The main classes of the existing solver shall be explained in more detail in the following paragraphs, followed by a description of our extensions and adaptations as well as a comparison of our simulations with experiments done with various techniques.

Structure and Performance of the Program

A director component controls the loading of involved components and determines when a simulation run is complete. It insures that read/write access on individual simulation stages is not violated.

One can choose between two standard techniques to evolve the magnetization configuration towards a state of equilibrium: Either by minimizing the total energy by using a conjugate gradient technique to find the path to a local energy minimum, or by integrating the equation of motion mentioned above in small time steps. Only with the second method one can follow the dynamic changes in the magnetization configuration, which is why we will focus on this evolver. Each evolver has its own driver that controls stopping criteria and the progression from one time step to another as well as different simulation stages, i.e., when the strength of the ex-

ternal field is ramped up in the case of simulation of hysteresis curves.

The time evolver uses integration methods to further the simulation one step. For that, two disjoint simulation states are used so that the data is clearly separated. In the current state all information needed for the advance of the magnetization configuration is stored. The next configuration is then determined by either using a simple Euler or an adaptive Cash-Karp implementation of the Runge-Kutta method that was implemented to speed up computation. The magnetization vectors then are the foundation for the computation of the effective field; its value is stored as a foundation of the computation of the next time step. Last the time step itself is evaluated against some error restriction to determine whether the step has to be repeated with a smaller step size.

The analysis of the run time performance showed that most of the computing time (62 to 82%) is spent in computing the demagnetization energy (see Figure 3); the results are in good accordance with other findings (Berkov et al. 1993). The complexity is due to the long-range nature of the stray field and the current algorithm used to compute the same. Because the influence of the stray field converges only slowly, the interaction with the magnetization vectors of each cell of a sample has to be considered to derive the field strength at every simulation cell, so that for N cells one would have a $O(N^2)$ complex algorithm if one would use this crude method.

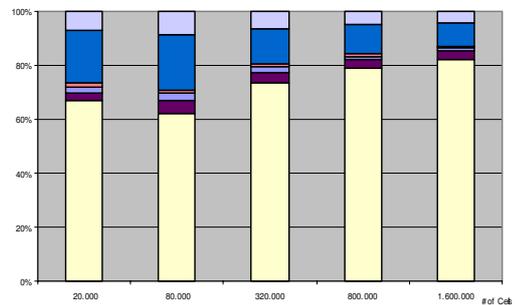


Figure 3: Break-up of the Runtime for Simulations with Different Input Sizes into the Main Routines. Colours represent (from bottom to top): Stray, Exchange, Anisotropy and Zeeman Field Computation, Integration and Computation of Derivative

In the existing simulation package a sophisticated algorithm as described in (Newell et al. 1993) is used which performs at $O(N \log N)$ due to Fast Fourier Transforms. This method uses a demagnetization tensor N to account for the shape of a magnetized body. The stray field

$$H_{Stray,i} = -\sum_j N(r_i - r_j) M_j \quad (7)$$

can be seen as a convolution of two vectors in spatial coordinates. It can be substituted for a simple product in frequency space via FFT so that its complexity is $O(N \log N)$, but the coefficient is large which is why it consumes most of the computation time.

Expansion and Results

From this introduction it can be seen that an effort to increase the computation speed should be focused on objectives: First, to decrease the number of steps necessary to derive a certain convergence criterion, and second, to reduce the time needed to compute one simulation step. The first objective can be obtained by applying well-known methods for integrating ODEs, such as Runge-Kutta integration or other methods. The second objective requires faster algorithms, especially for the demagnetization term. One possible way is to approximate the stray field by using a multipole expansion of the dipole field to a given order of accuracy as stated in (Yuan and Bertram 1992). The advantage for using this method, aside from having linear complexity, is that this algorithm would be parallelizable as suggested in (Singh et al. 1993). An excellent overview over the current state of the art is given in (Fidler and Schrefl 2000). One important enhancement for the existing code due to speed up purposes is the implementation of a Cash-Karp Runge-Kutta algorithm as described in (Press et al. 2002). The commonly available code uses the Euler-method for integration with an adaptive time step derived from error estimation. The speedup achieved was significant; depending on the problem's stiffness as given by the damping constant α , up to a factor of 20 (for small damping and relatively small problems).

Another part of our work included adapting the output of the simulation package to our experimental setup to allow direct comparison between experiments and the established theory, as implemented in the OOMMF code. We are using the magnetic-force microscopy (MFM) and in collaboration with Dr. P. Fischer (Fischer 1998; Fischer 2001) from the Max-Planck-Institute for metal physics in Stuttgart magnetic transmission x-ray microscopy (MTXM) to investigate the magnetic properties of permalloy and iron-nickel samples. As shown by (Barthelmeß et al. 2004) the MFM image can be effectively simulated using the second derivative of the magnetization's stray field in z-direction, while the MTXM-method shows the projection of the magnetization vectors on the axis of the transmitting x-ray beam as grey-scale values. The additional component to the OOMMF architecture takes the measuring height above the sample surface as a free parameter. Figure 4 shows examples of the good fits between experiment and simulation, others can be seen in (Meier et al. 2002; Barthelmeß et al. 2003; Pels et al. 2004).

Another application of the additional feature is that it allows direct comparison between the two techniques, serving as a bridge between two standard measuring methods (Meier et al. 2004). Figure 5 shows one such example: The experimental data were taken from magnetic transmission x-ray microscopy (upper left), which basically takes an x-ray image of a ferromagnetic sample – the grey scale values corresponding to the scalar product of the sample's magnetization with the direction of the transmitting x-rays – as well as from magnetic-force microscopy (lower left). The right column shows the simulated magnetization image (upper right) yield-

ing the same Landau structure as seen in the MTXM image, and the simulated MFM image (lower right) revealing the second derivative of the stray field akin to the one taken from magnetic-force microscopy. This example shows how the added module of the existing software package is a useful tool in both comparing and validating theoretical and experimental data, as well as bridging different measurement techniques.

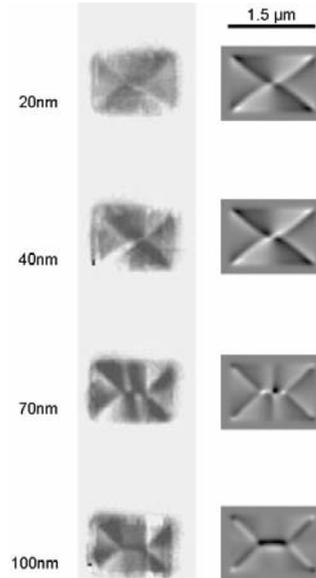


Figure 4: Magnetization patterns of $1 \times 1.5 \mu\text{m}^2$ permalloy samples with different thicknesses as indicated in the figure, left measured by magnetic-force microscopy, right from micromagnetic simulations. From (Barthelmeß et al. 2004).

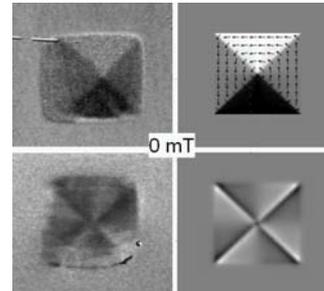


Figure 5: Isolated $2 \times 2 \text{ m}^2$ sized Micro Contact. Modified from (Meier et al. 2004). See Text for Explanation.

CONCLUSIONS

We outlined the problem by giving a brief discourse on the fundamentals of micromagnetism. We described the structure of the simulation software and showed several ways to decrease computation time by using more effective integration schemes as well as faster, parallelizable algorithms for the stray field computation. We extended the existing code to enable direct comparison with and between experimental data observed with two measurement techniques, yielding good fits. Speedup and

extension provide a good foundation for fast simulations working hand-in-hand with micromagnetic experiments.

REFERENCES

- Aharoni, A. 1996. *Introduction to the Theory of Ferromagnetism*. Clarendon, Oxford.
- Aharoni, A. 1998. "Demagnetizing Factors for Rectangular Ferromagnetic Prisms". *J. Appl. Phys.* **83**, 3432-3432.
- Barthelmeß, M.; Thieme, A.; Eiselt, R.; and Meier, G. 2003. "Stray fields of iron electrodes for spin-polarized transport". *J. Appl. Phys.* **93**, 8400-8404.
- Barthelmeß, M.; Pels, C.; Thieme, A.; and Meier, G. 2004. "Stray fields of domains in permalloy microstructures - measurements and simulations". *J. Appl. Phys.* **95**, 5641
- Berkov, D.V.; Ramstöck, K. and Hubert, A. D. V. 1993. "Solving micromagnetic problems: Towards an optimal numerical method", *Phys. Status Solidi A* **137**. 207-225.
- Bloch, F. 1932. *Z. Phys.* **74**, 295 (1932)
- Brown, W. F. Jr. 1963. *Micromagnetics*. Interscience Publishers, New York.
- Cohen-Tannoudji, C.; Diu, B. and Laloe, F. 1996. *Quantum mechanics*. John Wiley & Sons.
- Datta, S. and Das, B., 1990. "Electronic analogon of the electro-optic modulator". *Appl. Phys. Lett.* **56**, 665.1996. *Zf. Physik B* **101**, 313.
- Fiedler, J. and Schrefl, Th. 2000. "Topical Review: Micromagnetic Modelling – the Current State of the Art". *J. Phys. D: Appl. Phys.* **33**, R135-156
- Fischer, P.; Eimüller, Th.; Schütz, G.; Guttman, P.; Schmahl, G.; Prügl, K.; and Bayreuther, G. 1998. *J. Phys. D: Appl. Physics* **31**, 649.
- Fischer, P.; Eimüller, Th.; Schütz, G.; Denbeaux, G.; Lucero, A.; Johnson, L.; Attwood, D.; Tsunashima, S.; Kumazawa, M.; Takagi, N. Köhler, M.; and Bayreuther, G. 2001. *Rev. Sci. Instr.* **72**, 2322.
- Gilbert, T. L. 1955. "A Lagrangian Formulation of the Gyromagnetic Equation of Magnetization Field", *Phys. Rev.* **100**, 1243.
- Hertel, R. and Kronmüller, H. 1999. "Computation of the magnetic domain structure in bulk permalloy". *Phys. Rev. B* **60**. 7366-7378.
- Hubert, A. and Schäfer, R. 1998. *Magnetic Domains: The Analysis of Magnetic Microstructures*. Springer.
- Jackson, J.D. 1998. *Classical Electrodynamics*. Wiley Text Books, 3rd Ed.
- Landau, L. and Lifshitz, E. 1935. "On the Theory of the Dispersion of Magnetic Permeability in Ferromagnetic Bodies", *Physik. Z. Sowjetunion* **8**, 153.
- Meier, G. and Matsuyama, T. 2000. "Magnetic electrodes for spin-polarized injection into InAs". *Appl. Phys. Lett.* **76**, 1315.
- Meier, G.; Eiselt, R.; and Halverscheid, M. 2002. "Hall micromagnetometry on iron electrodes suitable for spin-polarized transport". *J. Appl. Phys.* **92**, 7296.
- Meier, G.; Eiselt, R.; Bolte, M.; Barthelmeß, M.; Eimüller, Th.; and Fischer, P. 2004. "Comparative study of magnetization reversal in isolated and stray field coupled micro contacts". submitted
- Möller, D. P. F.: *Mathematical and Computational Modeling and Simulation*, 2003, Springer Publ. Heidelberg
- Néel, L. 1955. *Compt. rend., Paris* **241**, 533.
- Newell, A.J.; Williams, W.; Dunlop, D.J. 1993. "A Generalization of the Demagnetization Tensor for Nonuniform Magnetization". *J. Geophys. Res.* **98**, No. B6. 9551-9555
- Pels, C.; Barthelmeß, M.; Bolte, M.; Thieme, A.; Meier, G. 2004. "Thickness and magnetic-field dependence of do-

- main switching in isolated and interacting permalloy contacts". submitted
- Press, W.; Teukolsky, S.A.; Vetterling, W.T.; and Flannery, B.P. 2002. *Numerical Recipes in C, The Art of Scientific Computing, 2nd Ed.* Cambridge University Press, New York, NY
- Prinz, G. 1995. "Spin-polarized Transport", *Physics today*, **Apr.** 58-63.
- Ramstöck, K.; Hartung, W.; and Hubert, A. 1996. "The Phase Diagram of Domain Walls in Narrow Magnetic Strips". *phys. stat. sol. (a)* **155**, 505.
- Rüdiger, U.; Yu, J.; Thomas, L.; Parkin, S.S.P.; Kent, A.D. 1999. "Magnetoresistance, Micromagnetism, and Domain Wall Scattering in Epitaxial hcp Co Films" *Phys. Rev. B* **59**, 11914-11918.
- Demokritov, S.O.; Serga, A.A.; Demidov, V.E.; Hillebrands, B.; Kostylev, M.; Kalinikos, B.A. 2003. "Experimental observation of symmetry breaking nonlinear modes in an active ring". *Nature* **426**, 159
- Thiaville, A.; Garcia, J.M.; and Miltat, J. 2002. "Domain wall dynamics in nanowires". *J. Magn. Magn. Mater.* **242-245**. 1061-1063
- Singh, J.P.; Holt, C.; Hennessy, J.L.; and Gupta, A. 1993. "A parallel adaptive fast multipole method". *Supercomputing '93. IEEE Computer Society*. 54-65.
- Stoner, E.C. and Wohlfarth, E.P. 1948. "A Mechanism of Magnetic Hysteresis in Heterogeneous Alloys", *Phil. Trans. Royal Soc. London* **A240**. 599.
- Weiss, P. 1907. *J. de Phys.* **6**, 661.
- Yuan, S. W. and Bertram, H. N. 1992. "Fast Adaptive Algorithm for Micromagnetics". *IEEE Trans. Magn.* **28**, No.5, 2031-2036

AUTHOR BIOGRAPHIES



MARKUS-A. B. W. BOLTE is currently working on his master thesis in both Physics and Computer Science at the University of Hamburg, Hamburg, Germany, on the topic of simulation of micromagnetic structures.



DIETMAR P. F. MÖLLER is a professor of computer science and engineering and chair of computer science at the University of Hamburg, Hamburg, Germany. He is currently working on embedded computing systems, virtual and augmented reality, modeling and simulation, mobile autonomous systems, soft computing, medical informatics and nanostructures.



GUIDO D. MEIER is a division leader in the research group of Prof. U. Merkt at the Institute of Applied Physics of the University of Hamburg, Germany. His research activities include semiconductors, ferromagnets and hybrid devices of both materials on the micro- and nanometer scale.

ALEXANDER THIEME is a graduate student at the University of Hamburg working on his master of physics and his M.D. on the topic of simulation of biomolecular processes.

Fuzzy modelling of mobile autonomous soccer-playing robots - An educational approach with LEGO Mindstorms robots

Birgit Koch, Dietmar P.F. Möller
Universität Hamburg, Fachbereich Informatik,
Arbeitsbereich Technische Informatiksysteme,
Vogt-Kölln-Str. 30, 22527 Hamburg
E-mail: {koch, dmoeller}@informatik.uni-hamburg.de

KEYWORDS

Mobile Autonomous Robots, Fuzzy Modelling,
RoboCup, Education

ABSTRACT

Soccer-playing robots are being used in education to foster understanding and interest in artificial intelligence, multimodal systems, engineering and science in order to solve complex problems through active learning comprehensive knowledge undergoing real-world applications. In 1997 the RoboCup Initiative started a broad international program of research and education, that has the aim to promote artificial intelligence and robot research by providing problems like soccer-playing robots or rescuing robots to be solved by integration of different technologies and collaboration of various resources. As the LEGO Mindstorms robotic construction kits are convenient in education, they are often used for RoboCup Junior competitions. To represent how fuzzy logic can be used in education, an example for moving a soccer-playing mobile autonomous robot based on LEGO Mindstorms is described.

INTRODUCTION

Over the past thirty years the “Epistemology and Learning Group” of the Massachusetts Institute of Technology (MIT) has searched for correlations between learning environments and learned skills. The idea of robotic construction kits is based on the research of Seymour Papert, who described the first steps of utilization of computers and robots in the learning environment of

children in his book MINDSTORMS: Children, Computers and Powerful Ideas [Papert 1980]. Starting from the development of turtle robots and the child-friendly programming language LOGO, in 1998 the MIT and the LEGO company came out with the first LEGO Mindstorms robotic construction kits. Even though the product is primary centered to children and teenager, many adults are also attracted by the robotic construction kits. There are many active and creative online-groups that develop alternative programming environments and advanced construction techniques.

In this paper we describe the robotic construction kits and programming environments that we use in the course “Hamburg Robocup: Mobile autonomous robots play soccer” at the University of Hamburg as well as some examples of teaching fuzzy logic with LEGO Mindstorms robotic construction kits.

ROBOTIC CONSTRUCTION KIT AND PROGRAMMING

To understand the relevance of using robotic construction kits at universities, it is important to know the elements of robotic construction kits and how they can be programmed. In the following this is exemplified through the LEGO Mindstorms robotic construction kit, that consists of the following ingredients: a programmable RCX-brick (Hitachi H8/3292-microcontroller with 16 KB ROM and 32 KB RAM), light sensors and touch sensors, motors, many common LEGO bricks, an infrared sender to transmit data to a computer, the programming environment Robotics Invention System (RIS) and a construc-

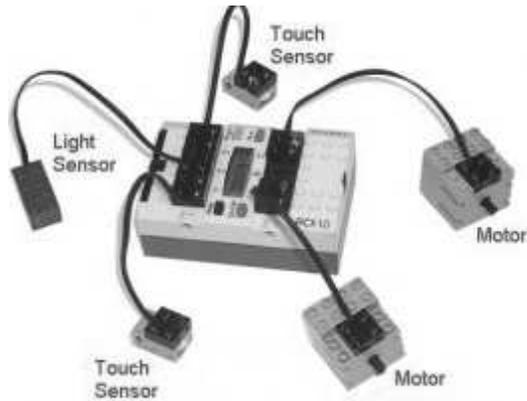


Figure 1: RCX-brick with sensors and motors

tion handbook. The RCX-brick contains three inputs for sensors, three outputs for motors or lamps, five spots for programs, a LCD-display, four control buttons, a speaker and an infrared interface. Fig. 1 shows an RCX-brick with two motors, a light sensor and two touch sensors.

The Robotics Invention System (RIS) is a graphic-based programming environment that works with blocks. Each block stands for one instruction. To program the robot the blocks are joined per Drag and Drop in form of a chain. Running the program the blocks are executed in sequence of the block chain. To read the values of the sensors while executing the program, parallel block chains can be used.

Beside the software RIS, that is part of the LEGO Mindstorms robotic construction kit and that is directed to children and teenager without programming skills, the active LEGO online-groups developed some other possibilities for advanced programmers to work with the robotic construction kits. Most of this software can be found as freeware in the internet. Here are some examples of other programming environments that was used in the course “Hamburg Robocup: Mobile autonomous robots play soccer” at the University of Hamburg:

ROBOLAB: works with kind of advanced flowcharts, based on LabVIEW, was developed especially for use in schools

RCX Command Center: with the programming language Not Quite C (NQC), a language similar to C, programs can be written text-based

LEGO Java Operating System (lejOS): an implementation of a Java Virtual Machine (JVM)

Other enhancements can be found in [Baum 2000, Baum et al. 2000, Erwin 2001, Knudsen 1999].

HAMBURG ROBOCUP: MOBILE AUTONOMOUS ROBOTS PLAY SOCCER

Over the past years the utilization of robotic construction kits in schools was analyzed and appreciated in many studies [Christaller et al. 2001, Müllerburg 2001], but in the context of universities they are often depreciated as toys and as a consequence as irrelevant for teaching students [Koch 2003]. Using “real” robots has often the disadvantage that they are very expensive, so that many students have to share one robot. Additionally it can be difficult to motivate students to work with robots and to teach the basics of robotics because the orientation time of a complex robot system often requires weeks or months. To avoid those problems, we decided to use robotic construction kits in the undergraduate course “Hamburg Robocup: Mobile autonomous robots play soccer”. The goal of the class is to work with mobile autonomous robots, so that the students develop and understand the complexity of non-linear systems. Additionally soft skills like communication, teamwork and project-management are supported. In the course “Hamburg Robocup: Mobile autonomous robots play soccer” the students have the exercise to construct and program a robot that plays soccer considering the rules of RoboCupJunior - the pupil league of RoboCup [Kroeger et al. 2000, Lund 1999, Lund et al. 1999, Lund and Pagliarini 2000, Stone and Veloso 1998]. RoboCup is a broad national and international program of research and education, that has the goal to promote artificial intelligence and intelligent robot research by providing problems to be solved by the integration of different technologies and the collaboration of various resources.

During the course contests are realized to test the performance of the robots. At the beginning some problems crop up: the robots don’t find the ball, the robots aim at the wrong goal, etc. But most of the students are stimulated by those

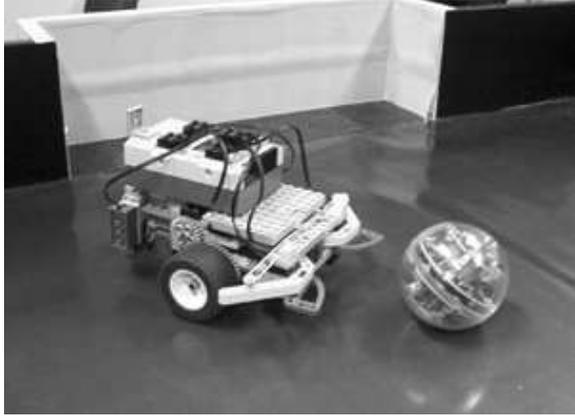


Figure 2: Soccer-playing mobile autonomous robot

problems to extend their work. One approach to deal with the fuzzy dynamic behavior of the robots - modelling the robots with fuzzy logic - is described in the following section.

FUZZY MODELLING OF MOBILE AUTONOMOUS SOCCER-PLAYING ROBOTS

Modelling a robot with Fuzzy logic is one example of a computer science branch that can be taught effectively by mobile autonomous robots. Fuzzy statements that are recognized by the students during the work with the robots can be precisely expressed with Fuzzy logic. Fig. 2 shows a soccer-playing LEGO Mindstorms robot on the playground. While the students work with the robots they find out that the robots do not move straight forward although both motors run with the same performance. In this situation fuzzy logic can be very helpful to solve the problem. The students learn by means of linguistic variables of the mobile robot to determine the membership function of the linguistic variables and to draw up rules to describe the dynamic behavior of the soccer-playing robots.

To represent how fuzzy logic can be used in a control system, in the following an example for moving a soccer-playing mobile robot based on LEGO Mindstorms, is discussed. For a brief introduction into fuzzy logic see [Möller 1995].

The position of the robot movement is determined by some linguistic variables: the direction angle, denoted as α , the distance from the object, which is for a soccer-playing robot the ball, de-

Distance x	Input variable
<i>RST</i>	right side of track toward goal
<i>CT</i>	center of track toward goal
<i>LST</i>	left side of track toward goal

Table 1: Distance variables

Dir. angle α	Input variable
<i>ND</i>	north direction toward the goal
<i>ED</i>	east direction toward the goal
<i>SD</i>	south direction toward the goal
<i>WD</i>	west direction toward the goal

Table 2: Direction angle variables

noted as x and the direction of the mobile robot movement, denoted as β , that is determined by the angle of the wheels steering position. For a given initial robot position within the specific area, the soccer playground, the goal for the mobile robot is to move toward the center of the ball. The desired final position is to let the robot moving the ball on a track toward the goal. α , β and x are the respective linguistic variables for this purpose. To each of these linguistic variables, a set of linguistic values can be assigned as shown in Table 1, 2 and 3:

As shown in Fig. 3, a range of numerical values can be assigned to each linguistic value of a linguistic variable. Each graph, called a membership function, indicates the degree to which an input value belongs to a particular linguistic value. Such a degree of membership ranges from 0 to 1. The value 0 indicates no membership, and the value 1 represents full membership. Hence a value between 0 and 1 represents a partial membership.

The rules have to be defined, describing the dynamic behavior of the soccer-playing robot. In general, each rule produces some output linguis-

Wheels angle β	Output variable
<i>TR</i>	turn right toward the goal
<i>SF</i>	straight forward toward the goal
<i>TL</i>	turn left toward the goal

Table 3: Wheels angle variables

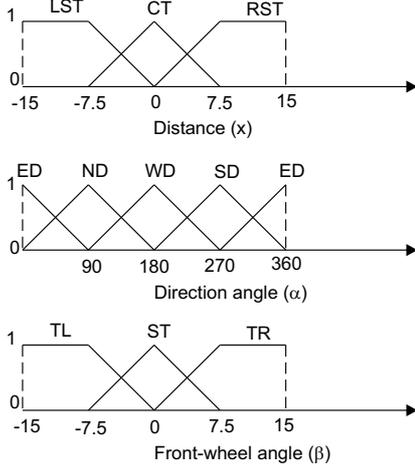


Figure 3: Membership functions for the distance x , direction angle α and wheels angle β

	<i>RST</i>	<i>CT</i>	<i>LST</i>
<i>ND</i>	<i>TL</i>	<i>SF</i>	<i>TR</i>
<i>ED</i>	<i>TL</i>	<i>TL</i>	<i>SF</i>
<i>SD</i>	<i>TR</i>	<i>TL</i>	<i>TL</i>
<i>WD</i>	<i>SF</i>	<i>TR</i>	<i>TR</i>

Table 4: FAM set of rules

tic values based on some input linguistic values. In the case of the mobile robot, some of the rules can be defined as

if $\alpha = ND$ *and* $x = LST$ *then* $\beta = TR$

if $\alpha = ND$ *and* $x = CT$ *then* $\beta = SF$

if $\alpha = ND$ *and* $x = RST$ *then* $\beta = TL$.

These rules can be extended to consider all the possible values for α ; thus there will be 12 rules in all, which can be represented in the fuzzy associative memory (FAM), shown in Table 4.

For given input values for x and α , the fuzzy-logic controller can determine an output value for β . For this purpose, for each input value the fuzzy controller determines the membership degree of its corresponding linguistic values. As a next step, for each rule, as shown for example in Table 4, the minimum of the membership degrees of its antecedents is chosen as a membership degree for the rules consequent, which is considered as a weight for the rules consequent. When there is more than one membership degree for a consequent, the MAXIMUM operator is chosen for that consequent. Hence the membership

degree is assigned to each linguistic value. If a crisp output is required from the fuzzy rule base rather than the fuzzy output set, a process called defuzzification is used to compress this information. The crisp output is generally obtained using a mean of maxima or a center of gravity defuzzification strategy. The most widely adopted method for defuzzifying a fuzzy set A of a universe of discourse Z , is the centroid defuzzification or center of gravity method, which is based on the centroid of area z_{COA}

$$z_{COA} = \frac{\int_Z \mu_A(z)zdz}{\int_Z \mu_A(z)dz}.$$

where $\mu_A(z)$ is the aggregated output of the membership function and z_{COA} is the control output, which equals the fuzzy centroid of A , where the limits of integration correspond to the entire universe of discourse Z of angular values of the steering wheel(s) velocity values.

The center of gravity method, *COG*, provides a weighted average of all linguistic output values. A simplified calculation is as follows:

$$COG = \frac{\sum_{i=1}^n c_i * L_i}{\sum_{i=1}^n L_i}.$$

where the L_i are the weights of linguistic output values and the c_i are the weighting factors.

As an illustration of the information process between the fuzzification and defuzzification, Fig. 4 shows the signal flow through a continuous fuzzy-logic system using the center of gravity defuzzification method. There exist p multivariate fuzzy input sets and q univariate fuzzy output sets.

Example 1: Let the starting point of the mobile robot be at direction $x = -10.0$, and the direction angle $\alpha = 90^\circ$. For these initial values the membership degree of the linguistic input values are for the distance $x = -10.0$: $\mu_{RST} = 0$; $\mu_{CT} = 0$; $\mu_{LST} = 1.0$, and for the direction angle $\alpha = 90^\circ$: $\mu_{ND} = 1.0$; $\mu_{ED} = 0$; $\mu_{SD} = 0$; $\mu_{WD} = 0$. Combining distance and direction, as shown in the fuzzy associative memory in Table 4, with the respective membership degree μ_i for each rule consequently results in the membership matrix, shown in Table 5.

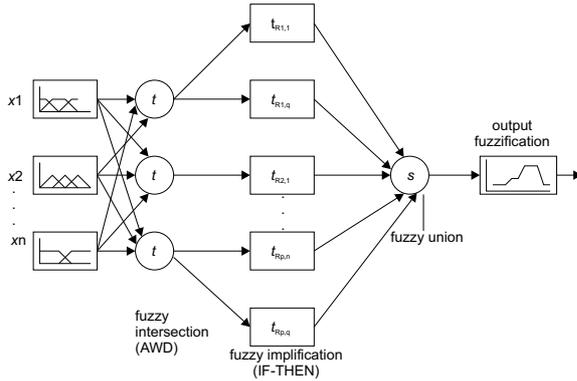


Figure 4: Information flow through a continuous fuzzy system with p multivariate fuzzy input sets and q univariate fuzzy output sets

	<i>RST</i>	<i>CT</i>	<i>LST</i>
<i>ND</i>	1	0	1
<i>ED</i>	0	0	0
<i>SD</i>	0	0	0
<i>WD</i>	0	0	0

Table 5: Membership degrees for each FAM rule based on the initial conditions

The system output-value calculation can be evaluated by using the center of gravity method, and the MAX operator, while there is more than one membership degree. In this formula the maximum degree of each of the four membership degrees is chosen, 1 for *TR*, and 0 and 0 for *ST* and *TL*, respectively. Based on these degrees we receive the mobile robot system output value as follows:

$$COG = \frac{(20 * 1) + (-20 * 0) + (0 * 0)}{1} = 20 .$$

That is, that the wheels of the mobile robot will turn right with an angle of 20° . The robot moves for a short distance and then the process repeats for the new position.

Example 2: Let the slope of a terrain range between -45° and $+45^\circ$, which can be divided into several memberships in between large negative and large positive. We will further assume that the terrain can vary between very rough, rough, moderate, and smooth, and the output speed of the fuzzy-logic system may range between 0 and

15 mph, divided into very slow, slow, medium, fast, and very fast. The rules of the fuzzy-logic mobile robot system are as follows:

if slope is large-positive and terrain is very-rough then speed of the robot is very-slow

if slope is large-positive and terrain is rough then speed of the robot is slow

if slope is large-positive and terrain is moderate then speed of the robot is medium

if slope is large-positive and terrain is smooth then speed of the robot is high

if slope is positive and terrain is very-rough then speed of the robot is slow

if speed is positive and terrain is rough then speed of the robot is medium

if slope is positive and terrain is moderate then speed of the robot is slow

if ...

CONCLUSION AND PERSPECTIVE

As shown in this paper soccer-playing robots can easily be used in education to foster understanding and interest in domains such as artificial intelligence, multimodal systems, engineering and science. Hence we have used soccer-playing robots for learning by doing education of soft computing methods. We have reported about our experience with fuzzy logic.

Our future work is focused on the expansion of soft computing methods to soccer-playing robots based on neural networks, genetic algorithms and image processing. Moreover we plan to use SONY Aibo robots for comparison the soft-computing methods of the different platforms with respect to easy implementation, better search strategies and adaption to machine learning. It is planned to report our results at international conferences and distribute new results via our website.

References

- [Baum 2000] Baum, D.: Dave Baum's Definitive Guide To LEGO MINDSTORMS. Apress (2000)
- [Baum et al. 2000] Baum, D., Gasperi, M., Hempel, R., Villa, L.: Extreme MINDSTORMS. An Advanced Guide to LEGO MINDSTORMS. Apress (2000)

- [Christaller et al. 2001] Christaller, T.; Indiveri, G.; Poigne, A. (eds.): Proceedings of the Workshop on Edutainment Robots 2000, 27th - 28th September 2000, St. Augustin, Germany, GMD Report 129, GMD-Forschungszentrum Informationstechnik GmbH (2001)
- [Erwin 2001] Erwin, B.: Creative projects with LEGO MINDSTORMS. Addison-Wesley (2001)
- [Knudsen 1999] Knudsen, J. B.: The Unofficial Guide to LEGO MINDSTORMS Robots. O'Reilly (1999)
- [Koch 2003] Koch, B.: Einsatz von Robotikbaukästen in der universitären Informatikausbildung am Fallbeispiel "Hamburger Robocup: Mobile autonome Roboter spielen Fußball". Diplomarbeit, Fachbereich Informatik, Universität Hamburg (2003)
- [Kroeger et al. 2000] Kroese, B.; van der Boogaard, R.; Hietbrink, N.: Programming robots is fun: Robocup Jr. 2000. In: *Proceedings of the Twelfth Belgium-Netherlands AI Conference BNAIC'00* (2000) 29-36.
- [Lund 1999] Lund, H.H.: Robot Soccer in Education. In: *Advanced Robotics Journal*, 13:8 (1999) 737-752.
- [Lund et al. 1999] Lund, H.H.; Arendt, J.A.; Fredslund, J.; Pagliarini, L.: Ola: What Goes Up, Must Fall Down. In: *Journal of Artificial Life and Robotics*, 4:1 (1999)
- [Lund and Pagliarini 2000] Lund, H.H.; Pagliarini, L.: RoboCup Jr. with LEGO MINDSTORMS. In: *Proceedings of International Conference on Robotics and Automation (ICRA2000)*, New Jersey: IEEE Press (2000)
- [Möller 1995] Möller, D.P.F.: Fuzzy Systems in Modelling and Simulation. In: Breitenacker, F., Husinsky, I. (eds.): *Proceedings of the 1995 EUROSIM Conference* Elsevier Publications, Amsterdam (1995) 55-64
- [Müllerburg 2001] Müllerburg, M. (ed.): Abiturientinnen mit Robotern und Informatik ins Studium, AROBIKS Workshop Sankt Augustin, Schloss Birlinghoven, 14. - 15. Dezember 2000, GMD Report 128, GMD

- Forschungszentrum Informationstechnik GmbH (2001)

- [Papert 1980] Papert, S.: Mindstorms: Children, Computers and Powerful Ideas. New York: Basic Books (1980)

- [Stone and Veloso 1998] Stone, P.; Veloso, M.: Towards Collaborative and Adversarial Learning: A Case Study in Robotic Soccer. In: *International Journal of Human Computer Studies*, 48 (1998)

AUTHOR BIOGRAPHIES

BIRGIT KOCH was born in Herzberg, Germany and studied Computer Science at the Universities of Hildesheim, Balearic Islands and Hamburg. She obtained her MSC degree in 2003 and started working as research assistant at the Institute of Computer Engineering at the University of Hamburg. Her current research interests include mobile autonomous robots, RoboCup, socionic and gender. Her E-Mail address is koch@informatik.uni-hamburg.de and her website can be found at <http://www.informatik.uni-hamburg.de/TIS/>.

DIETMAR P.F. MÖLLER was born in Preetz, Germany and studied at the Universities of Mainz, Bonn and Bremen, where he studied Electrical and Control Engineering. He obtained his doctoral degree in 1980 and in 1985 a degree in Medicine. From 1985 to 1991 Dr. Möller lead the anaesthesia division of Dräger AG in Lübeck, Germany. 1991 he became Full Professor for Computer Science at the Technical University of Clausthal. In 1998 he was elected from the University of Hamburg as Full Professor and Chair of Computer Engineering. Since 1998 he also holds the position as Adjunct Professor at the California State University Chico. His E-Mail address is dmoeller@informatik.uni-hamburg.de and his website can be found at <http://www.informatik.uni-hamburg.de/TIS/>.

Field Patterns for the RoboCupJunior League? - A Car-Park Problem with LEGO Mindstorms Robots

Thomas Oelkers, Birgit Koch and Dietmar P.F. Möller
Universität Hamburg, Fachbereich Informatik,
Arbeitsbereich Technische Informatiksysteme,
Vogt-Kölln-Str. 30, 22527 Hamburg
E-Mail: {6oelkers, koch, dmoeller}@informatik.uni-hamburg.de

KEYWORDS

Mobile Autonomous Robots, RoboCup, Field Patterns, Self-Localization

ABSTRACT

In the RoboCupJunior league the robots on the soccer field may use the whole space of the field, but to get good results in playing soccer the critical element is that the robots should always know where they are located. Unfortunately a field with color tone is used where it is not possible to make an approximation of the position parallel to the goals as long as only light sensors are used. This paper investigates the use of different field patterns in solving a car-park problem for LEGO Mindstorms robots. Solutions based on guide lines, grid patterns, chessboards and color-tones are described, tested and compared to handle the problem of knowing the position and orientation of a mobile robot situated in a structured unstable environment.

INTRODUCTION

Today robots are used in many different scenarios. It is the goal of the scientists to improve the methods and techniques, which are useful to the society (e.g. mine cleaning robots, Mars-expedition robots, household robots). In order to advance the research in robotics, numerous competitions of different categories take place in the world, in which researchers present their own approaches. Examples of such competitions are disciplines in RoboCup. Each year small robots build for example by

LEGO Mindstorms [Lego Mindstorms WWW] equipped with different sensors and motors can be admired in the RoboCupJunior league [RoboCupJunior WWW].

With LEGO Mindstorms robots different problems can be solved. Thus a student thesis [Oelkers 2002] evolved, in which a LEGO Mindstorms robot should accomplish the following tasks: First to drive into a car-park and second to look for a free space and to park in it. All this should work based on given landmarks on the floor, which have to be recognized by a photosensitive sensor.

THE ROBOTIC CONSTRUCTION KIT

The most important part of the LEGO Mindstorms Robotics Invention System is the RCX unit, a programmable LEGO component, based on a microprocessor of the Hitachi H8-family (H8/3292 processor). The programs for the RCX are written on a usual computer. They will be transferred via an infrared interface to the RCX, that also disposes of the following characteristics: a LCD display for the output of short and simple information, an integrated speaker to produce simple acoustic signals, an infrared interface for communication with other RCX units or a computer, four system timers and a writable memory of 32KB.

THE PARKING TASK

A robot constructed with LEGO Mindstorms should be built and programmed in such a way, that it can cope completely autonomously with the following tasks (Figure 1). The robot must

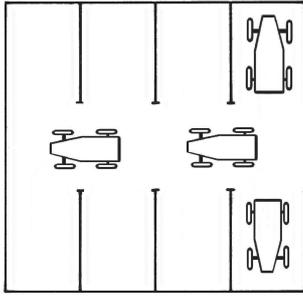


Figure 1: A typical situation in a car-park



Figure 2: The robot for the parking task

been able to drive into a parking lot. Before driving in, the robot must stop and look, whether a parking lot is free. If a parking lot is occupied, the robot must recognize this and look for another one. If a parking lot is free, it must drive in correctly. If both parking lots are occupied, the robot must drive to the next parking row, where it has to proceed like before. If the robot is reaching the end of the parking lot, it must recognize this: it must not drive beyond this point. If the end of the parking lot is reached and no free parking lot was found, the robot must leave the parking lot. While driving the robot must not touch or damage another vehicle or drive over forbidden areas.

EQUIPMENT OF THE PARKING ROBOT

The parking robot (Figure 2) is a vehicle with three wheels. Two different motors activate two parallel wheels. A single wheel which is not powered serves for the support. The wheels can be differently arranged.

Contact sensors recognize physical contacts, they know only two conditions: pressed or not pressed. The robot is equipped with three of

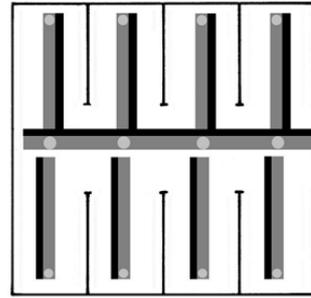


Figure 3: Leading lines for the robot

these sensors, one in the back and two in front, because the robot should be able to differentiate between left and right side.

Light sensors are active sensors, because they need energy from the RCX unit. Light sensors are not able to recognize different colors, they only measure the intensity of the light. Unfortunately, this type of sensor is easily influenced by other sources of light.

The **programming environment** used for the robot was leJOS ("LEGO Java operating system"), an implementation of a Java virtual machine for the RCX unit.

PROPOSALS FOR SOLUTIONS

In this paper a structured unstable environment is used that is suited to the needs of the robot. It is sufficient if the robot gets enough information about the environment to solve its task. For further information on environmental modulation see [Nehmzow 2000][Dudek and Jenkin 2000].

Guide Lines

The vehicle should drive on a grey line (Figure 3). Circles on the floor mark the individual landmarks, i.e. the individual parking spaces, that the robot must recognize. The robot should be limited in its freedom of movement as far as possible. Only those places which serve the purposes of the task should be reachable. In case of leaving the grey line, a control routine must be activated immediately, which leads the robot fast and easy back to the grey line. With this behavior, it is guaranteed that the vehicle drives straightforward and remains strictly on the way which leads to the individual destinations.

Problems

There are some problems, that occur with the lines approach:

- unevenness of the floor and thus wrongly noticed brightness
- because of too many "IF instructions" the measurements become more unreliable because they take place more irregularly
- landmarks for park row recognition are sometimes counted double or not at all
- reflectors are not always recognized

Advantages

To drive a vehicle on a given line can be compared with usual rail guidance. Without dirt on the guidelines, which would disturb the data taken up of the light sensor, it is very sure that the vehicle does not leave the way. The danger to drive over forbidden lines or bump against another parking vehicle is relatively small. Additionally the software is more or less easy to implement. The equipment with sensors is limited to a minimum. Thus, the vehicle can be guided quite simple. All tasks are actually fulfilled.

Disadvantages

As surely and reliably this approach of the solution works and as easily it is implemented, it is extremely inflexible. If two vehicles meet on the center line, it is not possible for them to avoid each other. The robot in this experiment has to leave the parking lot in order to make place for the other vehicle. Additionally the robot cannot drive around obstacles. The robot is practically joining with the line and cannot deviate from it. A possible solution would be to provide two additional leading lines which run alongside to the left and right of the main line. In order to avoid another vehicle, the robot can look for the guideline on the right side and evade to it, while the other vehicle evades to the other side. After this manoeuvre both robots go back to the main leading line.

Grid Pattern

Now the entire floor, on which the robot moves around, is filled with a grid (Figure 4), so that the soil consists of many squares equal in size. The idea is to use the individual small boxes as indication for coordinates. Each box stands for two values. The first value describes the distance to the left, the second value the distance to the

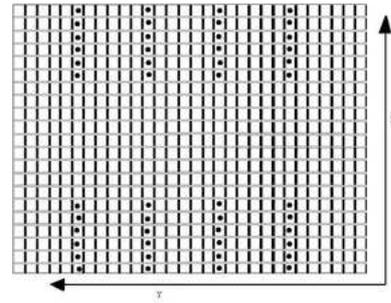


Figure 4: The grid on the floor with x- and y-axis

bottom edge of the field. If the robot counts each crossed line it always knows at which place (in which square) it is. This solution is substantially more flexible. It permits the robot to execute more manoeuvres. Additionally the lines serve the robot to navigate straightforward. In order to differentiate the horizontal lines from the vertical ones, they have different colors. The lines parallel to the x-axis are darker, the ones parallel to the y-axis are brighter than the floor.

Problems

There are some problems, that occur with the grid approach:

- lines are not always recognized reliably and therefore x- and y-values are not always correctly
- 90° turns with a sufficient result are very hard to implement; the robot turns too far or not far enough and therefore leaves its way

Advantages

The vehicle can move more freely and more flexibly than in the line approach. If the robot leaves a grid line it will immediately look for a new line by itself. Avoiding another vehicle is relatively simple. Special landmarks are not needed. Only a grid pattern on the floor and the knowledge of the starting point are necessary to navigate the robot. Only two different colors are used in order to distinguish the x- and y-coordinates. Thus, errors by unwanted false values are prevented completely.

Disadvantages

The software becomes more complex. If the robot makes an error counting the lines (which often happens during turns) it has no possibility to correct it. The robot looks on his x- and y-values to discover a parking lot. If the values do

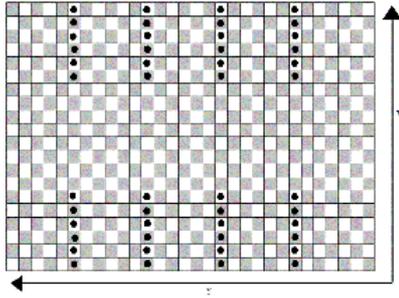


Figure 5: The chessboard on the floor

not correspond with reality, the robot will stop in the wrong place. The further a park row is away, the more likely it is to have a miscount. The rotation is particularly incorrect. During avoidance of another vehicle, a robot must turn four times. Failures with each turn sum up, so at the end the result is rather dissatisfying. Driving parallel to the y-axis and driving parallel to the x-axis must be treated differently. Although the grid works with only two colors (grey and silver), in reality three colors are used, since the floor has the color white. Therefore the robot has to recognize three colors to be able to distinguish the floor from a silver or a grey line.

Chessboard

In this approach the robot should move over a black and white tiled field. The two colors selected for the tiles are not important, as long as the light sensor of the robot can clearly differentiate between them (Figure 5). With the help of the small boxes, the robot is able to move with the same flexibility like in the grid approach but with a better accuracy. The chessboard field seems to be very similar to the grid pattern but there are a few fundamental differences: The chessboard field operates with only two colors which makes the distinction safer and faster. In addition, a square is completely filled out with a color. There are no thin lines, only a special floor color. Hence the risk that a square is not recognized due to irregular measurements is reduced.

Problems

The following problems occur with the chessboard approach:

- the robot drives straight diagonal lines over the field instead of straightforward

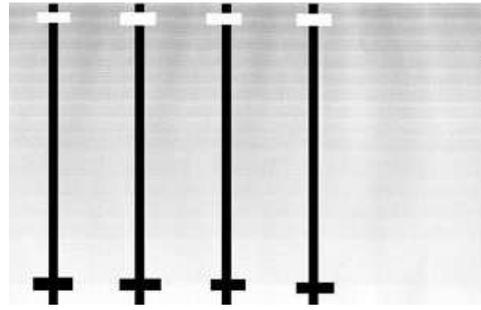


Figure 6: The color tone on the floor

- during a turn of 90° to the left or right very often incorrect results arise
- other vehicles are not always recognized reliably

Advantages

In the chessboard approach only two colors are used. Regardless of the direction the robot is driving, it can always use the same routine. A single method is sufficient to realize all rotations. A diagonal line (45° angle) is reliably possible.

Disadvantages

Rotations remain to be a problem. They are not always completed reliably. That is because of the fact that the robot is shifted a little bit when it is stopped before a rotation. Thus the radius that the light sensor pulls over the tiles can be shifted easily so that the number of color changes vary.

Color Tone

The fourth solution consists of a field with a color tone. As shown in Figure 6, the field is painted with a color tone parallel to the y-axis. The color tone runs from dark grey to white. Parallel to the x-axis four black lines are painted, which have an approximate thickness of 1.5 cm. They serve as park row marks and can also help the robot during the drive-in. At the end of the line another mark must exist, which highlights the end of the parking lot. These end marks are made with silver strips. The idea is the following: the robot needs only one sensor to measure the hue of the floor. As long as it is searching for a parking lot, it may drive only straightforward, parallel to the y-axis. With the help of the color tone, this is simply possible. The robot has to be programmed in such a way that it may drive only on a certain color brightness. The values depend on the field and can only be evaluated by

tests. If a color which is not permitted is read the robot must correct its way immediately. The color lines, which mark the parking lot, are of lower intensity than the lowest value of the color tone. Therefore the lines can always be clearly identified. Thus a good orientation seems to become possible along the x-axis.

Problems

There are some problems, that occur with the color tone approach:

- the color tone is not continuous; sometimes brighter or darker values are taken on the same parallel
- the sensor data of the light sensor is quite unreliable, since it depends on the battery power

The main problem is the bad color tone on the floor. The robot does not really drive between two invisible leading lines. Sometimes it corrects the driving direction, while it does not behaves like this a few millimeters later. Because the interval of the permitted values is very small little deviations are fatal. Maybe this problem can be reduced if a color tone with a stronger gradation would be provided. The more clearly the gradations on the field can be recognized by the robot the more accurately it can move.

Advantages

A small number of landmarks is needed to mark the parking rows. The color tone alone gives the robot the information in which x-position it is. Inaccuracies of the measurements have only a small effect on the process. The best advantage is the simplicity of the algorithm. It requires only small adjustments and tests until the robot supplies satisfying results.

Disadvantages

Exact positioning along the y-axis is not possible. The positioning at the x-axis is possible but not as exact as in the two previous solutions. Compared to the other approaches the robot is more flexible while moving on the floor. In contrast to the other approaches in this solution avoiding is possible within the parking lot, but the danger to over-drive a line or to push a parking vehicle is very big. Avoiding another vehicle is still difficult to accomplish. However testing generates extremely unwanted results. Most likely the color tone must be of better quality.

Comparison Of The Different Methods Of Orientation

After adjustment and correction each method was tested 100 times counting the errors. An error occurred when the robot got off its actual line, a turn was dissatisfied, a marking was overlooked or counted twice and some other errors. On the basis of these results the different methods were evaluated. It is interesting that the simplest solution (the latter) showed the most reliable results. The more complex solutions had to cope with substantially higher error rates. Most errors that appeared with the first and the last solution could be repaired with small adjustments of the algorithm, so that they worked nearly free of errors. Unfortunately the grid pattern and the chessboard showed a substantially higher error rate. On the grid pattern the turns were very unreliable. Possibly another algorithm could give better results. However the selected light sensors did not meet the demands. The measured light values were unreliable and not very differentiated. The most significant problem on the chessboard was that the vehicle counted markings double or not at all. In this case a marking was a change between black and white. Furthermore the robot often got off the way. It drove straightforward but changed the trace sometimes, so that the x-coordinate was not correct any longer. This had consequences on the parking. The vehicle stopped too early or too late. Finally it can be said that the solution with the chessboard showed the highest number of errors. On the other hand it was very flexible. When the vehicle got off the way during the experimental phase of the guide line method it did not find the way again. On the chessboard and on the grid pattern the robot adjusted itself back to the right path again. The idea of the chessboard approach is certainly improvable. More exact positioning would be possible with a better control algorithm and better sensors.

CONCLUSION AND PERSPECTIVE

Each solution in robotics has its individual advantages and disadvantages. It is important to consider in which scenario a solution is needed. The main problem of the study was that the robot should drive autonomously without human help over the floor. The different solutions can be evaluated from different points of view. If one

would create a real world parking system, then the line approach would be preferred although it is very inflexible. However it is very reliable and simple. There are only several places on which the vehicle may be located. A chessboard or a grid pattern would not add advantages. In the opposite these approaches are more error-prone. But in a real world parking lot errors are not acceptable. This is different on a soccer field: the robots may use the whole space of the field. For playing soccer an accurate location is less important than flexible driving. In this case the solutions of the grid pattern or chessboard are more suitable. Unfortunately in the RoboCupJunior league a field with color tone is used. It is not possible to make an approximation of the position parallel to the goals. On a chessboard this problem is not given. Both x- and y-coordinates can be determined. To avoid that errors sum up, special control places should be painted. If a robot crosses this field, it corrects its x and y-values. Other applications would be in the industry. Here robots are used to carry loads from place A to place B. If the floor would be tiled like a chessboard, then a transport to any place would be possible. In this case, a grid pattern or a chessboard would be quite conceivable.

References

- [Dudek and Jenkin 2000] Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics. USA: Cambridge University Press (2000)
- [Koch 2003] Koch, B.: Einsatz von Robotikbaukästen in der universitären Informatikausbildung am Fallbeispiel "Hamburger Robocup: Mobile autonome Roboter spielen Fußball". Diplomarbeit, Fachbereich Informatik, Universität Hamburg (2003)
- [Lego Mindstorms WWW] The LEGO Mindstorms website. <http://www.legomindstorms.com/>
- [Nehmzow 2000] Nehmzow, U.: Mobile Robotics: A Practical Introduction. London: Springer-Verlag (2000)
- [Oelkers 2002] Oelkers, T.: Parksysteme mit Lego-Mindstorms. Studienarbeit, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Technische Informatiksysteme (2002)
- [RoboCupJunior WWW] RoboCupJunior website: <http://www.robocupjunior.org>
- [Seeboerger-Weichselbaum 2002] Seeboerger-Weichselbaum, M.: Mindstorms: Das Einsteigerseminar Java 2. Bonn: bhv (2002)

AUTHOR BIOGRAPHIES

THOMAS OELKERS studies Computer Science at the University of Hamburg. He wrote his studies thesis about Park systems with LEGO Mindstorms robots. Currently he writes his diploma thesis about fuzzy logic with LEGO Mindstorms robots. His E-Mail address is 6oelkers@informatik.uni-hamburg.de.

BIRGIT KOCH studied Computer Science at the Universities of Hildesheim, Balearic Islands and Hamburg. In 2003 she started working as research assistant at the Institute of Computer Engineering at the University of Hamburg. Her current research interests include mobile autonomous robots, RoboCup, socionic and gender. Her E-Mail address is koch@informatik.uni-hamburg.de and her website can be found at <http://www.informatik.uni-hamburg.de/TIS/>.

DIETMAR P.F. MÖLLER studied at the Universities of Mainz, Bonn and Bremen, where he studied Electrical and Control Engineering. In 1998 he was elected from the University of Hamburg as Full Professor and Chair of Computer Engineering. His E-Mail address is dmoeller@informatik.uni-hamburg.de and his website can be found at <http://www.informatik.uni-hamburg.de/TIS/>.

Fundamentals and Case Studies for a Modeling and Simulation Model Curriculum

Dietmar P. F. Möller¹⁾

Roy E. Crosbie²⁾, Ralph C. Hilzer²⁾, John J. Zenor²⁾

¹⁾University of Hamburg, Faculty of Computer Science, Chair Computer Engineering
Vogt-Kölln-Str. 30, D-22525 Hamburg, Germany

²⁾College of Engineering, Computer Science, and Technology
California State University,
Chico, CA95929-0003, USA

ABSTRACT

Since several years, the authors suggest that the development and publication of a Model Curriculum for MS programs in Modeling and Simulation would facilitate the development of such programs. This paper presents a first draft of a Model Curriculum developed on the basis of a recent published textbook of Mathematical and Computational Modeling and Simulation at the University of Hamburg (UHH), Germany and the California State University, Chico (CSUC). The aim of the draft is to stimulate further discussion in the M&S community with the goal of arriving at a generally acceptable outline that can serve as a guideline for new Modeling and Simulation programs.

1 INTRODUCTION

As the demand for Mathematical and Computational Modeling and Simulation professionals continues to increase, much attention has been focused recently on what constitutes a proper educational preparation for a career in Mathematical and Computational Modeling and Simulation. At the Western Simulation Conference in 2000 (WSC 00) Dr. Roy Crosbie proposed the development of a Model Curriculum for a MS degree in Modeling and Simulation (M&S) to assist schools planning new programs in the field. This suggestion was based on experience in the 1980s of developing a new BS program in Computer Engineering, a task that was greatly facilitated by the availability of a model curriculum for Computer Science and Engineering (CSE) published by IEEE-CS and ACM.

The task of defining a curriculum for a MS degree in Mathematical and Computational Modeling and Simulation is particularly difficult, partly because M&S is normally not seen as a discipline in its own right in

most universities. It is usually regarded as a fragmented subject with concepts in a range of disciplines including, but not necessarily restricted to computer science, various engineering disciplines, mathematics, business, natural and physical sciences, medicine, pharmacy and social sciences. Furthermore, the range of applications of M&S is so wide that almost any university department could possibly offer a course on M&S applied to its particular discipline. This results in a situation in which course offerings relevant to M&S, and that might be included in an M&S degree program, tend to be scattered throughout the university, with no central responsibility for M&S as a discipline.

A recent White House report identified computational modeling and simulation as one of the key enabling technologies of the 21st century. Its application is universal.

One way of handling this various individual courses will go through departments of interdisciplinary degree, which is the mechanism currently used at US universities, which can offer a flexible MS in Interdisciplinary Studies (MSIS) that allows a student, with faculty support, to devise a program of study from the course offerings of multiple departments that can address specific academic goals not provided for in the regular graduate programs of the university. The MSIS in Simulation Science at CSUC requires 9 semester units (3 courses) of M&S fundamentals covering methodology and tools and techniques, and a project of at least 3 units on a topic related to M&S. The student selects the remaining courses without specific restrictions as to department or discipline as long as they conform to the student's academic goals and receive faculty approval. In comparison to the CSUC model at UHH students enrolled in Computer Science or Engineering can select special courses on Modeling, Simulation and Optimization, each of which have 3

credit units, as well as specific seminars and projects on M&S topics.

Another approach is for universities to combine forces, and UHH and CSUC are also pursuing this route in a consortium of 3 European and 3 US universities that are together running an international graduate program in M&S making use of on-line based course material, faculty exchange and student exchanges in order to broaden the base of M&S courses available to students from a particular campus. The basis for such a graduate program in M&S is based on the textbook “Mathematical and Computational Modeling and Simulation: Fundamentals and Case Studies” published by Springer Pub. Heidelberg.

Some choices have been made in selecting the material of this book. First the methodology of mathematical and computational modeling is described, because this represents the largest portion of system analysis. In addition, the mathematical background describing real-world systems is introduced on a basic level as well as on a more advanced one and its correspondence with the respective modeling methodologies. Second the most interesting simulation software systems at the language and logic level are described, as well as their use in several case study examples. However, a textbook can not describe all available simulation software systems in detail, for this reason students are referred to use the specific written material such as textbooks, reference guides, user manuals, etc., as well as the web based information addressed to the several simulation software systems. Thirdly, an algorithmic approach to ill defined and distributed systems based on the respective mathematical frameworks are introduced.

The purpose of this book is to expose undergraduate and graduate students to the use of mathematical and computational modeling and simulation as a basis for developing an understanding of the response characteristics of a broad class of real-world systems. Mathematical and computational modeling is based on systems theory as a mathematical form of representation while building models of real-world systems. The simulation methodology behind is used for a better understanding of the time dependent transient behavior of the complex models developed, which are in relation to the continuous-time and discrete-time description. The subject matter of the textbook can be considered to form an introduction to the methodology of mathematical and computational modeling of real-world systems, as well as into simulation software systems, to gain experience, which result from the different application domains introduced as case study examples in this book.

The nature of the material in the book can be seen more or less difficult, if the reader is new to such approach, which also is due to the fact, that mathematical

and computational modeling and simulation is a multi-disciplinary domain, founded in computer science, engineering, mathematics, physics, chemistry, biology, life science, etc. The material may not be read and comprehended either quickly or easily. This is why specific case study examples, from the various disciplines, have been embedded due to the related topics of system theory representation of the material, to master the material, at least for most individuals of the several scientific disciplines. It is assumed that the reader has some previous background in mathematics through calculus, differential equations, Laplace transforms, and matrix fundamentals. The most common simulation software systems will be introduced and their performance will be discussed based on several case study examples. But real-world systems often are ill defined and the important parameters which should be known for modeling may not be known and/or not measurable, which call for parameter identification methods to estimate unknown parameters. Moreover, virtual reality and soft computing methods show up recently which are now being added on as part of the methods used for modeling and simulating real-world systems, which are used for the respective examples and case studies.

Whatever the administrative basis for a graduate program in M&S, it can only be helped by the availability of a set of generally agreed guidelines on the structure and composition of the degree. Having said this, it is important that the guidelines should be capable of flexible interpretation, avoiding a “*one size fits all*” approach. This was achieved in the CSE guidelines by specifying course content in the form of subject areas rather than complete courses. This allows the program developers to mix and match different subject areas to produce individual courses that best fit the needs and resources of a particular institution. As long as the mix of courses provides for all of the subject areas that are required, and adequately provides for those that are optional, the program conforms to the guidelines.

2 COMPONENTS OF THE M&S DEGREE

The referenced WSC 00 paper of Dr. Roy Crosbie (Crosbie, 2000) recommended that the M&S degree contain the following four major components:

- *Fundamentals of M&S;*
- *Simulation Tools and Techniques;*
- *Applications of M&S;*
- *M&S Project.*

This initial draft assumes an MS program of 30 semester units (equivalent to approximately 10 semester courses). Students entering the program are assumed to have an adequate background in basic topics in

computer science, mathematics, physics and their primary BS discipline.

Although the guidelines do not prescribe the number of units to be allocated to each of the above areas, it is recommended that at least two courses (6 semester units) be allocated to each of the first three areas and 3 semester units to the project. This accounts for 21 units and leaves 9 units (3 courses) at the discretion of the program or for use as electives.

Henceforth the textbook covers the previous mentioned scopes due to the content and time schedule. It can be used in courses in various ways. It contains more material than could be covered in detail in a quarter-long (30-hour) or semester-long (45-hours) course, leaving instructors with the possibility of selecting their own topics and add on own case study examples. Sections denoted with an asterisk report on advanced topics and can be skipped in a first reading or in undergraduate courses. Moreover, it covers the fundamentals of M&S, as well as simulation tools and techniques, applications of M&S and several M&S projects, introduced as case study examples for the variety of M&S application domains.

The book can also be used for self-study or as a reference for graduate engineers, scientists and computer scientists for training on the job or in graduate schools.

3 DETAILS OF PROGRAM CONTENT

The four Subject Areas (SA) for a MS in M&S are as follows:

1. SA1: Fundamentals of Modeling and Simulation
2. SA2: Tools and Techniques for Modeling and Simulation
3. SA3: Applications of Modeling and Simulation
4. SA4: Modeling and Simulation Project

Each subject area is divided into a number of sub-areas. Each sub-area can be regarded as the basis for a single course, part of a course, or even more than one course. Topics can be taken from different sub-areas and combined to form a course and the topics in a given sub-area may be distributed between different courses. The aim is to produce a general topology for the material that should fit into the course content of the actual degree program. Because of the wide variety of simulation applications, as covered by the textbook, it is likely that individual programs will emphasize different aspects of the material, and may find it necessary to include additional material.

We make no claim that these subject area definitions are complete. They are merely offered as a starting point for discussion and elaboration. We

would welcome all comments, both positive and negative, including suggestions for alternate ways of arriving at a set of guidelines that are both sufficiently prescriptive to provide positive guidance to program developers and sufficiently flexible to avoid undue restrictions on the development of innovative programs in an area that has so many facets and multi-disciplinary aspects.

4 CONCLUSIONS

A first version of a model curriculum for a graduate degree in Modeling and Simulation is proposed. We hope that this can be used as a basis for wide-ranging discussions that will lead to a published version sponsored by interested professional bodies such as ACM, SCS and IEEE-CS.

5 REFERENCES

- Crosbie, R. E. 2000. A Model Curriculum in Modeling and Simulation: Do We Need It? Can We Do It? In *Proceedings of the 2000 Winter Simulation Conference*, Piscataway, New Jersey:Institute of Electrical and Electronics Engineers.
- Möller, D.P.F. 2003. *Mathematical and Computational Modeling and Simulation: Fundamentals and Case Studies*. Springer Publ. New York

6 AUTHOR BIOGRAPHIES

DIETMAR P. F. MÖLLER is a Professor of Computer Science and Computer Engineering at University of Hamburg, Germany. He is Director of the McLeod Institute of Simulation Sciences at UHH. His current research interests include simulation methodology, virtual and augmented reality, embedded systems, mobile autonomous systems and robots, and e-Learning and e-work.

ROY E. CROSBIE is a Professor of Electrical and Computer Engineering at California State University, Chico. He is Director of the McLeod Institute of Simulation Sciences at CSUC. His current research interests include simulation methodology and distributed and real-time simulation.

RALPH C. HILZER is a Professor of Computer Science at California State University, Chico. His interests include simulation of parallel systems.

JOHN J. ZENOR is a Professor of Electrical and Computer Engineering at California State University, Chico. He is Associate Director of the McLeod Institute for Simulation Sciences at CSUC. His interests include distributed simulation and real-time operating systems.

VIRTUAL REALITY FRAMEWORK FOR SURFACE RECONSTRUCTION

Dietmar P. F. Möller

Chair Computer Engineering

& McLeod Institute of Simulation Sciences, German Chapter at University of Hamburg

Email: dietmar.moeller@informatik.uni-hamburg.de

University of Hamburg, Department Computer Science

Vogt-Kölln-Str. 30, D-22527 Hamburg, Germany

Abstract. We briefly introduce the methodology of virtual reality as a framework for surface reconstruction based on morphing. Henceforth the scope of our work should give a deeper insight into complex research work showing the power of deformable models, as part of morphing, in a virtual reality framework in the spacious areas of the geological application domains. Based on that facts we will give a case study examples of virtual reality application in the field of applied geotechnology. Currently we are working on other applications of morphing in geology.

1. Virtual Reality applied to geology

Applying the virtual reality methodology to the geological domain could be stated as combining distributed virtual environments, in order to support collaboration among team members working with space distance, developing plans and procedures, doing measurements and data processing in geological procedures, research projects, geo-technology oriented support systems development and evaluation etc. in order to attempt to manage new investigations and organizations in a collaboratively manner, as it is needed in global as well as international project development.

One of the most interesting new paradigms in virtual reality methodology in this domain is that three dimensional representations are not only the lonely possibility of a setting.

Many virtual applications in geology, if not already now, will in future make use of specific graphics. The virtual domain will be visualized in space, which means in terms of three dimensions, and time. People in charge with virtual reality in the geological domain are able to interact with space and time, e.g. like walking through the water and wastewater underground infrastructure for inspection of safety and security at the walls intima, or interacting with other geological disciplines for consultancy through a graphical user interface in the manner of computer supported cooperative work, as well as designing the plastic view of sustainable interventions in underground infrastruc-

ture. The interweaving of functionality, distribution, efficiency, and openness aspects is very noticeable in computer graphics. The virtual space is graphically visualized flamboyance and for the most part the people in charge with the virtual space application domain should see the same image.

Therefore, for virtual reality applications, a three-dimensional, multi-user virtual reality tool for the geological application domain as been developed, consisting of the following main components:

- space ball and cyber gloves for tactile interaction in virtual space
- head mounted devices for visual interaction in virtual space
- 3-dimensional geometric body creation and motion methodology for "virtual space feeling" capability
- 3-dimensional visual interactive system for definition, manipulation, animation and performance analysis of geological geometric bodies
- object oriented data base for efficient data management in virtual reality applications
- hardware for the power of computing in space and time
- objects organization into inheritance hierarchies for virtual reality system transparency

When geological objects are created, they inherit the properties and verbs of their ancestors. Additional verbs and properties as well as specializations of inherited components may be defined to give the new object its unique behavior and appearance.

Based on that assumptions a virtual reality simulator for the geological application domain has been build up.

2. Morphing as basis for Visualization in Geology

The presentation of process states is of importance, which has to be realized time dependent, bringing together real scenarios as well as virtual scenarios of the geological project under realization as real research project, in order to find out e.g. optimal geometries, based on Non Uniform Rational B-Splines (NURBS).

This special kind of B-Spline representation is based on a grid of defining points $P_{i,j}$, which is approximated through bi-cubic parameterized analytical functions.

$$P_{i,j} = \left\{ \begin{array}{cccc} P_{1,1} & P_{1,2} & \cdots & P_{1,n} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m,1} & P_{m,2} & \cdots & P_{m,n} \end{array} \right\}, P_{i,j} = (x, y, z)$$

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}}$$

$$0 \leq u, v \leq 1$$

This method allows to calculate the resulting surface or curve points by varying two (surface) or one (curve) parameter values u and v of the interval $[0, 1]$, respectively, and evaluating the corresponding B-Spline basis function $N_{i,p}$.

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases},$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

$$U = \{u_0, \dots, u_m\}, u_i \leq u_{i+1},$$

V analogous

As the parameter values u and v can be chosen continuous, the resulting object is mathematically defined in any point, thus showing no irregularities or breaks.

There are several parameters that adjust the approximation of the given points and thus changing

the look of the geological object under test (description), hence, if needed, interpolation of all points can be achieved.

First of all, the polynomial order describes the curvature of the resulting surface or curve, giving the mathematical function a higher level of flexibility. Second, the defining points can be weighted according to their dominance in respect to the other control points. A higher weighted point influences the direction of the surface or curve more than a lower weighted. Further more, knot vectors U and V define the local or global influence of control points, so that every calculated point is defined by smaller or greater arrays of points, resulting in local or global deformations, respectively.

NURBS are easy to use, as modeling and especially modifying is achieved by means of control point movement, letting the user adjust the object by simply pulling or pushing the control points (Figure 1).

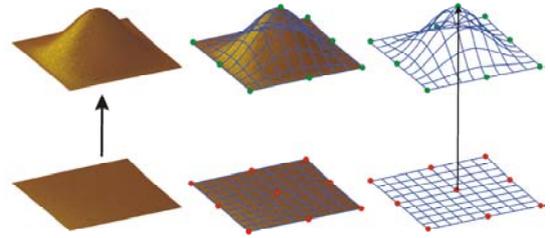


Fig. 1 Modeling and modification of a NURBS surface

Based on these concepts a methodology to interpolate a given set of points, for example the results of scanned data of application domain surface measurements, has been developed. Huge sets of scattered data points of a geological object are used to generate the resulting object, outgoing from the very simple geometric object of a cylinder.

Using multiple levels of surface morphing, this multi level B-Spline Approximatton (MBA) adjusts a predefined surface, i.e. a flat square or a cylinder. Constraints like the curvature or direction at special points can be given and are evaluated within the algorithm.

Based on OpenGL, a quasi standard for three dimensional modeling and visualization, we are able to create geometric medical bodies of every shape and size and move them in real time.

3. Deformable Models in Geological Surface Reconstruction

Mathematically geometric (geological) subjects can be interpreted as embedded contour within an image plane

$$(x, y) \in \mathbb{R}^2$$

of a virtual reality framework concept. The contour itself can be assumed as

$$\exists(s) = (x(s), y(s))^T$$

where x and y are the coordinate functions and $s \in [0, 1]$ the parametric domain. The shape of a contour subject to an image $I(x, y)$ can be described [McInerney et al., 1999] by the functional

$$\mathfrak{J}(\exists) = E(\exists) + \Gamma(\exists)$$

The functional given above can be interpreted as representation of the energy of the geological contour. Hence the final shape of this geological contour corresponds to a minimum of energy. Due to that the first term of the functional given above can be introduced as internal deformation energy

$$E(\exists) = \int_0^1 \Lambda_1(s) \left| \frac{\delta \exists}{\delta s} \right|^2 + \Lambda_2(s) \left| \frac{\delta^2 \exists}{\delta s^2} \right|^2 ds$$

This equation describes the deformation of a stretchy, and flexible geological contour, with $\Lambda_1(s)$ as tension of the geological contour and $\Lambda_2(s)$ as rigidity.

In accordance with the calculus of variations, the geological contour $\exists(s)$, which minimizes the energy $\mathfrak{J}(\exists)$ must satisfy the Euler-Lagrange equation [McInerney et al., 1999]

$$-\delta/\delta s (w_1 * \delta \exists / \delta s) + \delta^2 / \delta s^2 (w_2 * \delta^2 \exists / \delta s^2) + \nabla P(\exists(s, t)) = 0$$

The vector partial differential equation, introduced above, describe the balance of internal and

external forces when the geological contour rests at equilibrium. Therefore the first two represent the internal stretching and bending forces respectively, while the third term represents the external forces that couple the contour to the image data.

4. Conclusions

The potential of virtual reality is huge. We only scratched the surface of this important area due to the geological application domain. The potential of morphing contains an incredible number of solutions to different problem depending domains.

5. References

1. Crilly, A.J., Earnshaw, R.A. and Jones, H., Applications of Fractals and Chaos. Springer Verlag, Berlin, 1993.
2. Encarnacao, J.L., Peitgen, H.-O., Saka, G. and Englert, G., Fractal Geometry and Computer Graphics, Springer Verlag, Berlin, 1991.
3. Gilfillan, L. and Harbison, K., Using distributed virtual environments (DVE) for collaborative program planning and management: Problems and potential. In: Proc. VWSIM'98 (Eds.: Landauer, C. and Bellman, K.L.), SCS Publishers, San Diego, 1998, 39-46
4. Möller, D.P.F., Virtual Reality: Simulation Synergy in Laboratories and Outer Space Domains. In: Simualtion: Past, Present and Future (Eds.: Zobel, R. and Möller, D.P.F.), Vol. II, SCS Publishers, Delft, 1998, 64-66
5. Schneider, M., Spatial Data Types for Database Systems. Springer Verlag, Berlin, 1997.
6. Singh, A., Goldgof, D. and Terzopoulos, D., Deformable Models in Medical Image Analysis, IEEE Press, Los Alamitos, USA, 1998.
7. Straßer, W. and Seidel, H.-P., Theory and Practice of Geometric Modeling, Springer Verlag, Berlin, 1989.
8. Yachik, T.R., Synthetic Scene Quality Assessment Metrics Development Considerations. In: Proc. VWSIM'98 (Eds.: Landauer, C. and Bellman, K.L.), SCS Publishers, San Diego, 1

SOFT COMPUTING ANALYSIS AS PART OF MICRO ARRAY DATA ANALYSIS

Dietmar P. F. Möller

Chair Computer Engineering

& McLeod Institute of Simulation Sciences, German Chapter at University of Hamburg

Email: dietmar.moeller@informatik.uni-hamburg.de

University of Hamburg, Department Computer Science

Vogt-Kölln-Str. 30, D-22527 Hamburg, Germany

Abstract. The paper presents the application of soft computing methodology in medicine. Applied to micro array data analysis one may introduce soft computing as part of sensitive networks and classifiers. While sensitized nets enable the conditioning of case study specific classifiers, scientists are able using specific linguistic IF-THEN rules to create appropriate fuzzy sets, that can be helpful for use in micro array data analysis. Henceforth, scientists will be empowered handling this classifiers in situ, phased to their measuring equipment and/or case study specific parameters, under case investigation.

1. Introduction

We live in a world of an imposing complexity and variety, a world where events more or less never repeat exactly. Human-world interaction based on a scientific approach seems to have a normalisation in models, an abstract representation, as a powerful tool to understand real world phenomena. Therefore a big part of scientific work consists in formalization, which yields models of real world systems studied. This task clearly is scientifically oriented, in the sense to gain sufficient understanding of real world phenomena, generating respective representations, based on experiments and observations. Because the scientist attempts to create representations and laws that formalize verified hypothesis concerning real world phenomena.

The formalizations are only useful if they succeed in seizing the essential features of the real world. They permit extrapolation, that allows to generalise, often correctly, from past experience to future events from which we can learn how the real world system can be manipulated for own purposes, which is a kind of uncertainty. In our world which is more or less precise understandable or predictable, we are more conscious of uncertainty, that appears in form of imprecision, vagueness and ill defined, ill separable, and doubtful data. For this kind of uncertainty, or better soft information, we have to learn to understand the intrinsic systems dynamic.

Keeping formalizations running or doing extrapolation, deals with effectively information processing, which is a task, done by computing machines, has been directly introduced as a suitable tool of the scientific approach. But using non-precise information, which is called soft information, e.g. the blood pressure is lower than normal, represented by linguistic terms such as low, high, small, medium, large, big and so on, needs a specific form of computation, which is called soft computing. Soft computing, deals with fuzzy sets, neuronal nets, genetic algorithms, evolution strategy, probabilistic methods etc. Usually, these approaches in uncertainty, combining soft information with conventional scientific methods in a so called ad hoc manner, can be investigated using computation to show the validity of the approaches in relation to the specific case study. Therefore, during the past years, processing of uncertainty or soft information processing had been applied by different disciplines for a large variety in formal representations in the several scientific application domains. Applying soft computing techniques for those formalizations, one can impart an understanding that the formalization itself can not provide. Because soft computing is a collection of methods which can be expressed in terms of algorithms, belonging to the respective disciplines, that has been proved to be of vital importance to progress in all fields of endeavor.

Applying soft information in formalization of complex non-linear real world systems may call for the integration of soft computing system correspondings such as fuzzy sets or neuronal nets, to meet the challenge modeling complex non-linear dynamic systems.

In general, the common problems arising from formalization in science, and especially the possibility of applying in a wide range of scientific research the same methods while solving problems, has improved the co-operation between different disciplines and removed the rigid barriers of the past between them. However, although it seems that engineers and scientists, like physicians, will have the same goals in studying their systems. But there is still an essential difference between them, for-

malizing a real world research problem. For instance, the engineer is interested in system formalization reflecting normal operating conditions. His aim is to use the model in case of normal operating conditions, e.g. for optimized system control, or at least to keep it in a relative close vicinity of safe operating conditions and avoid the danger resulting from the formalized system running out of control. Anyhow there is no outstanding interest for engineers formalizing plants behavior outside its allowable operating conditions range.

In contrast, the scientist, like the physician, is not solely interested in formalization a real world problem under normal conditions. He would prefer that the formal description adequately describes the systems un-nominal behavior, i.e. the systems behavior outside of normal limitations, like pathological states in case of hypertension versus normotension, or gene expression analysis in relation to the type or growth of a tumor, and there are serious limitations. But engineering techniques dealing with uncertainty are sometimes as much statements comparable to cognitive and linguistic sciences as they are about engineering, and hence they are comparable to science, like medicine.

In practice, the formalization of models itself is an iterative process, consisting of measurements at the real world system –if possible–, and computing strategies by changing the structure of the formal description in an effort to closely match the complex dynamic systems behavior. The computing strategy may be based on the category in the nearest neighbor sense, if the adapted representation is close enough to the previous one. In fact the formalization has served its purpose when an optimal match is obtained between computed results and data obtained from the real world system under test.

Soft information processing generate the basic insight that categories are not absolutely clear cut, they belong to lesser or greater degree to that category. Hence soft computing systems break with the tradition, that real world phenomena can be precisely and unambiguously characterized, which means divided into categories, and then manipulated according to precise and formal rules. From the mathematical point of view soft computing means multi-valuedness or multi-valence. Logical paradoxes and the Heisenberg uncertainty principle led to the development of multi-valence, and in the 1930s quantum theorists allowed for indeterminacy by including a third truth value in the bivalent logical framework. Systems scientist Zadeh in 1965 introduced the term fuzzy into the technical literature, and inaugurated a second wave of interest in multi-valued structures –from systems to topologies– extending a bivalent indicator function iA of non-fuzzy subset A of X to a multi-valued indicator or

membership function $m_A: X \rightarrow [0,1]$. This allows to combine multi-valued or fuzzy sets with point-wise operators of indicator functions for the large variety of fuzzy systems.

2. Soft Computing Systems

Soft computing systems is a name for systems with directly relationship to soft computing concepts like fuzzy sets, neuronal nets, genetic algorithms etc. The soft computing concepts based on fuzzy sets can be classified into pure fuzzy systems, Takagi and Sugeno fuzzy systems and fuzzy systems with fuzzification and defuzzification.

2.1 Pure Fuzzy Systems

The basic configuration of a pure fuzzy systems is based on a fuzzy rule base that consists of a collection of fuzzy IF-THEN rules, and the fuzzy inference engine that uses these fuzzy IF-THEN rules in order to determine a mapping output universe of discourse $U \subset \mathbb{R}^n$ to fuzzy sets in the output universe of discourse $V \subset \mathbb{R}$ based on fuzzy principles. Fuzzy IF-THEN rules are of the following form:

$$R(k): \text{ IF } x_1 \text{ is } F_1^{(k)} \text{ AND } \dots \text{ AND } x_n \text{ is } F_n^{(k)} \text{ THEN } y \text{ is } G^k \quad (1)$$

where $F_i^{(k)}$ and $G^{(k)}$ are the respective fuzzy sets, $x = (x_1, \dots, x_n)^T \in U$ and $y \in V$ are input and output linguistic variables, respectively, and $k = 1, 2, \dots, m$.

Each fuzzy IF-THEN rule defines fuzzy set $F_1^{(k)} \times \dots \times F_n^{(k)} \rightarrow G^{(k)}$ in the product space $U \times V$. Let A be an arbitrary fuzzy set in U , then the output determined by each fuzzy IF-THEN rule of equation (1) is a fuzzy set $A \circ R^{(k)}$ in V whose membership function is

$$\mu_{A^{(k)} \circ R^{(k)}}(\mu) = \sup_x \mu_U [\mu_A^{(x)} * \mu_{F_1^{(k)}} \times \dots \times \mu_{F_n^{(k)}} \rightarrow G^{(k)}(x,y)] \quad (2)$$

with $*$ as operator such as MIN, MAX, PRODUCT, or others. μ_A is used to represent the membership function of a fuzzy set A .

The final output of a pure fuzzy system is a fuzzy set $A \circ (R(1), \dots, R(1)^{(k)})$ in V , a combination of the respective fuzzy set. Hence a pure fuzzy system constitutes the essential part of fuzzy systems as a general framework in which linguistic information is quantified and fuzzy principles are used to realize systematic use of linguistic information.

2.2 Takagi and Sugeno Fuzzy System

Instead considering fuzzy IF-THEN rules in form of equation (1), Takagi and Sugeno in 1985 proposed using fuzzy IF-THEN rules in the form:

$$L^{(k)} : \text{IF } x_1 \text{ is } F_1^{(k)} \text{ AND } \dots \\ \text{AND } x_n \text{ is } F_n^{(k)} \text{ THEN } y^k = c_0^k + c_1^k x_1 + \dots + c_n^k x_n \quad (3)$$

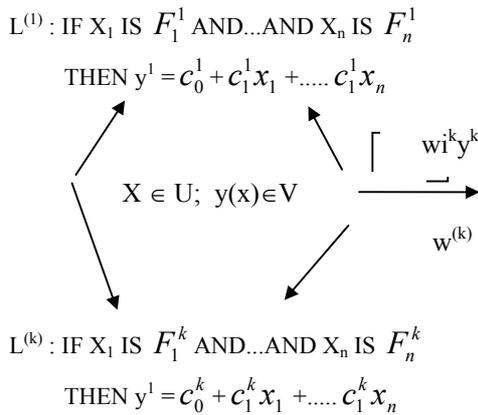
where $F_i^{(k)}$ are fuzzy sets, c_i are real-valued parameters, $y^{(k)}$ is the Takagi-Sugeno fuzzy system output due the rule $L^{(k)}$, and $k=1,2,\dots,m$. That is, they considered rules whose IF part is fuzzy but whose THEN part is crisp. For a real-valued input vector $x=(x_1,\dots, x_n)^T$, the output $y(x)$ of Takagi and Sugeno fuzzy systems is a weighted average of $y^{(k)}$:

$$y(x) = \frac{\sum_{k=1}^m w^{(k)} y^{(k)}}{\sum_{k=1}^m w^{(k)}} \quad (4)$$

where weight $w^{(k)}$ implies the overall truth value of the premise of rule $L^{(k)}$ for the input and is calculated as

$$w^{(k)} = \prod_{i=1}^n \mu_{F_i^{(k)}}(x_i) \quad (5)$$

which is shown in the following representation.



2.3 Fuzzy Systems with Fuzzification and Defuzzification

Compared with the pure fuzzy system we may add a fuzzifier to the input and a defuzzifier to the out-

put of the pure fuzzy system. The fuzzifier maps crisp points in U to fuzzy sets in U , and the defuzzifier maps fuzzy sets in V to crisp points in V . The fuzzy rule base as well as the fuzzy inference engine are the same as those in pure fuzzy logic systems, as shown in Figure 1.

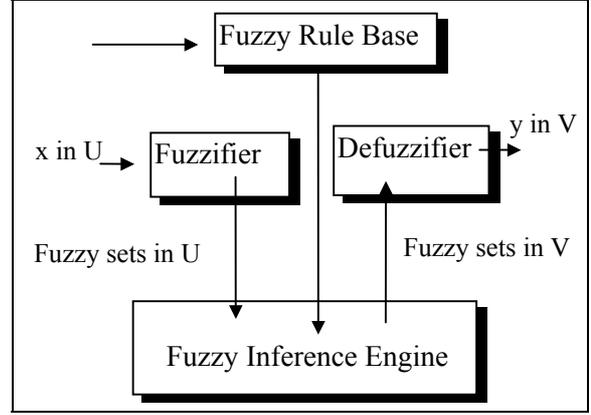


Fig. 1: Fuzzy system with fuzzifier and defuzzifier

2.4 Neuronal Nets

A neural net consists of input variables and weighting factors, activation layers and output variables. The physiological pendant of the inputs are the dendrites as part of the anterior motoneurons extend for one-half to one millimeter in all directions from the neuronal soma. Therefore, these dendrites can receive signals from a fairly spatial area around the motoneuron. This provides vast opportunity for summation of signals from many separate pre-synaptic neurons. The weighting functions physiological pendant are the synapses. The synapse could be interpreted as the juncture between one neuron and the next, based on three major parts, the soma, which is the main body of the neuron; a single axon, which extends from the soma into the peripheral nerve; and the dendrites, which are thin projections of the soma that extend up to one millimeter, into the surrounding areas of the cord. The output has its physiological pendant in the axon, which is the central core of a nerve fiber. The biological neuron and the artificial neuron are shown in Figure 2.

From Figure 2.b one can assume that the synapses of an artificial neuron receive an activation x_i with a specific strength w_i from another artificial neuron, which will be part of the summing process of the output, the axon. The basic concept results in an input vector $x = (x_1, \dots, x_n)^T$, a weighting vector $w = (w_1, \dots, w_n)^T$ and the resulting activity as sum of the weighted input, which could be assigned as activity function z :

$$z(\mathbf{w}, \mathbf{x}) = \sum_j \mathbf{w}_j \mathbf{x}_j = \mathbf{w}^T \mathbf{x}$$

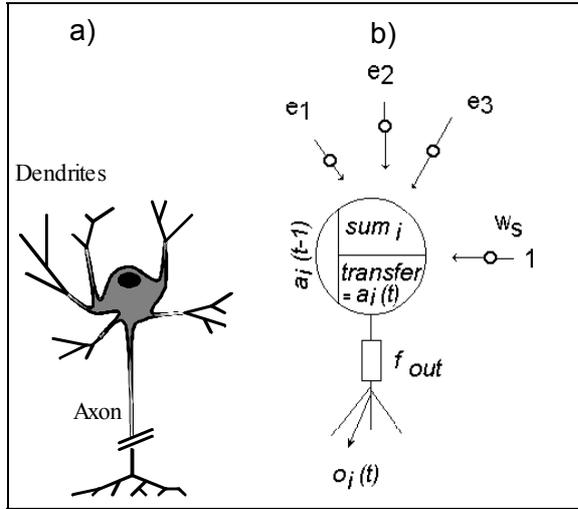


Fig.2: Biological neuron (a) and artificial neuron (b); for more details see text

Often there exists a threshold, which has to be passed, to activate the output. Modeling the threshold results in the relation

$$z(\mathbf{w}, \mathbf{x}) = \mathbf{w}^T \mathbf{x} - T$$

where T indicates the threshold. Assuming

$$\mathbf{x} \rightarrow \mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n, 1)^T$$

and

$$\mathbf{w} \rightarrow \mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_n, -T)^T$$

we receive the scalar product

$$z(\mathbf{w}, \mathbf{x}) = \sum_j \mathbf{w}_j \mathbf{x}_j - T =$$

$$(\mathbf{w}_1, \dots, \mathbf{w}_n, -T) (\mathbf{x}_1, \dots, \mathbf{x}_n, 1)^T = \mathbf{w}^T \mathbf{x}$$

which can be rearranged as follows

$$z = \mathbf{w}^{(0)} + \sum_i \mathbf{w}_i^{(1)} \mathbf{x}_i$$

whereby the notation^(k) indicates the correlation's of the \mathbf{x} components.

Modeling high-order synapses then can be directly derived from the above equations as follows:

$$z = \mathbf{w}^{(0)} + \sum_i \mathbf{w}_i^{(1)} \mathbf{x}_i + \sum_{ijk} \mathbf{w}_{ijk}^{(2)} \mathbf{x}_j \mathbf{x}_k + \sum_{ijkl} \mathbf{w}_{ijkl}^{(3)} \mathbf{x}_j \mathbf{x}_k \mathbf{x}_l + \dots$$

This type of artificial neurons are the so called sigma-pi-units.

The output of an artificial neuron can be described by the function $S(\cdot)$ depending on the internal activity z ,

$$y = S(z)$$

The complete output of an artificial neuron hence can be stated as transfer function of type y

$$y = f(\mathbf{x}, \mathbf{w}, z, S).$$

3. Theory of sensitization of neural nets

The idea to condition a neural net first by well defined easy distinguishable data sets and then to deepen and to enlarge the stored information by sensitization can be learned from cognitive psychology in the context of the chunking problem. Chunking is more or less the adaptation of a new fact or a so far unknown situation with the help of knowledge facts or models. Only out of old facts or acting strategies one can develop new strategies for understanding of so far unknown. Transforming this idea to the handling with neural nets means, that first a net has to learn a basic concept. To prevent that the net used includes typical output ranges in its classification behavior due to the measured data for analysis, and therefore looks for measures with a high output, it is necessary to normalize the input data set by using an appropriate-processing method [3]. To handle changes in the global state of the case study it is favorable to use a pre-processing step that results in a gradient vector that is calculated by the difference between the data of a case study as requested when no pathological situation is present minus the present actual data.

Once the neural classifier can separate all trained states which represent a basic concept in a sufficient way the weaker evolutionary states of the different case study states should be trained. The definition of the basic concept can be called: "the worst cases of a case study behavior" can simply be defined as problem oriented data set representation of a satisfactory classification state.

By presenting the weaker states after the basic concept is settled it can be ensured that the net will be forced to change its classification structure slightly out of its former structure, without destroying the older structure. This means that the net will be sensitized. Especially when a back propagation network is used the learning rules force the net to sensitize its structure in such a way that only the case study specific state representing structure is modified, as the classification results have to be the same over the whole sensitization period. If weaker

and weaker states will be presented successively the classification structure will change accordingly, until the similarity of the different evolutionary state representations will be so little, that the net can not be forced to change it's structure anymore.

Figures 3.a and 3.b show the sensitized net working structure. Since the classification potential is changed locally, the net changes its classification behavior not in general by learning the evolutionary data sets, but shows the according adaptive behavior. This local change surely can lead to the unification of so far divided concepts, a fact which will open the door to a wide range of so far unknown or unnoticed intrinsic relations of the micro array data sets, representing the case study states.

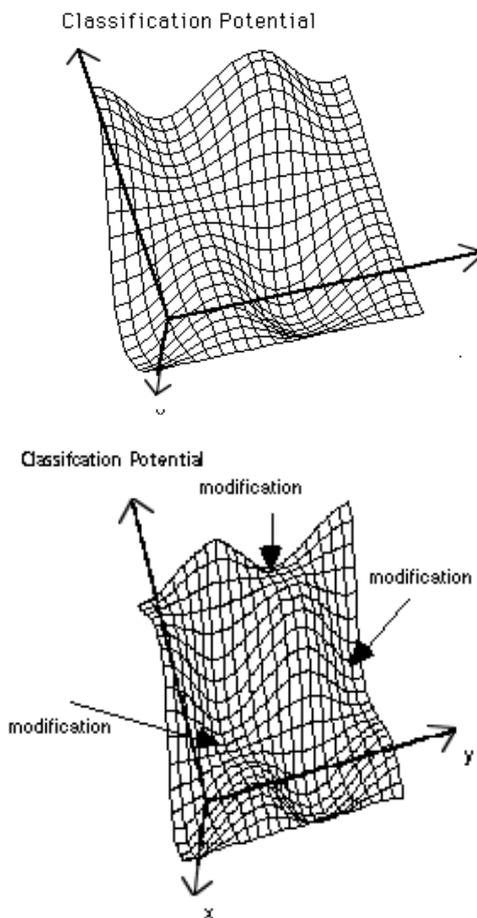


Fig. 3: Classification-potential a) E before sensitization and b) E' after sensitization

Henceforth, sensitization in case of micro array data analysis can be introduced as intelligent pre-processing for clustering analysis that means normalization and filtering, that is necessary due to

- Systematic experimental errors,
- Uneven hybridization gel,
- Background variations,

- Wavelength dependency,
- Intensity dependency.
- Image processing algorithm-dependency
- Etc.

Hence, the importance of using intelligent pre-processing algorithms is really based on the hypothesis underlying micro array analysis that the measured intensities for each arrayed gene represent its relative expression level. However, before the levels can be appropriately compared, one generally performs a number of transformations on the data to eliminate questionable or low quality data, to adjust measured intensities to facilitate comparisons, and to select those genes that are significantly differentially expressed, which explains the need for a pre-processing methodology beyond.

The sensitized neuronal nets classifier in general is able to separate all trained states representing a powerful concept of weaker evolutionary states of different case study situations, to be trained, as shown in Figures 4, 5 and 6. As shown in Figure 4 the actual situation for a patient may change after 3 sec from a normal state under anaesthesia to a critical state, an actual auricular fibrillation (tachycardia ventricular). The left column represents the possibilities in their special coded colours, indicated as stripes with different gay shadings.

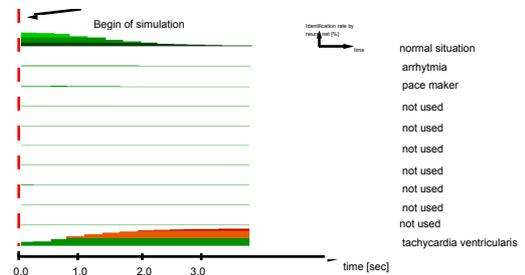


Fig. 4: Heart beat state change detected by a sensitized neural net

Figure 5 shows the time differences between an early warning of a common neural classifier and a sensitized neural network. Both nets have had the same warning criteria, setting an alarm when the probability for an pathological state is higher then 85%. It can be seen that a sensitized neural classifier is able to decrease the alarm time by a factor 5, as the net classifies the evolutionary state of the begin of an auricular fibrillation rather early.

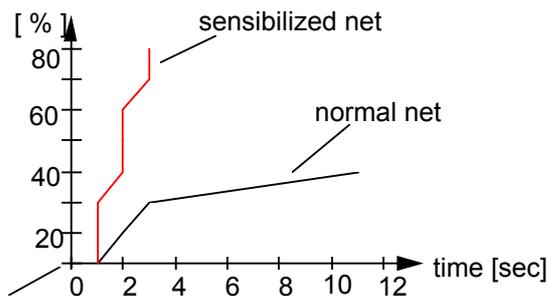


Fig. 5: Difference between early warning of a sensitized and a normal neural net

Another classification example is shown in Figure 6. The first line represents the pathological state of the patient, followed by the different pathological and orthological states listed below.

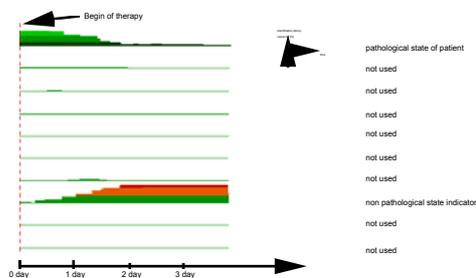


Fig. 6: GUI used to supervise patients states

As clearly can be seen over the supervised time period the patient state changed from the first to the eight state, whereby the classifier is able to detect fast transients ($< 0,5$ sec), slow changes over hours, days, as well as very slow transients over weeks, months, etc.

Due to the fact, that anaesthesiologists have a great interest in monitoring the depth of anaesthesia, not only in critical cases but also during routine anaesthesia, expanding soft computing based algorithms for anaesthesia monitoring is the state of the art for the challenge of more accuracy, more safety and more quality in anaesthesia, the vision of an physiologically gained monitoring.

In practical terms this means adequate information with respect to the depth of anaesthesia, which means very clearly show of depth changes that anaesthetic agents cause in the EEG, sensitivity with respect to artefacts, particularly induced by today's modern electro surgery equipment, sensitivity on the frequency below which the bulk of EEG activity occurs, called the spectral edge frequency, which is related to the end tidal concentrations of various anaesthetic agents, including halothane and

enflurane, as well as to blood levels of thiopental and fentanyl etc.

In case of micro array data analysis sensitized nets can be developed for usage in

- Total Intensity normalization,
- Iterative linear regression normalization,
- Standard deviation regularization,
- Cross Slide Replicates T-test,
- Signal/Noise checking,
- Slice analysis,
- Etc.

As an measure example one can assume that two factors contributed to the gene X, the biological factors one is interested in, and experimental factors, one is not interested in. This requests for a possibility to extract the experimental factors which can be done initializing the pre-processing task of the statistical analysis. The statistical analysis behind can be

- Pre-processed local linear regression model,
- Pre-processed least squares,
- Etc.

4. Conclusions

The potential of soft computing for micro array data analysis is huge. We only scratched the surface of the complex due to a brief view insight possible medical application domains. The potentially of soft computing and pre-processing contains an incredible number of solutions to the several problem depending domains.

5. References

- A. Kaufmann and M. M. Gupta, Introduction to Fuzzy Arithmetic, Van Nostrand Reinhold, 2001
- B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall International, 2002
- D. P. F. Möller, Soft Computing Methods, In: Mathematical and Computational Modeling and Simulation, Springer Verlag, Heidelberg, 2003
- H. Takagi and M. Sugeno, Fuzzy Identification of Systems and its Applications to Modelling and Control, IEEE Trans. on Man, and Cybern., SMC-15(1), 1985, pp. 116-132
- L. X. Wang, Adaptive Fuzzy Systems and Control, Prentice Hall International, 1994
- L. A. Zadeh, Fuzzy Sets, Informat. Control, Vol. 8, 1965, pp. 338-353

LATE PAPERS

Image retrieval at low bit rates: BSP Trees vs. JPEG

Michal Stich^{1,2} and Gerald Schaefer¹

¹School of Computing and Technology
The Nottingham Trent University, Nottingham, U.K.

²Dept. of Computing, Electronics and Automated Control
Silesian University of Technology, Gliwice, Poland

KEYWORDS

Image retrieval, image compression, BSP tree, JPEG.

ABSTRACT

Content based image retrieval (CBIR) has still a long way to go before it will become capable of distinguishing similar rather than almost identical images. These limitations result both from a lack of ‘intelligent’ enough algorithms as well as problems in terms of computational restrictions which force algorithms to be as simple as possible, while maintaining reasonably high effectiveness. Images are not analysed in terms of their physical or logical contents in the way humans perceive it but most often as statistical information such as spatial correlation of colors or intensities of separate pixels. This information is usually extracted at a very local (pixel) level and may be miss-interpreted if pixel values change (even though the overall image will still appear similar), which is especially the case in image compression. Results of earlier experiments [10] show that this can cause notable problems for CBIR algorithms. In this paper, we try to address the problem by investigating a more appropriate compression algorithm based on binary space partitioning trees and how it can improve the retrieval performance of compressed images. Simulations on two different image databases show that BSP compression outperforms JPEG for colour-based image retrieval while for texture-based indexing JPEG performs better.

INTRODUCTION

In our previous work [10] we have studied the influence of standard JPEG compression on the effectiveness of several image retrieval approaches: color histograms, QBIC histograms, colour moments, colour correlograms, spatial-chromatic histograms and color coherence vectors. Results of these experiments showed that while slight compression has little effect on colour based CBIR it plays a significant role when higher compression levels are applied resulting in a notably reduced image retrieval performance.

In this paper we try to present one possible solution to this problem by evaluating an alternative compression approach – binary space partitioning (BSP) trees [9,8].

This compression method we compared to the most commonly used image coding algorithm: JPEG [14]. Our experimental imagery set is UCID, an Uncompressed Color Image Database [11] which is the same dataset used in [10]. In the experiments we compress these images to very low bitrates using JPEG and to a level giving slightly lower bit rate using BSP. Our results show that in contrast to JPEG image compression based on binary space partitioning can be used in color image retrieval for low bit-rates almost without any loss in performance. On the other hand, texture retrieval results obtained from the Brodatz texture set shows a significant drop in retrieval performance.

The rest of the paper is organised as follows: the next section describes the BSP compression algorithm used in the experiments. Then the colour and texture retrieval algorithms evaluated are briefly explained. The following section presents our experimental results while finally concluding remarks are presented.

BSP TREE COMPRESSION

Binary Space Partitioning (BSP) is a relatively new approach to image compression. It was originally used to represent a three dimensional space for convenience of hidden surface removal algorithms and was recently successfully applied to 2D imagery as well [9,8].

The idea is to divide an image plane by one of several pre-defined straight lines. This process is recursively repeated for each of the two sub-regions created by the previous partition. At some point partitioning is stopped and basic information about the fraction of the entire space is stored. The coded information is organised in a tree structure where each node indicates a part of space and contains information about the partitioning line, while its two children point to two sub-spaces created as a result of the partitioning of their parent. The bottom-most nodes of the tree contain information about that region (e.g. color of this region of the image).

The first attempt to compress images using BSP was made by Radha *et al.* [9] whose algorithm utilised the idea of moment preserving thresholding and was only applied to gray-scale images. Qiu and Sudirman extended this idea to colour images [8]. In our implementation, rather than using moment preserving thresholding we calculate the average color of the

partitioned regions and then the resulting errors as the sum of differences between intensities for each pixel in the region and the average intensity of region to which this pixel belongs. The partitioning line is then chosen so as to minimize the error. While this approach is clearly computationally more expensive it has the advantage of ‘optimality’ and hence better image quality. We also employ the CIEL*a*b* color space [2] which gives improved quality in comparison to RGB.

CBIR ALGORITHMS

In this section we provide a brief description of the colour and texture CBIR algorithms that were used for the experiments.

Colour Histograms – Histogram Intersection

Given a bounded, discrete signal one can build a histogram simply by counting the number of occurrences of each signal value. Swain and Ballard [13] were the first to use colour histograms to describe images in order to perform object recognition and image retrieval. Indeed, it was Swain and Ballard’s work that laid the foundations for the field of CBIR as we know it today. As distance measure they introduced (the complement of) histogram intersection defined as

$$d_{\text{HIS}}(I_1, I_2) = 1 - \sum_{k=1}^N \min(H_1(k), H_2(k)) \quad (1)$$

where H_1 and H_2 are the colour histograms of images I_1 and I_2 , and N is the number of bins used for representing the histogram. It can be shown [13] that histogram intersection is equivalent to the L_1 norm and hence a metric. We used 8 x 8 x 8 RGB histograms in our experiments.

Colour Histograms - QBIC

An alternative to the L_1 norm is to use the Euclidean distance (L_2 norm) between two histograms. This approach was taken in the QBIC system [4] where they also addressed the problem of possible false negatives due to slight colour shifts by taking into account the similarity between separate histogram bins. This can be expressed in a quadratic form distance measure as

$$d_{\text{QBIC}}(I_1, I_2) = (H_1 - H_2)A(H_1 - H_2)^T \quad (2)$$

where H_1 and H_2 are again the two colour histograms (in the form of a vector) and A is an $N \times N$ matrix containing the inter-bin distances. We used the Munsell colour space divided into 256 bins (16 for hue, 4 for chroma and value respectively) to generate these histograms.

Colour Moments

Stricker and Orengo [12] used colour moments as a compact colour descriptor for CBIR. The n^{th} central

(normalised) moment of a colour distribution is defined as

$$M^n(I) = \sqrt[n]{\frac{1}{N} \sum (M^1(I) - c(x, y))^n} \quad (3)$$

with

$$M^1(I) = \frac{1}{N} \sum c(x, y) \quad (4)$$

where N is the number of pixels in an image and $c(x, y)$ describes the colour of the pixel at location (x, y) . For our experiments we used the first three moments in the HSV colour space. The distance between two images is defined as the sum of absolute distances between their moments (L_1 norm)

$$d_{\text{MNT}}(I_1, I_2) = \sum_{i=1}^n |M^i(I_1) - M^i(I_2)| \quad (5)$$

Color Coherence Vectors.

Pass and Zabih [7] introduced colour coherence vectors as a method of integrating spatial information into the retrieval process. Colour coherence vectors consist of two histograms: one histogram of coherent and one of non-coherent pixels. Pixels are considered to be coherent if they are part of a continuous uniformly coloured area and the size of this area exceeds some threshold τ where τ is usually defined as 1% of the overall area of an image. The L_1 norm is used as the distance metric between two colour coherence vectors

$$d_{\text{CCV}}(I_1, I_2) = \sum_{k=1}^N \left[\left| H_1^c(k) - H_2^c(k) \right| + \left| H_1^s(k) - H_2^s(k) \right| \right] \quad (6)$$

where H_i^c and H_i^s are the histograms of coherent and non-coherent (scattered) pixels respectively. In our implementation we first blurred the image using a 3 x 3 averaging filter and used 8 x 8 x 8 RGB bins for representing the histograms.

Colour Correlograms

Another approach to incorporate information on the spatial correlation between the colours present in an image was proposed by Huang *et al.* [5]. They introduced the notation of colour correlograms (CCRs) defined as

$$\gamma_{c_i c_j}^{(k)}(I) \equiv \Pr_{p_1 \in I_{c_i}, p_2 \in I} \left[p_2 \in I_{c_j}, |p_1 - p_2| = k \right] \quad (7)$$

with

$$|p_1 - p_2| = \max\{|x_1 - x_2|, |y_1 - y_2|\} \quad (8)$$

where c_i and c_j denote two colours and (x_k, y_k) denote pixel locations. In other words, given any colour c_i in

the image, γ gives the probability that a pixel at distance k away is of colour c_j .

As full colour correlograms are expensive both in terms of computation and storage requirements, usually a simpler form called auto-correlogram (ACR) defined as

$$\alpha_c^{(k)} \equiv \gamma_{c,c}^{(k)}(I) \quad (9)$$

is being used, i.e. only the spatial correlation of each colour to itself is recorded. Two CCRs are compared using

$$d_{CCR}(I_1, I_2) = \frac{\sum_{i,j \in [m], k \in [d]} \left| \gamma_{c_i, c_j}^{(k)}(I) - \gamma_{c_i, c_j}^{(k)}(I') \right|}{\sum_{i,j \in [m], k \in [d]} \left(1 + \gamma_{c_i, c_j}^{(k)}(I) + \gamma_{c_i, c_j}^{(k)}(I') \right)} \quad (10)$$

We used ACRs with 8 x 8 x 8 RGB colours, for k we chose $\{1, 3, 5, 7\}$.

Spatial-chromatic Histograms

Cinque *et al.* [3] introduced spatial-chromatic histograms (SCHs) as an alternative method for representing both colour and spatial information. SCHs consist of a colour histogram

$$h(k) = \frac{|A_k|}{n * m} \quad (11)$$

where A_k is a set having the same colour k , and n and m are the dimensions of the image; and location information on each colour characterised through its baricentre

$$b(k) \equiv \left(\frac{1}{n} \frac{1}{|A_k|} \sum_{(x,y) \in A_k} x; \frac{1}{m} \frac{1}{|A_k|} \sum_{(x,y) \in A_k} y \right) \quad (12)$$

and the standard deviation of distances of a given colour from its baricentre

$$\sigma(k) = \sqrt{\frac{1}{|A_k|} \sum_{p \in A_k} d(p, b(k))^2} \quad (13)$$

The SCH is then given as

$$H_{SCH}(k) = [h(k), b(k), \sigma(k)] \quad (14)$$

and similarity between two SCHs calculated as

$$d_{SCH}(I_1, I_2) = 2 - \sum_{i=1}^N \min \left(h_{I_1}(i), h_{I_2}(i) \right) \left(\frac{\sqrt{2} - d(b_{I_1}(i), b_{I_2}(i))}{\sqrt{2}} + \frac{\min(\sigma_{I_1}(i), \sigma_{I_2}(i))}{\max(\sigma_{I_1}(i), \sigma_{I_2}(i))} \right) \quad (15)$$

In our implementation we divided the Munsell colour space uniformly into 512 areas whose centres were used as the colours to describe the SCH.

3.7. Local Binary Patterns

Local Binary Patterns (LBP) [6] is a very simple method for texture retrieval. For each pixel in the image and its 8 neighbours the following operation is performed:

$$LBP = \sum_{i=1}^8 T_i * W_i \quad (16)$$

with

$$\begin{aligned} T &= 1 \text{ if } p_i \geq P \\ T &= 0 \text{ if } p_i < P \end{aligned} \quad (17)$$

where P is the intensity of the centre pixel, p_i the intensity of its i^{th} neighbor and W_i is the weight (a power of 2) associated with p_i . The idea is to create histograms of 256 possible values (since each pixel has 8 neighbours). Two LBP histograms H_1^{LBP} and H_2^{LBP} can be then compared by the histogram intersection

$$d_{LBP}(I_1, I_2) = 1 - \sum_{k=1}^{256} \min(H_1^{LBP}(k), H_2^{LBP}(k)) \quad (18)$$

Rotation invariant LBP

A rotation invariant version of the LBP algorithm can be deduced by finding all groups of patterns that can be obtained through rotation of the 8 neighbouring pixels [6] by

$$riLBP(k) = \sum_{i=1}^8 T_i * W_i^{(k)} \quad (19)$$

and then choosing the smallest LBP value

$$riLBP' = \min(riLBP(k)), k = 1, 2, \dots, 8 \quad (20)$$

which results in 36 possible descriptors that are again summarized in histograms. Two rotation invariant LBP histograms H_1^{riLBP} and H_2^{riLBP} are compared by

$$d_{riLBP}(I_1, I_2) = 1 - \sum_{k=1}^{36} \min(H_1^{riLBP}(k), H_2^{riLBP}(k)) \quad (21)$$

EXPERIMENTAL RESULTS

For our experiments we adopted the UCID (Uncompressed Colour Image Database) [11] for evaluating the colour descriptors while we used Brodatz textures [1] for performing texture based image retrieval. The UCID set consists of 1338 colour images all preserved in their uncompressed state. It also provides a ground truth of 262 assigned query images each together with a set of corresponding model images that an ideal image retrieval system would return. The

112 Brodatz textures we divided into 9 non-overlapping parts each and assigned the centre block the query and the rest the model images that should be retrieved.

As performance measure we use the retrieval effectiveness from [4] which is given by

$$RE_Q = \frac{\sum_{i=1}^{S_Q} R_i}{\sum_{i=1}^{S_Q} I_i} \quad (21)$$

where R_i is the rank of the i^{th} matching image and I_i is the ideal rank of the i^{th} match (i.e. $I = \{1, 2, \dots, S_Q\}$). The average retrieval effectiveness ARE is then taken as the mean of RE over all query images.

For the experiments both query and model images were heavily compressed using the JPEG algorithm (q-factor of 5) and to a level giving similar (although even slightly lower) bit rate using BSP compression. Initial observations lead to the conclusion, that low bit rate BSP compression retains global colour information of the image, which is not the case for JPEG – heavily compressed JPEG images tend to almost entirely lose their colour content while the contours are still visible (though heavily affected by blocking effects). Conversely, BSP images keep their colours, while shapes are preserved as clearly.

Table 1: Image retrieval results from the UCID dataset.

CBIR algorithm	JPEG		BSP		BSP	
	Q orig M orig	Q cmp M orig	Q orig M cmp	Q cmp M orig	Q orig M cmp	Q orig M cmp
Colour hist.	4.31	25.73	13.48	4.71	4.63	
QBIC	5.64	23.86	19.56	6.75	7.47	
Colour moments	7.33	27.59	33.93	7.86	11.64	
Colour coh. vect.	4.69	22.00	13.97	4.65	5.35	
Colour correlogr.	5.32	11.80	8.91	5.69	5.97	
Spat.-chrom. hist.	3.69	23.88	12.49	4.25	3.80	



Figure 1: Example UCID query together with top 9 retrieved (using spatial-chromatic histograms) images both based on JPEG (left) and BSP (right) encoded images.

We performed image retrieval on the datasets mentioned above, first by using only uncompressed images in order to get a ‘best possible’ retrieval performance which we can use as a benchmark. We

then retrieved the images once with the model images compressed and once with the query images compressed to the levels indicated above, both for JPEG and for BSP compression. The results from the UCID dataset are listed in Table 1. It is obvious that our previous expectations are confirmed if not surpassed: while image retrieval performance at low bitrates drops significantly for JPEG compression, image retrieval based on BSP is almost independent of compression. Retrieval performance is still very good even at these high compression ratios of well over 1:100! An example of this performance difference can be seen in Figure 1 where a UCID query image is shown together with the top nine retrieved images, both for JPEG and BSP encoded images. While the BSP based retrieval retrieves all three model images in top ranks, none of the JPEG models are returned. Also, we can see that the images retrieved in ranks 4-9 are much more useful (three taken at the same location) than those returned by JPEG compressed images.

Table 1. Image retrieval results from the Brodatz dataset.

CBIR algorithm	JPEG		BSP		BSP	
	Q orig M orig	Q cmp M orig	Q orig M cmp	Q cmp M orig	Q orig M cmp	Q orig M cmp
LBP	3.60	64.40	33.24	80.12	71.03	
Rot.inv. LBP	5.05	81.69	57.94	95.45	92.29	

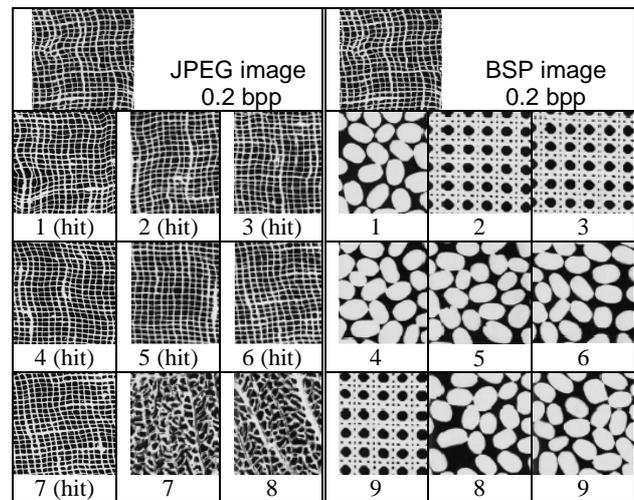


Figure 2: Example UCID query together with top 9 retrieved (using LBP) images both based on JPEG (left) and BSP (right) encoded images.

Results for texture retrieval based on the Brodatz set show a different picture. Since BSP essentially divides an image in areas of uniform colour pixel-based statistics such as LBP and riLBP get heavily distorted. The image retrieval results are given in Table 2 from where we can see that while JPEG based image retrieval suffers from a significant performance drop this drop is even greater so for BSP images. Again we provide a visual example of this which is given in Figure 2.

CONCLUSIONS

Through an extensive set of simulations on two different image databases we have demonstrated that different compression algorithms will affect the outcome of image retrieval differently at high compression. While JPEG suffers from a significant performance drop for colour based CBIR, retrieval of BSP compressed images is almost independent of the compression rate. Therefore we suggest BSP as a much more appropriate compression method that can be used for image retrieval even at very low bitrates. On the other hand, texture based retrieval results are much less favourable, here BSP performs even worse than JPEG. Finally we want to emphasise that in this paper we made use only of the pixel based representations of the compressed images (hence, essentially decompressing them again). A more promising route is to couple the retrieval algorithm with the compression hence achieving compressed domain image retrieval. This alternative is not only attractive in terms of computational complexity but seems also to provide very good retrieval results as some initial work in [8] shows.

REFERENCES

- [1] P. Brodatz. "Textures", Dover, New York, 1966.
- [2] CIE "Colorimetry", 2nd edition, *CIE Publ. No. 15.2*, Commission International de L'Eclairage, Vienna, 1986.
- [3] L. Cinque, S. Levialdi, A. Pellicano. "Color-Based Image Retrieval using Spatial-Chromatic Histograms." In *Proc. IEEE Multimedia Systems 99*, pp. 969-973, 1999.
- [4] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, R. Barber. "Efficient and Effective Querying by Image Content." *Journal of Intelligent Information Systems*, Vol.3, No.3/4, pp.231-262, 1994.
- [5] J. Huang, S. R. Kumar, M. Mitra, Wei-Jing Zhu, R. Zabih. "Image Indexing Using Color Correlograms." In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 762-768, 1997.
- [6] T. Ojala, M. Pietikäinen, T. Mäenpää, "Gray Scale Rotation Invariant Texture Classification with Local Binary Patterns." *Proc. European Conference on Computer Vision*, 2000.
- [7] G. Pass, R. Zabih. "Histogram Refinement for Content-Based Image Retrieval." In *IEEE Workshop on Applications of Computer Vision*, pp. 96-102, 1996.
- [8] G. Qiu, S. Sudirman "Color Image Coding, Indexing and Retrieval using Binary Space Partitioning Tree." In *Proceeding to 8th Color Image Conference*, pages 195-201, 2000.
- [9] H. Radha, et al "Image compression using binary space partitioning tree." *IEEE Trans. On Image Processing*, vol.5, pages 1610-1624, 1996.
- [10] G. Schaefer, M. Stich "On the influence of image compression on the performance of content based image retrieval." In *6th International Conference on VISual Information Systems*, 2003.
- [11] G. Schaefer, M. Stich. "UCID – An Uncompressed Colour Image Database." In *Storage and Retrieval Methods and Applications for Multimedia 2004*, 2004.
- [12] M. A. Stricker, M. Orengo. "Similarity of Color Images." In *Storage and Retrieval for Image and Video Databases III*, pp. 381-392. 1995.
- [13] M. Swain, D. Ballard. "Color Indexing." *International Journal of Computer Vision*, 7:1, pp. 11-32. 1991.
- [14] G.K. Wallace. "The JPEG Still Picture Compression Standard." *Communications of the ACM*, Vol. 34, No. 4, pp. 30-44, 1991.