



HAL
open science

Contribution of simulation in the product-driven systems production activity control

Olivier Cardin, Pierre Castagna, Etienne Chové

► **To cite this version:**

Olivier Cardin, Pierre Castagna, Etienne Chové. Contribution of simulation in the product-driven systems production activity control. 17th IFAC World Congress, 2008, Seoul, South Korea. pp.CDROM. hal-00620877

HAL Id: hal-00620877

<https://hal.science/hal-00620877>

Submitted on 8 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contribution of simulation in the product-driven systems production activity control

Olivier CARDIN*
Pierre CASTAGNA**
Etienne CHOVE***

*IRCCyN – 1, rue de la Noë – BP 92 101 – 44321 Nantes CEDEX 03

FRANCE (Tel: + 33-(0) 228-092-113; e-mail: olivier.cardin@ircsyn.ec-nantes.fr).

** IUT de Nantes - Département QLIO – 2 Avenue du Prof Jean Rouxel – BP539 – 44475 Carquefou

FRANCE (Tel: +33-(0) 228-092-110; e-mail: pierre.castagna@univ-nantes.fr).

*** IRCCyN – 1, rue de la Noë – BP 92 101 – 44321 Nantes CEDEX 03

FRANCE (Tel: + 33-(0) 228-092-116; e-mail: etienne.chove@ircsyn.ec-nantes.fr).

Abstract: While promoting the emergence of product driven production activity control concept, the development of RFID technologies had many benefits on numerous elements of the Supply Chain. Indeed, product driven systems enable a great flexibility by decentralizing the decision centres. However, this decentralization does not allow any more to have a global vision, generally necessary for the system's production activity control. We suggest in this article to use simulation to obtain this global vision, in preparation for providing behaviour prevision functions.

1. INTRODUCTION

Today, agility in supply chain requires a quick response to the changes, not only in the quantity but also in the kinds of products, without losing the productivity. To achieve this challenge, the structure of control for production facilities is radically changing. One of the trends in production control consists in improving existing manufacturing production planning and control systems. We gradually slip from a predictive control to a reactive control. Product-driven control in manufacturing system falls under this evolution.

For a long time, the planning and scheduling functions made it possible to foresee the dates of realization of each production activity. Then, the person in charge of production had, with much of difficulties, to carry out that pre-established planning. This way of making was very rigid because the decisions were global and frozen. It was very difficult to react to uncertainties and risks.

Conversely, product-driven control does not need the realization of a scheduling. The product circulates in the production system and seeks itself for the resources able to carry on its recipe (Pétin *et al.*, 2007). This new way of control was made possible thanks to two significant evolutions.

The first one consists in the emergence of distributed, intelligent control systems (Trentesaux, 2002). Classical centralized and hierarchical approaches based on time or constraints aggregation show limitations to solve the manufacturing control problem, particularly when the system faces numerous random events. Mass customisation, small volume and/or high variety order management require new approaches. These approaches support the use of cooperative and autonomous units, self-organised in an open structure to

offer a very high operational and structural level of flexibility. It implies the use of distributed artificial intelligence (Multi-Agent Systems, MAS) to be implemented.

Besides, advances in automatic identification, in particular radio-frequency identification (RFID), enable to associate some easily accessible data to a product (Cea *et al.*, 2004). Using RFID tags, the product might become an information-storing and processing item (McFarlane *et al.*, 2002).

In chapter 2, we will deal with the decision-making in the context of a product-driven system, noticing there still exists a need for global decisions in a local decisions context. Chapter 3 presents the solution we studied to extend the vision a decision centre can have on a product-driven system. The conclusion of this chapter presents the observer concept, used in chapter 4 for the prevision of the future behaviour of the system. Finally, last chapter presents an application on Quest of all these concepts on an industrial system.

2. DECISIONS IN A PRODUCT-DRIVEN SYSTEM

Due to its distributed intelligence, product-driven control replaces predictive decisions by reactive and local ones. We could think that global decisions disappeared. It is not true. Indeed, at the root of the concept of supply chain, there is a necessary global vision of the industrial system. A major goal of the supply chain is to "improve the flow of material between suppliers and customers at the highest speed" (Proth, 2006). This goal suggests making global management decisions, which are not removed by product-driven control. In addition, the local decisions sometimes require the knowledge of the global behaviour of the system.

In a product-driven control system, the product utilizes decision rules. To make its decision, the product applies these

rules to its parameters to know what has to be chosen. But in the majority of the situations, the product is not the only actor of the decision-making. The decision is made by both the product and the production line decision centre. Let us present the example of a ranking criterion in a queue. A product arriving in a queue has to decide its rank in this queue. From its parameters (Due date, priority...), it applies a ranking criterion (SPT, EDD, CR/SPT...) to know its position in the queue. But it is the decision-making centre of the production system that chooses the ranking criterion which has to be used by the product. Thus, the decision is made on two hierarchical levels.

In the problem solving procedure, a decision centre has a set of given alternatives and is aware of the consequences of each. In a general way, a decision requires two conditions. First, it is necessary to know the whole parameters influencing the decision-making. Then, the decision maker must have a forecasting tool allowing him to foresee, in the future, the consequences of his decision.

To make these global decisions, the managers need to have a global vision of the production system behaviour.

The problem, in product-driven systems, is that the global behaviour results from the whole local decisions which will be made in real time. It is thus very difficult for the manager to know the current state of his production system (Vision) and foresee its future evolution (Prevision).

3. VISION

Our first objective was to determine an efficient way of extending the vision that can be obtained on a product driven production system. We chose to examine three main possibilities to achieve this goal. The following paragraphs present these possibilities. Their conclusions deal with their applicability conditions.

3.1. Data directly coming from the production system

First solution we studied was to obtain the data directly from the production system, through the floor control, the MES, or higher components of the control architecture such as the ERP or the APS (Fig. 1).

This solution provides obvious advantages: the data obtained are reliable, and represent the actual present state of the system. Furthermore, it only requires a minimal additional architecture to work properly.

The most important component of this data collection is the performance of the sensors settled on the system. Some of them are able to report continuously the evolution of a variable: a temperature sensor may for example give the continuous conditions inside an oven. However, in general, a lot of data are not measured in this way. Let us consider for example the position of an Automated Guided Vehicle (AGV) inside a workshop. Sensors settled at certain points of the network (intersections, work stations, etc.) are sufficient to drive the fleet.

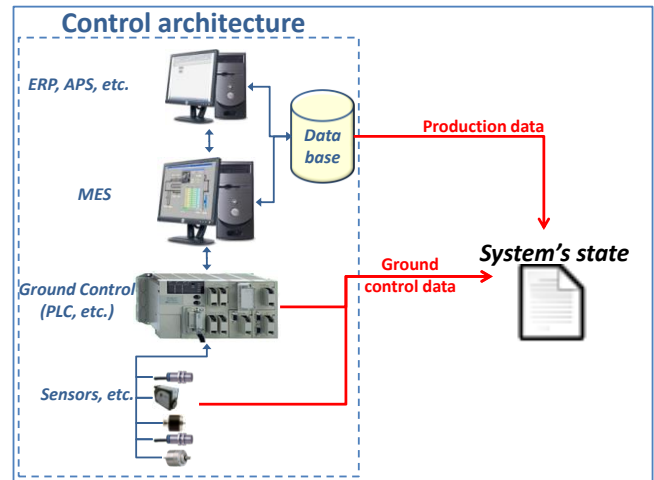


Fig.1. Data directly coming from the production system

The remaining problem is that the use of such devices creates a great uncertainty. Indeed, when an AGV faces a sensor, it is possible to know its position at the current date. As soon as it moves between two of these sensors, its position is totally unknown (except for the information that it is situated on the track between the sensors), and its arrival date in front of the next sensor is unknown a priori. It is therefore not possible to have the whole fleet of AGV in front of sensors of the system each time the information of their location is needed. As a matter of fact, the ground control, the MES, etc. only have a partial view of the system's state: this is what we defined as the space and time uncertainties in (Cardin *et al.*, 2006).

In this work, we chose to use the only sensors strictly necessary to the ground control. To explain this, let us consider again the previous example. The use of a GPS module, or a video camera coupled with an image analyser is possible to solve the problem of positioning of AGVs in a workshop. But, this type of solution is quickly expensive as it would generally be necessary to add several of these technologies to know the complete set of data needed to build the entire state of the complete system.

The conclusion we drew of the use of this solution is that it is not applicable in the general case as many data, essential in the definition of the state, are missing or unavailable through the sensors settled on the system. However, it could be recommended in the rare cases enabling its optimal use, as the data is particularly reliable considering the relative simplicity of the setup. We may cite as a good example of application field the Air Traffic Control (Rogers *et al.*, 1991), where position, flight plan and current trajectory of each aircraft are perfectly known at any time.

3.2. Using a simulator

For the last few years, most of the simulation software editors provide "real-time" editions of their tools. These tools have the specificity to run at the same speed as the wall clock, in opposition to classical software in which the simulation engines run as fast as possible. They were developed in order to meet the industrial requirements to be able to use simulation as an emulation of the production systems. The

initial idea was to reduce the setup time of complex production systems by making tests of the control system directly on this emulation, without waiting for the existence of the physical system.

Our idea here was to use such a simulation running in parallel with the production. This simulator is meant to react to production solicitations coming from the higher layers of the control architecture (MES, APS, etc.), in the same way the real system does. Doing so, it creates a permanent “image” of the system, which state can be considered as being the state of the real system at any time (Fig. 2).

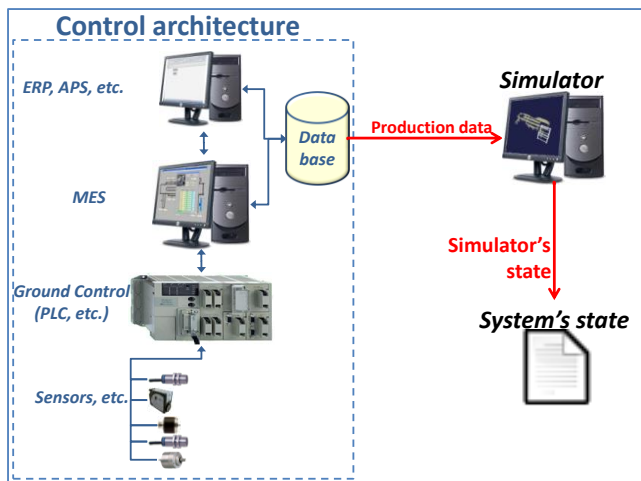


Fig.2. Using a simulator

This solution has two main advantages compared to the exclusive use of data directly coming from the production system. First, it suppresses the space and time uncertainties problem that we defined in the previous section. About the example of the movement of an AGV previously presented, the simulator enables to know its current position at any time for example. Furthermore, the state gathered with this method is well formatted for a later computer treatment. Indeed, gathering the state is not useful if it is not possible to treat the data in an effective way. On the other hand, there are several problems in the application of this solution. First, the initialisation of this simulator must be done with a perfectly known state. Typically, this state is chosen “idle and empty” because it is the easiest state to express by hand. Secondly, inaccuracies in the system’s model and possible hazards inevitably occurring along the production cause a deflection on the final results. The additive aspect of this deviation along time would end up having an effect far from being negligible.

To guarantee the applicability of this solution on a given production system, several constraints have to be set. First, the simulator has to be extremely reliable to limit the effects of modelling inaccuracies. Then, it is preferable to deal with systems which regularly go back to a known state (generally the idle and empty state mentioned before). Let us cite as a classical example the systems that are empty at the end of each working day. It is possible on such systems to restart the simulation, and by doing so, reset the deviation. Finally, the system shall have a low hazard occurrence frequency relatively to the length of simulation without reset. Indeed,

these hazards are not taken into account into such a model, and are thus directly deviating the results.

As we can see, the class of systems that may be considered is relatively limited due to the inevitable growing deviation of the results. On the other hand, this is a credible alternative for applications that cannot afford the equipments necessary to the previous solution and which do not need a high precision on the results.

3.3. Using an observer

The conclusion we drew at this point was that both solutions previously presented turned to be well adapted to a limited class of systems. As we wanted to make a solution applicable to the widest class possible, we had the idea to hybridize these solutions. By making so, we thought we could get the advantages of the simulator (about the data availability) and of the direct use of real system’s data (about the data reliability).

The Fig. 3 presents the principles of such a solution. Compared to the previous solution, the simulator is replaced by an observer, meant to deal with the data coming from the ground control. Using these data, it is able to reset the deviation of its predictions. As a matter of fact, this observer is meant to run at the wall clock speed (real-time) as the simulator does.

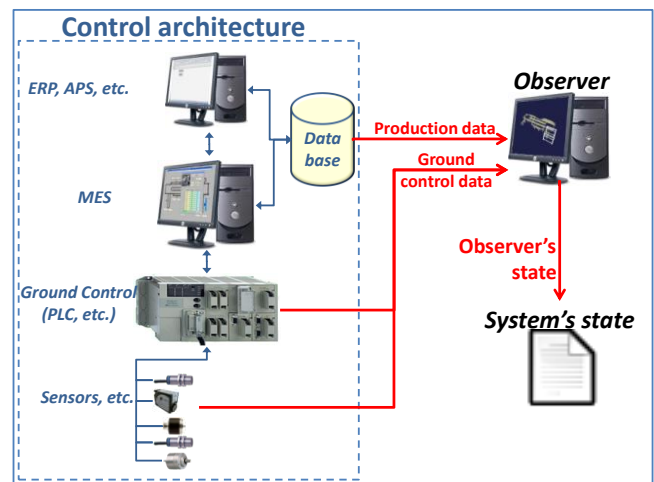


Fig.3. Using an observer

We believe that this solution is the most powerful that can be implanted with the constraints presented before. But, even if the implantation costs are relatively low, this solution is not so easy to implant and is thus made for large facilities to justify a fast Return on Investment (ROI). The purpose of this paper dealing with the product driven production systems, this constraints is very often respected as these systems are generally relatively large.

Let us now examine the implantation of such an observer. A lot of technologies might be used, according to the objectives that were designed. We chose to use discrete-event simulation. Indeed, a lot of simulation pieces of software meet the requirements of our study. First, it was widely used to

model the behaviour of such production systems. As a matter of fact, the model that was eventually made for the design of the facility can be used again in the production phase. This implies an interesting diminution of the investment time and costs. Then, the available means of communication are generally very well adapted to the communications inside such control architecture. Their graphical user interfaces is also very interesting, as it enables a clear vision on the behaviour of the system for the operator.

Finally, the state gathered on the observer is particularly well adapted to use to initialize other simulations in order to predict the future behaviour of the system. This is what is described in the next section.

4. PREVISION

We suggest in this section the use of a discrete event simulation based decision support tool. In the end of the 90s, several studies were lead about the future of discrete event simulation. Their main idea was to switch from a design aimed simulation to a decision support simulation. This concept was named in (Davis, 1998) as “online simulation”.

Product-driven systems local behaviour is generally simple to express, as it is made of simple rules that each agent respects. Thus, the global evolution of the system starting from a perfectly known state is easily predictable (in absence of hazards) by means of simulation. The remaining problem is the knowledge of this initial state. The observer is of course very helpful to solve this problem, as the data contained in the system’s state it provides perfectly fits with the data needed for the initialisation of the simulation. Simulation can be interfaced with almost every component of the architecture, starting with the PLC to the APS. When one of them asks for a simulation, the computation starts. First step is the gathering of the state of the observer, considered as the best possible approximation of the actual state of the system. When the computation is over, the results are sent back to the element that asked for the simulation. Fig. 4 presents this solution, with a simulation request coming from the MES and the results getting stored in the database. Voluntarily, all the communication means are not represented on the figure to avoid overloading it. As we said before, the primary aim of our work was to support a human operator in his decisions along the production. As a matter of fact, the architecture we suggest opens the possibility to add an optimization algorithm. Furthermore, it happens to be possible to couple these tools directly with the ground control in order to extend the rules involved in the control of the system. Indeed, from now on, it is possible to make a decision taking into account the past events and the prevision of the future events happening on the system. The rules may therefore extend their vision, which is generally typically local.

5. AN APPLICATION EXAMPLE

5.1. The assembly line

The assembly line was built for educational and research purposes by the Institut Universitaire de Technologie of

Nantes (Fig. 5). This job shop production system is made of six workstations. The goods are transported with pallets, which move on unidirectional conveyors. The pallets will be called “transporters”.

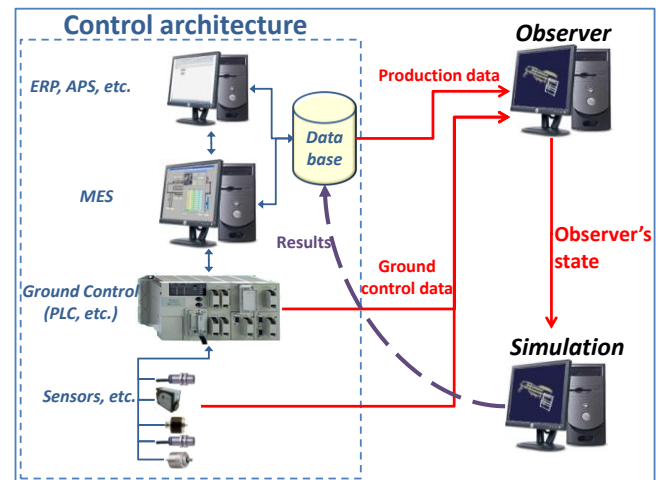


Fig.4. Relationship between simulator, observer and control architecture

A transporter storehouse (an accumulation conveyor) enables the storage of the free transporters. The 42 transporters are equipped with smart tags. The production data of the transporter are written on the tag when it leaves the storehouse: number of products to transport (sequentially), recipe of each product in term of operations, etc. In parallel, each station has a list of operations it is able to perform. Therefore, when the transporters move on the main loop and arrive at the entrance of a station, a comparison between the next operation of the recipe and the list of operations the station is able to perform is made. According to the chosen rule, the transporters may enter the station or continue on the main loop.

5.3. Online simulation

This kind of working enables a distributed approach of the control, as all the decisions after the transporters leave the storehouse are made on a local point of view. On the other hand, when a decision has to be made by a human operator, it is very difficult for him to evaluate the situation, and even more to foresee the consequences of his decisions on the system’s behaviour.

The particular example we developed is about the calculation of due dates of production orders according to the orders’ parameters. The main parameter on which the human may act in a production order is the number of transporter that is allocated to this order. This number determines the number of products the transporter will transport, and therefore will have a great influence on the makespan of the order. For example, to produce 40 products, is it better to allocate 1, 40 or any other number in between of transporters? A low number extend the makespan of the order, but has a low influence on the other orders on the system, whereas a high number enable a high parallelism between the operations, but overload the system.

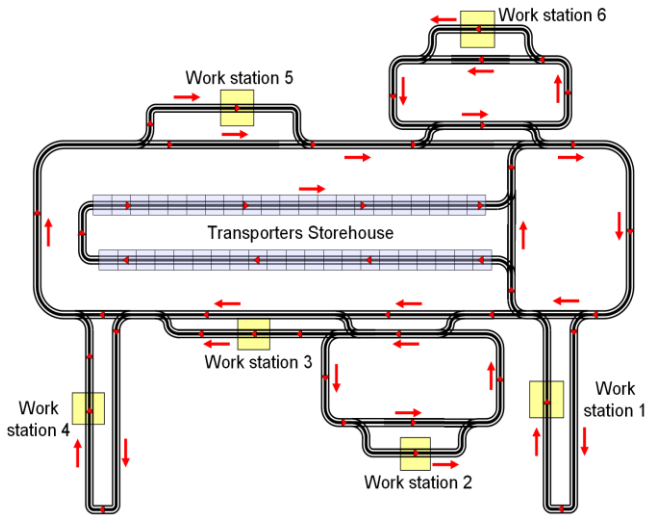


Fig.5. The job-shop production system

The whole tool is controlled via the MES screen. Indeed, it is meant to be used by non-specialists of simulation, and a special effort must therefore be made on the human-machine interface. Our application is relatively easy to use. Each time the operator clicks on the simulation button, the figures representing the simulated end date of each order of the orders table are updated with the new simulation results. These orders are either running or still not launched. This last feature enables the operator to evaluate the pertinence of the parameterization of these orders.

Behind the screen, after the operator pushed the button, the simulation model (simulator) starts gathering all the information needed for this run, including the orders table or the stations parameters for example. These data are located inside the MES database. On the other hand, it contacts the observer to ask its actual state. The observer copies its state in external text files, and then acknowledges the request of the simulator. At this point, the simulator starts initializing at the correct state collected through the text files. When the simulation ends, the results are stored in the database, and an acknowledgement is sent to the MES. The results are then displayed on the screen. In our application, for orders of about an hour long, the results are displayed less than five seconds after the operator pushed on the simulation button.

5.2. The Quest observer

The objective of the observer is to extract, at any date, an estimation of the system state, using a simulation model and the piloted system trace. The main problems of using Quest for this application are external communications, real-time management and synchronization of simulated transporters according to the system trace. In this part, we present those problems and the solutions we developed in our application.

The first problem is the lack of communication media between Quest and external applications. There are two possibilities offered by Quest for synchronous exchanges with others applications: TCP/IP sockets and pipes. Pipes are very well supported on UNIX systems, but difficult to use on Windows platforms. We chose to use sockets to communicate

with sensors. We had to use intermediates between the observer and the sensors due to incompatible data formats (Fig. 6).

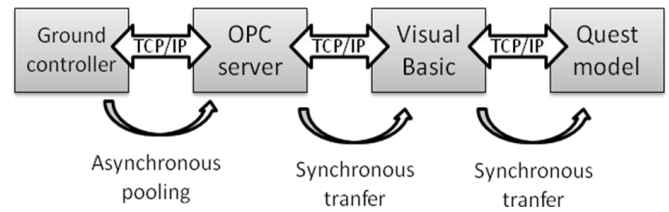


Fig.6. Communication from ground controller to Quest

The second problem met is the synchronisation of the simulation clock with the world clock. Quest, in opposition with others simulation tools such as Siman Arena RT, does not include any real-time module. The solution we suggest is to incorporate a dummy element in the Quest model, piloted by a customised BCL program. This element connects itself to an external time server and controls the INTERVAL parameter to accelerate or decelerate time stream (Fig. 7).

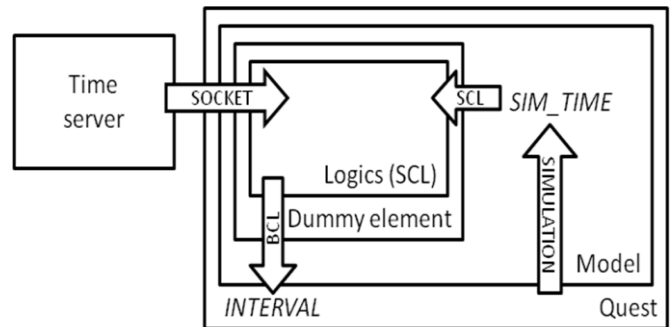


Fig.7. Quest real-time overriding control

This overriding control links the Quest model clock and the world clock. In consequence, it reduces the time stream gap between the controlled system and the observer model.

The last problem is to synchronise the Quest transporters according to the controlled system events. In this application, we chose to respectively model the real conveyors, the shuntings and the transporters with some Quest conveyors, decision points and parts. The advantages are the use of accumulation capabilities, speed control and routing solutions. The two main problems we encountered are to stop a part when the corresponding transporter is late or stopped, and to move the part when the transporter is seen at another location than the one the observer had forecasted. The first problem is solved by blocking the parts in the decision points' logics. The second one is solved by connecting "Out decision points" every x millimetres to "In decision points" through a buffer (Fig. 8). When a transporter is identified at an unforeseen location, the corresponding part is marked and extracted from the nearest "Out decision point". The buffer routes the part to the point the transporter was seen.

5.2. Precision of the architecture

When studying the results of a set of online simulations, the question of the precision of the forecasts is essential.

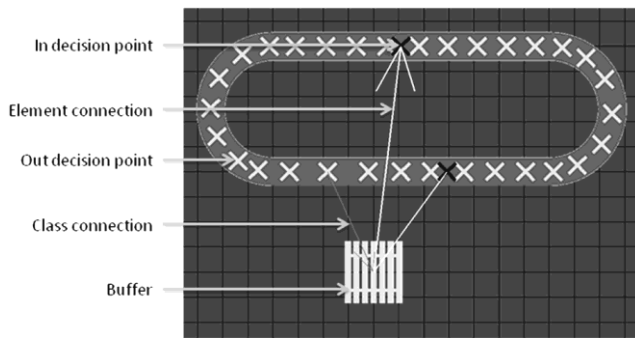


Fig.8. Decision points on a Quest conveyor

Thus, we tried and identify the main causes of uncertainty in our own architecture. First, as long as we could have worked on the simulation model, it cannot be the exact representation of the reality: some hypotheses inevitably have to be made, which make the model behaviour diverge from the actual one.

Then, the initial state of the simulator is not the exact present state of the system. A first reason for this is that the data acquisition is not immediate, but a gap of time exists between an event on the system and its detection by the observer. We made some benchmarks on the tardiness of the acquisition chain presented in section 5.2 (Chove, 2007). The measured tardiness, with optimised OPC configuration, can be represented by a normal law with an average of 100 ms and a standard deviation of 50 ms. A second reason is the inaccuracy of the modelling of the system in the observer, which makes the prevision of behaviour not accurate. Both of these reasons on our application give an estimated global inaccuracy of less than 500 ms in average, which represents a distance of 10 cm of travel for the transporters.

The problem we encountered to evaluate the global uncertainty is that a slight inaccuracy on the initial state of the simulator may have a great effect on the global behaviour of the system. In order to avoid these side effects, we chose to run a full test on the prevision of the end date of a production order on our application. To do so, we ran a production order on the line, and we simulated every 3 seconds the end of the production. We then compared every simulated date with the real one, obtained after the real production is over. This test was run with several orders, representing different loads of the system. As a result, we noticed that the simulator's previsions were globally less than 5% far from the actual data (Table1). Furthermore, the uncertainties tend to decrease as the order goes, as the simulation length decreases.

6. CONCLUSION AND PERSPECTIVES

The aim of this paper was to present how simulation could be a powerful tool for the production activity control of a product-driven system. This tool is meant to support a human operator who needs to make a decision about the global behaviour of the system. As a matter of fact, this paper is focused both on a Vision support, enabling the operator to have an overview of the system's present state at any time, and on a Prevision support, enabling him to foresee the alternative consequences of the decisions he could make.

Table 1. Error evaluation of the previsions

System's load	40%	60%	80%	100%
Mean	1.63%	1.18%	1.08%	1.10%
Standard Deviation	2.77%	2.98%	2.45%	1.64%

This tool is based on an observer for the Vision and a simulator for the Prevision. The observer is also used in the Prevision phase to provide an initial state to the simulator reflecting as accurately as possible the present state of the system. Furthermore, building the observer with the same language as the simulator is enables great data compliance between these two components.

At the present time, our work is particularly focused on the production phase of the supply chain. Future directions of this work will extend the concepts we developed here to the whole supply chain, in order to be able to foresee more accurately the consequences of the operator's decisions on the whole chain.

REFERENCES

- Cardin, O. and P. Castagna (2006). Handling uncertainty in production activity control. In: *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2006*, Saint-Etienne, France, 579-584.
- Cea, A., and E. Bajic (2004). Ambient Services for Smart Objects in the Supply Chain Based on RFID and UPnP Technology. In: *Proceedings of the Third Conference on Management and Control of Production and Logistics MCPL'04*. Santiago de Chile.
- Chové, E. (2007). *Observateur par simulation sous Quest*, Master Thesis, Université de Nantes.
- Davis, W. J. (1998). Online simulation: Need and evolving research requirements. In: *Handbook of simulation* (J. Banks (Ed)), 465-516, John Wiley and Sons, New-York.
- McFarlane, D., J. Carr, M. Harrison and A. McDonald (2002). Auto-ID's three R's: Rules and Recipes for product Requirements. *Auto-ID Centre White Paper*
- Pétin, J.-F., D. Gouyon, and G. Morel (2007). Supervisory synthesis for product-driven automation and its application to a flexible assembly cell. *Control Engineering Practice*, **15**(5), 595-614.
- Proth, J.M. (2006). Scheduling: New Trends in Industrial Environment.. In: *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2006*, Saint-Etienne, France, 41-47.
- Rogers, P. And M. T. Flanagan (1991). On-line simulation for real-time scheduling of manufacturing systems. *Industrial Engineering*, **23**, 37-40.
- Trentesaux, D. (2002). *Pilotage hétérarchique des systèmes de production*, HDR Thesis. Université de Valenciennes et du Hainaut-Cambresis.