

# SIMULATION OF RECONFIGURABLE SYSTEMS: FROM CONTROL CODE SIMULATION TO REFLECTIVE SIMULATION

Pascal Berruet  
LESTER CNRS FRE 2734  
Université de Bretagne Sud BP 92116  
56321 Lorient cedex, France  
E-mail: Pascal.Berruet@univ-ubs.fr

## KEYWORDS

Component, reconfiguration, nested simulation, control simulation, reconfigurable manufacturing systems.

## ABSTRACT

Reconfigurable systems, that offer flexibility and robustness to efficiently manage quality of service in spite of uncertainties and disturbances, are a necessary evolution of classical controlled systems. As they can change their organization during their use, so their control laws, they become more complex. Two simulations types are presented to analyze such systems. The first one enables to check the behavior of the system controlled with the code that will be implemented. It is performed using a simulator based on virtual reality and physical engine. The second one takes parts in the reconfiguration procedure evaluation. It consists in the simulation of elements that simulate their own environment using their own models. A reconfigurable system using look-ahead simulation for its reconfiguration can then be simulated. The developments have been performed in collaboration with firms and Professor E. Kindler.

## INTRODUCTION

Reconfigurable systems have gained more and more interest last decade. Before, systems were traditionally designed to keep the same structure and behavior all along their lifespan. In case of failures occurrence or changes in objectives, they may need external intervention to continue their mission.

Reconfigurable systems offer the opportunity to choose the organization of their elements very late in the conception and to modify it dynamically or not during exploitation.

Such systems have been studied in several domains as: electronics (Auguin, et al. 2003), communications (Mitchell, et al. 1998), control (Wills, et al., 2001; Cotting and Burken, 2001), manufacturing (Combacau, et al. 2000) or robotics (Kotay and Rus 1999; Kamimura et al. 2001). Even if they target different applications, these systems share common concepts. This paper focuses more on reconfigurable manufacturing systems (RMS).

Figure 1 depicts a part of an RMS consisting of a great cycle called main ring and providing five working areas (W1 to W5). Different machining functions can be affected to  $W_i$ . This example gives an idea of the numerous possibilities to configure the system and to react in case of a failure occurrence.

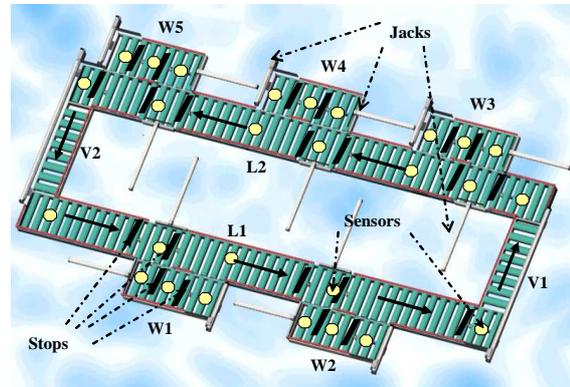


Figure 1: a reconfigurable conveyor

As the field is extended, this leads to different simulation problematic: As one control can be assigned to each configuration, RMS includes several control versions. How can we ensure that each control is adequately designed? The need for control code simulation before implementation is increased. On the other hand, simulation can be used during exploitation to help in the choice of the future organization. As simulation is classically used to evaluate a system during its design, how is it possible to simulate a reconfigurable system that chooses its configuration using on-line simulation techniques?

Before answering these two questions, general concepts regarding RMS and simulation typologies are presented.

## RECONFIGURABLE MANUFACTURING SYSTEM DESCRIPTION

An RMS is composed of several resources which can either execute machining, transport or stocking operations. The transport resources link together stationary (machining or stocking) resources. The model has to describe the relations between its components and the products processed at one level of granularity.

The key point for the RMS description is to separate the architecture of the system from its configuration (cf. Figure 2). Architecture consists in the description of resources and products that can be processed on it. The configuration describes how elements of the architecture are used to achieve a goal.

Architecture is separated into logical and physical parts. Logical part describes machining functions performed on the product and their association to form function sequences (named logical operating sequences) to obtain a finished product. Physical architecture describes the elements in the system and the links between them. Connections represent the potential transfer links between stationary resources; these connections are associated with the transport resources that can perform them. Potential operations complete the description of the architecture and links physical and logical parts by bridging functions or products and resources.

Configuration is split into logical and physical configuration. The logical aspect is constituted of function instances and uses logical operating sequences from the logical architecture. A function instance is the realization of a function on a product. Physical configuration is constituted of the components taken from the logical architecture and transfer sequences. Transfer sequences are used to describe the transfer from one resource to another. As with architecture, the correspondence between physical and logical configuration is done through operations. Operations also links the architecture with the configuration by using some reserved operations defined in the architecture.

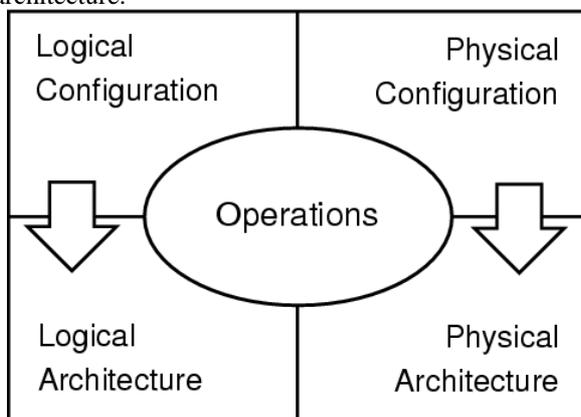


Figure 2: Organization of the model

Operations play a special role in the RMS approach. As a consequence the determination of a configuration can be performed from the knowledge of active operations. In the following, the choice of a new configuration remains to obtaining a new set of operations.

## SIMULATION TYPOLOGIES

Simulation is largely used in industry. The major interest is that it is quite easy to practice compared with proof techniques. This is particularly true for analyzing complex systems that can not be easily modeled using

mathematical or analytical models. The drawback is that the system is only validated according with a set of scenarios.

Simulation is classically classified into in three categories: static simulation, continuous simulation and discrete event simulation. Static or Monte Carlo simulation enables to solve stochastic problems without needing explicit time representation (Page 1994). The two last kinds of simulations concern more dynamic systems. In continuous simulation, state variables evolve during the time without any interruption as in discrete event simulation they change according to event occurrences (Ray and Claramunt 2003).

In manufacturing systems, simulation software tools can be classified into two kinds: discrete event simulation and geometric simulation (Klingstam and Gullander 1999).

Discrete event simulation (also called flow simulation) is suitable for analyzing system and its performances. This kind of simulation often expresses flows and has the advantage of rapidly providing results based on large simulated periods. But as the control behavior is embedded in the model, it is not relevant for control code testing. Examples of flow oriented simulators are Arena, Extend, Cadence and Quest. Others are based on simulation languages such as Simula (SIMULA 1986).

At the opposite geometric simulation simulates the geometry of a part, or the whole manufacturing system. Geometric simulation often refers to continuous simulation. Generally this kind of simulation allows testing control code of the system. Two dimensions and three dimensions are the two techniques to display the system evolution. Two dimensions display simulation is not precise enough. An example of two dimensions display simulator is ControlBuild. Three dimensions display simulation applications are as follows: virtual factory (Wenbin et al. 2002) and robotic (Ju and al. 1997). In robotic applications, simulation often concerns a small part of system.

Recently another simulation has been introduced to be used on line as a decision helping tool. The use of on-line simulation was introduced by G. R. Drake and J. S. Smith (Drake and Smith 1996). This concept was extended to look-ahead simulation in (Peters 1998) in order to test decisions. The works only consider a high level view of the system. Different works used simulation in decision procedures (Tomizuka 2002). Gupta focus on on-line scheduling (Gupta et al. 2002) and more recently, Cardin and Castagna (Cardin and Castagna 2006) extend this kind of simulation to solve the problem of the number of workstation setups. By the way they propose a method called proactive simulation to synchronize the simulation during on-line exploitation.

## CONTROL SIMULATION USING SIMSED

For reconfigurable systems, one objective is to test and validate the control codes before on-site implementation. As RMS include more control codes than classical manufacturing systems, this part is essential and has not to be time consuming in front of the final user. This goal is achieved using joint simulation of the system's material part and control part. The framework and applications developed are presented below.

### Approach

The global process is part of a traditional flow allowing from a simulation to validate or modify the parameters of the design. It also integrates a component-based approach to facilitate design process. Simulation concerns both operating and control parts; the control program being associated with the operating part.

The procedure described in figure 3 involves three steps: material part design, control part design and simulation. Operating part design and control part design is realized by using libraries. After validation, control program can be loaded in a PLC. If simulation does not fit the specifications, the control part and if needed the operating part are modified.

The following of the paper presents the three main steps emphasized in figure 3.

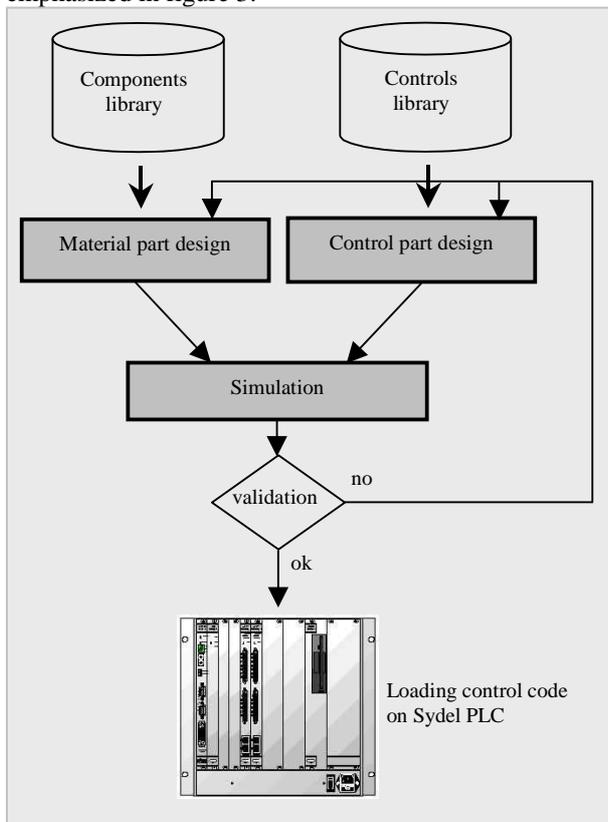


Figure 3: Simulation procedure

The system simulator used is a three dimensions display simulator. Operating part simulation is performed

simultaneously with control program simulation. Products and parcels are simulated as individual entities, which allows a precise simulation taking into account collision problems. This simulator is called SimSED (Lallican et al. 2005).

### Operating part design

The material part design is performed using SimSED DESIGNER tool.

A component-based model approach has been adopted to provide easy way to reuse previously modeled elements. The complete model of a system is seen as an assembling of components (Berruet et al. 2005). The component description uses black-box formalism. All components include parameters: static parameters such as position, orientation in 3D environment and dynamic parameters, for example the speed for a motor. Components are stored in a library. This software is seen as an ergonomic interface for 3D simulator. To design a system it is sufficient to select components from the library and to parameterize them according to system features.

### Control part design

The control part is written using a software compatible with the IEC 61131-3 standard. STRATON software is used to write the different controls and to download them to virtual machine for the simulation (Copalp, 2004).

The control part implementation is realized with STRATON Workbench and its simulation uses STRATON virtual machine. The interest is to simulate and test controls that will be really implemented, without any transcription. This tool has been chosen because the our partner the Sydel society has developed a PLC based on Vx Works operating system and a STRATON Virtual Machine.

### Analysis using joint simulation

The validation of the control part is performed using simulation of control code coupled with operating part one. The dedicated simulator is called SimSED SIMULATION.

This method uses continuous simulation respecting the PLC cycle. Synchronization management between the two softwares is dedicated to SimSED SIMULATION. It is also charged to control STRATON simulation execution.

Figure 4 described the simulation cycle. The cycle is divided into 2 parts. The goal of the first part, carried out by SimSED, is first to memorize outputs Straton values. Then components execute one simulation cycle. After Straton inputs are updated. The second part makes it possible to evolve the control program according to the new values of these inputs. When control program simulation cycle is performed, outputs Straton are updated.

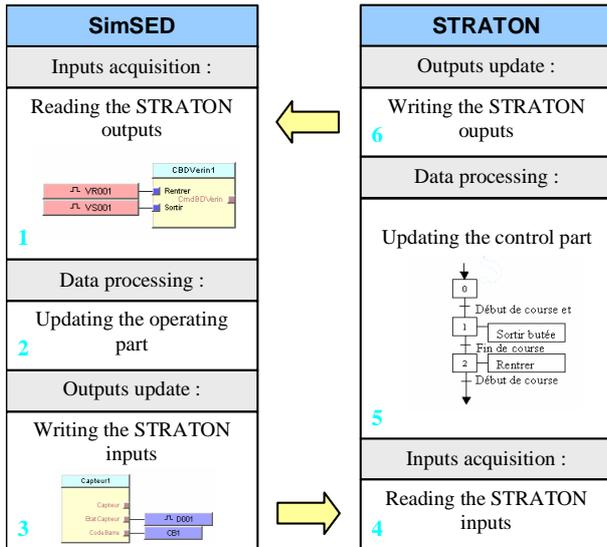


Figure 4: Simulation cycle description

### SimSED Features

To be relevant, a simulation has to be as close as possible to the real system. As the control is checked, it can be seen as a parameter of the other part of the system that is the operating part. To provide a realistic behavior of the material part, SimSED SIMULATION integrates a dynamic engine called OpenDynamic Engine (ODE 2004). This open source library enables to simulate rigid body dynamics. It has advanced joint types and integrated collision detection with friction. Problems like critical speed or acceleration, low sensor tolerance, parcels collision can be pointed out.

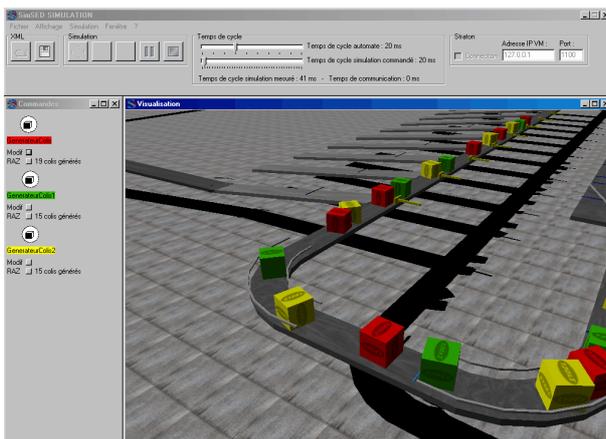


Figure 5 SimSED simulation interface

Animation is also an important feature. 3D animation helps to visualize clearly the behavior of the simulated workshop and emphasizes the understanding of the system. The proposed 3D animation enables the designer to zoom in on a specific part of the workshop to watch it in detail, or to zoom out for overall review. It is also possible to follow an object moving around the virtual system. Designer can also change viewpoints as

desired. All these features enable to detect critical points more easily.

### Interest for reconfigurable systems

As previously mentioned, reconfigurable systems have more than one control. As these controls may contain errors, each control program has to be checked before on-site implantation. The advantage of the proposed method is to notably reduce the test time spent in front of the client. It also enables to test more control versions during a fixed time period and to debug more quickly the controls.

Evolutions concern integration of failures occurrence in order to test a relevant reaction and to focus on control versions switches.

### REFLECTIVE SIMULATION OF RMS

Another simulation use tackles with the evaluation of the reconfiguration process itself.

### Reconfiguration process

Reconfiguration process requires first to localize the faulty part of the system, to analyze the impact on the rest of the system, to decide a new organization of the system and then to apply corrective actions to reach the proposed organization. The decisional step requires the knowledge of the potentialities of the system and the operating sequences. Previous works using graph models and graph theory enable to determine (Berruet et al., 2000):

- if there is a possibility for the manufacturing system to go on with the current production;
- if some resources have to be set in production mode;
- the path, a part can follow, to complete its logical operating sequence: the sets of possible controls.

But the presented procedures took very few dynamics parameters into account. To complete the procedure that gives several configurations, an evaluation step should be performed. The result should be to find the most appropriateness configuration according with the current situation.

### Reconfiguration using look-ahead simulation

For an RMS, simulation enables to anticipate the behavior of the system configuration based on different sequences of operations.

Let  $\Sigma$  be the RMS composed of a tolerant architecture (this means that its architecture is not only composed of critical elements the failure of one totally paralyse the system). Let  $C_{init}$  be the initial configuration. During the exploitation, a failure occurs at  $t_f$ . A new configuration  $C_{new}$  has to be determined in order to enable the system to go on. A first module determines the possibility to continue and whether the system has to change its

configuration. A second module determines a set of configurations  $SC = \{C_1, \dots, C_n\}$ . These configurations are evaluated in order to choose the appropriate one. Then the chosen configuration  $C_c$  can be applied.

The evaluation of  $C_i$  is not trivial because the system is quite complex, composed of several process that may evolve in parallel with different types of synchronisms. In the proposed approach, look-ahead simulation is used for performing the step that enables to choose  $C_c$  from  $SC$ . Each  $C_i \in SC$  is simulated and the best one is selected according with some criteria such as the completion time for the current job.

Therefore the real RMS constructed according to the variant  $C_c$  is to be an anticipatory system using simulation model. It has to be noticed that this kind of simulation is performed during the system existence. Let such a simulation be called *internal simulation*, the models used by it be called *internal models* and the anticipation be called internal anticipation (Kindler et al. 2004).

### Reflective simulation

The most common simulation is the simulation of a system during the design phase. Let such a simulation be called *external simulation*, the models used by it be called *external models*.

As any system, an RMS has to be evaluated during its design phase in order to evaluate the reconfiguration process. If simulation models are used for anticipating the system's behavior during the design, the point is then to have models that enable to reflect that the modeled systems are anticipatory ones, i.e. to have models that enable different levels of simulation. Indeed, one must simulate systems holding elements that simulate their own environment using their own models. In such cases, we can speak about nested simulation, expressing that the simulated systems themselves contain elements that handle simulation models. Moreover, we speak about reflective simulation, expressing that the simulating elements held by the simulated systems simulate parts of systems that holds them. Reflective simulation is thus a special case of nested simulation.

### Principle of reflective simulation

At the design phase of  $\Sigma$ , external simulation models are classically used. If the designers know that the system will use a control computer that will run simulation (i.e. that will handle with one or more internal simulation models), that computer (including the internal models) has to be reflected by the internal model. The computer itself does not need to be reflected in its many details, but the run of the internal models on it has to be reflected in details (Kindler 2000a, b). Then this internal model has to be reflected in the external one. Otherwise the external model would anticipate the behavior of  $\Sigma$  in a way different from the real one. Let the next analysis

be limited to the case that all computing processes used for the internal anticipation (including internal simulation models) run on only one computer  $\pi$  existing in  $\Sigma$ .

The external simulation model of  $\Sigma$  should reflect the components of  $\Sigma$  and also  $\pi$ . The components have to be reflected according to their mutual interactions. It holds for  $\pi$  in the same manner as for the other components (e.g. machines, transport tools, storage, material units, etc...). Therefore  $\pi$  has to be reflected in the model so that both its interactions with the other elements of  $\Sigma$  and its isolated actions are taken in account: the interactions cover the controlling instructions which  $\pi$  sends to its environment and the phase when  $\pi$  is "watching" for its environment in  $\Sigma$  in order to prepare the internal model. The isolated actions cover the building and run of internal models. The internal models are used many times during the existence of  $\Sigma$ . Each of them should be generated and start to reflect the instantaneous situation in  $\Sigma$ . As the situation can vary, the initial structures of the internal models can differ. Therefore, considering reconfiguration process based on forward looking simulation the principle of, reflective simulation, that enables the evolution of such a reconfigurable system to be evaluated, is very appropriate.

Implementing reflective simulation requires avoiding problems like more simulation time axes, the same languages used for the external and internal models, which concern different "worlds" that must communicate but that must be secure against erroneous mutual mixing, etc... These problems were successfully solved, using the properties of SIMULA language and the principles (block orientation, transplantation avoidance, model copying) presented in (Kindler 1994).

### Simulation of reconfiguration process

In the case of the design phase of an RMS, it is of great importance to evaluate the reconfiguration process.

In this section, the choice of the appropriate configuration is based on the completion time that is considered as the unique criterion to be satisfied. The general process is the following: External simulation gives the behavior of  $\Sigma$  under  $C_{init}$ . At time  $t_c$ , a failure occurs. External simulation is handled. Internal models are updated according with the products and actuators positions given by the external model. They also get parameters from the knowledge of one configuration  $C_i$ . Internal simulations are performed to anticipate the behavior of  $\Sigma$  under each  $C_i$ . Then a configuration is chosen and external model gets parameters from the knowledge of the configuration  $C_c$ . External simulation runs again to give the behavior of  $\Sigma$  under  $C_c$  at time  $t > t_c$ .

Even if the two models external and internal are similar, the number of internal simulations may vary depending on two parameters:

- The number of configurations to be evaluated ( $|\text{SC}|$ );
- The determinism of the models.

The different cases are detailed in the following. For the sake of simplicity, the number of configuration to be evaluated is equal to 2.

*Type 1: deterministic internal & external models*

This expresses the case where no difference exists between real system and the representation of system given to the decisional module. The number of internal anticipations is equal to  $|\text{SC}|$ . At time  $t_c$ , external simulation is handled. Internal simulation 1 is performed to anticipate the behavior of  $\Sigma$  under  $C_1$ . This anticipation is carried out until the batch is completed (noticed  $t_1$ ) or until a predetermined time  $t_f$  ( $t_1 < t_f$ ). Then internal simulation 2 is performed to anticipate the behavior of  $\Sigma$  under  $C_2$ . It runs until  $t_f$  or  $t_1$  or until time  $t_2$  when the batch is completed under this configuration ( $t_2 < t_1 < t_f$ ). If both internal simulations stop at  $t_f$  then, the chosen configuration is the one that enables to manufacture the greater number of parts. If both internal simulations stop at  $t_1$ , then  $C_1$  is the chosen configuration. If internal simulation 2 stops at  $t_2$  and internal simulation 1 stop at  $t_f$  or  $t_1$ , then  $C_2$  is the chosen configuration (Figure 6).

Notice: When external simulation runs again under  $C_c$ , there is no difference between the internal simulation of  $C_c$  and the external one because the models are exactly the same.

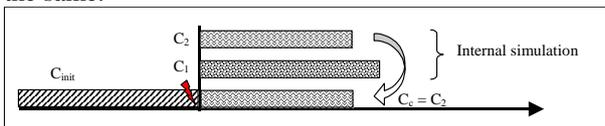


Figure 6: reflective simulation - type 1 or 2

*Type 2: deterministic internal model - nondeterministic external model*

This expresses the case where the real system may have some variation (for example in its machining times) but the representation of the system used by the reconfiguration module is deterministic. If the simulation gives good results, this shows that a simulation based on a deterministic model is sufficient. The sequence of external internal simulations is similar as for the preceding case. On the other hand, the behavior of  $\Sigma$  under  $C_c$  given by the external simulation differs (slightly or not) from the one given by internal simulation. In Figure 6, the time completion of  $\Sigma$  under  $C_c$  might not be exactly the same as the time completion of  $\Sigma$  under  $C_2$ .

*Type 3: nondeterministic internal model - deterministic external model*

This expresses the case where the decisional module computes a non-deterministic model. In that case, a single look-ahead simulation experiment is not sufficient. In that case, performing reflective simulation also enables to find the suitable number of internal simulation experiments.

The procedure slightly differs. As the internal model is not deterministic, several internal simulations have to be performed to get a quite good view of the anticipation of  $\Sigma$  under  $C_i$ . The number of simulation experiments  $N$  has to be determined as well as the procedure to get the “global” completion time issued from  $N$  experiments. This problem refers to data analysis or decisional statistic.

Let us come back to the procedure. At time  $t_c$ , external simulation is handled. For each configuration,  $P$  experiments are carried out. Internal simulations  $IS_{11}$  to  $IS_{1P}$  are performed to anticipate the behavior of  $\Sigma$  under  $C_1$ . Practically,  $IS_{ij}$  uses the same laws. Only the seeds differ. Based on these possible anticipations, a completion time the batch can be determined (noticed  $t_1$ ). Then internal simulations  $IS_{21}$  to  $IS_{2P}$  are carried out to anticipate the behavior of  $\Sigma$  under  $C_2$ . This enables to obtain a completion time under configuration  $C_2$  (noticed  $t_2$ ). The comparison of  $t_1$  and  $t_2$  enables to determine  $C_c$ .

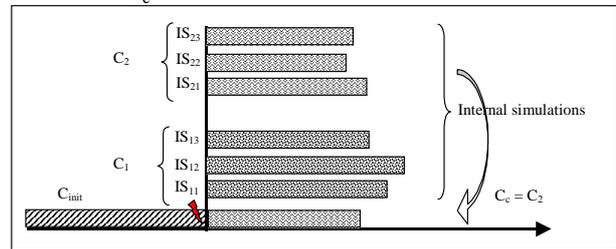


Figure 7: reflective simulation - type 3

**Experiments**

A computer modeling tool has been developed based on flow simulation using discrete time description. It accepts in input the physical architecture of the system, the logical architecture of the system, its current configuration based on the active logical operating sequences, active operations and the future configurations to be evaluated based on sets of operations. It provides the transit time for the products and the best configuration according with the time completion of the remaining production.

More precisely, the initial input of data allowed the operator to introduce several configurations in case of failure occurrence of an element of the physical architecture. For each configuration, the number of internal repetition is also an input parameter. After that, the external model goes on with the best variant. The tool provides a large spectrum of simulation

experiments enabling to evaluate reconfigurations based on the anticipation of the behavior from type 1 to type 3.

## CONCLUSION

Two uses of simulation have been presented. The first one contributes to increase the verification of a large panel of controls associated with different configurations. The second one helps for reconfigurable systems design. Comparing the system with and without configuration change, it can be used to prove that reconfiguration provides advantages. Reflective simulation can also be helpful for defining parameters used for decisional procedures that will choose the new configuration.

This paper points out interest of both continuous and flow simulation for analysis of reconfigurable systems. These simulations are mainly used during the design phase. But as reflective simulation refers to look-ahead simulation, these last is clearly promising for the determination of the new configuration when the environment is changing.

RMS are treated as examples, but other fields such as disabled people assistance, embedded systems could be successfully investigated with these techniques.

## REFERENCES

- Auguin, M., K. Ben Chehida, J.P. Diguët, X. Fornani, A.M. Fouilliant, C. Gamrat, P. Kajfasz and Y. Le Moullec. 2003. Partitioning and Co Design tools & methodology for Reconfigurable Computing: The EPICURE philosophy. In: The Third International Workshop on Systems, Architectures, Modeling Simulation SAMOS03.
- Berruet P, A.K.A. Toguyeni, S. Elkhatabi, E. Craye. 2000. Toward an implementation of recovery procedures for flexible manufacturing systems supervision, *Computers in Industry*, Vol 43, pp 227-236.
- Berruet P, J.L Lallican, A. Rossi, J-L. Philippe. 2005. A component based approach for the design of FMS control and supervision", *IEEE SMC 2005, Hawaii*, pp. 3005-3011.
- Cardin O, P Castagna. 2006. Handling uncertainty in production activity control using proactive simulation, *INCOM 2006*.
- Combacau, M., P. Berruet, E. Zamaï, P. Charbonnaud, A. Khatab. 2000. Monitoring and Supervision of Manufacturing Systems. In: *IFAC MCPL2000*, pp. 348-353.
- Copalp, *STRATON handbook*, 2002-2004.
- Cotting, C. and J.J. Burken. 2001. Reconfigurable Control Design for the Full X-33 Flight Envelope. In: *AIAA Guidance, Navigation and Control Conference*.
- Drake G R., Jeffrey S. Smith. 1996. Simulation system for real-time planning, scheduling, and control, *Winter Simulation Conference*.
- Gupta A K., A I Sivakumar, S Sarawgi. 2002. "Shop floor scheduling with simulation based proactive decision support", *Winter Simulation Conference*.
- Ju-Yeon J., Yoohwan K., Andy P., Wyatt S. 1997. Virtual Testing of Agile Manufacturing Software Using 3D Graphical Simulation, In proceedings of the 1997 IEEE International Conference on Robotics and Automation, New Mexico, pp. 1223-1228.
- Kamimura, A., S. Murata, E. Yoshida, H. Kurokawa, K. Tomita and S. Kokaji. 2001. Self- Reconfigurable Modular Robot. In: *IEEE/RSJ Int. Conference on Intelligent Robots and Systems IROS2001*, pp. 606-612.
- Kindler E. 1994. Simulation of systems containing simulating objects, *Simulation und Integration'94*, P. Lorenz (Ed.), Magdeburg – Dortmund, ASIM, pp. 65-76.
- Kindler, E. 2000a. Nesting Simulation of a Container Terminal Operating With its own Simulation Model. *JORBEL (Belgian Journal of Operations Research, Statistics and Computer Sciences)* 40, 169-181.
- Kindler, E. 2000b. Chance for Simula. *ASU Newsletter* 26, 2-26.
- Kindler E., T. Coudert, P. Berruet. 2004. Component-Based Simulation for a Reconfiguration Study of Transitive Systems, *SIMULATION: Transactions of The Society for Modeling and Simulation International*, Vol. 80, No. 3, pp. 153-163.
- Kotay, K. and D. Rus. 1999. Locomotion Versatility through Self-reconfiguration. In: *Robotics and Autonomous Systems*, Vol. 26-2,3, pp. 217-232. Elsevier.
- Lallican JL, P. Berruet, J-L. Philippe. 2005. SimSED : a tool for modeling and simulating transitive systems, *I3M CMS 2005, invited session, Marseille*, pp.199-204.
- Mitchell, S., H. Naguib, G. Coulouris and T. Kindberg. 1998. Dynamically Reconfiguring Multimedia Components: A Model-based Approach. In: *8th ACM SIGOPS European Workshop*.
- ODE user guide, 2001-2004 RUSSELL SMITH.
- Page, E. 1994. *Simulation Modeling Methodology : Principles and Etiology of Decision Support*. These de doctorat, Department of Computer Science, Virginia Tech., Blacksburg, Virginie, USA.
- Ray C., C. Claramunt. 2003. A distributed computing system for the simulation of disaggregated data flows, *Knowledge-Based Systems*, volume 16(4), pages 191-203, Elsevier Science publications.
- SIMULA Standard. 1986. SIMULA a.s., Oslo.
- Tomizuka, M. 2002. *Mechatronics: From the 20th to 21st Century*. *Control Engineering Practice* 10, 877-886.
- Wenbin Z., Xiumin F., Juanqui Y., Pengsheng Z. 2002. An Integrated Simulation Method to Support Virtual Factory Engineering », *International Journal of CAD/CAM* vol. 2, No. 1, pp. 39-44.
- Wills L., S. Kannan, S. Sander, Mu. Guler, B. Heck, J.V.R. Prasad, D. Schrage and G. Vachtsevanos. 2001. An Open Platform For Reconfigurable Control. In: *IEEE Control Systems Magazine*. Vol. 21, pp. 49-64.

## AUTHOR BIOGRAPHY



**PASCAL BERRUET** was born in Orléans, France and went to the Ecole Centrale de Lille, where he studied automation, production management and obtained his degree in 1998. He moved in 1999 to the University of South Brittany where he is now leading a research thematic in the field of reconfigurable discrete event systems. His research areas are supervision, analysis, simulation and automatic control generation of reconfigurable systems. His e-mail address is: [Pascal.Berruet@univ-ubs.fr](mailto:Pascal.Berruet@univ-ubs.fr).