Defining A Command Architecture Enabling Proactive Simulations On A Complex Manufacturing System

Olivier Cardin, Pierre Castagna

CMS'2005
Defining A Command Architecture Enabling Proactive Simulations On A Complex Manufacturing System

O. CARDIN, P. CASTAGNA
IRCCyN – Institut de Recherche en Communication et Cybernétique de Nantes
1, rue de la Noë BP 92101
44321 Nantes Cedex 01, FRANCE
Olivier.Cardin@irccyn.ec-nantes.fr
Pierre.Castagna@irccyn.ec-nantes.fr

Abstract – **Proactive simulation is a new tool, which can be especially useful for driving complex manufacturing facilities. Indeed, the data collected after such simulations can be very useful to be able to take the best decision in the shortest time as the facility is currently running. This paper provides an insight into the different concepts of proactive simulation. Special emphasis is placed on the place of simulation in the control loop, and especially its relationship with the decisional center. The facility supporting this architecture is also briefly presented.**

I. INTRODUCTION

Facing a constantly evolving market, industries use more and more complex production facilities. The dynamic behavior of such facilities is thus more and more complicated to predict through analytical methods. Nowadays, simulation meets a growing success in the industrial world, even if it does not completely solve the problem: it only gives the response of the system subjected to a set of values of the adjustment parameters. On the other hand, there is no limit in the complexity of the studying facilities. So, simulation is at least a help to solve the problem.

That is why simulation has become a very used tool in the new facilities conception phase, or for the study of an evolution of an existing facility. It enables validation of technical choices, studying the dynamic behavior of the system [Ait Hssain, 2000]. Furthermore, it can help the engineer to understand the behavior of the system, and evaluate different strategies [Law & Kelton, 1982].

This article is focused on the use of simulation as a decision support for the pilot of a manufacturing device (illustrated here by an automated assembly line). As the system needs a great number of decisions to be taken, the simulation tool will provide the administrator a view of the future behavior of the system to take the best decision possible [Pujo, 2004].

In the first part, we will introduce the previous researches lead on the subject and the main benefits of such a tool. Then, we will study the place of proactive simulation in the software environment and the problems that occur. Finally, the link between the real-time simulation and the assembly line itself will be presented, to try and give solutions to the difficulties encountered.

II. THE PROACTIVE SIMULATION CONCEPT

A . Needs calling for simulation

To run a complex system, the administrator needs to use a model enabling him to predict the consequences of his choices. The problem is that production facilities became more and more complex. This complexity stems from four principal origins. The first one is the structural complexity. The production system is made of a lot of elements or departments that interact.

The second one is the flow complexity. The elements of the system exchange a lot of things like products, tools and information.

The third one is the resource planning complexity. The elements of the production system utilize resources that cannot be divided.

The last one is the stochastic complexity. In a production system, a lot of unpredictable events occur. Because of this complexity there are no simple models to predict the dynamic behavior of the system. Solution is to call for simulation technique.

B. Problems Faced

The model simulation is a computer program that predicts the course of events in the system. The use of simulation as a decision support is not easy.

The first difficulty is to have a reliable model. The model behavior must be a reliable image of the real system behavior.

Another difficulty is to capture the initial state of the system at the date to make the decision.

This initial state means a huge quantity of data. For an effective decision system, the capture of initial state must thus be done in a short time.

To do that, the presence of a Manufacturing Execution System (M.E.S) is required. The interest of MES is that it automatically interfaces to the shop floor control layer, which is manufacturing product. Thus, it is the ideal vehicle to automatically collect any and all data needed by the simulation initialization.

The problem is that MES are incapable to provide the real time state of the system at any time, although the “model must be initiated to the current state of the system” [Davis, 1999]. Reference [Mebarki, 2001] shows the difficulty is that the real system is in perpetual evolution, and the initial state of the simulation will thus never be the same.

For example, let us consider a conveyor transporting goods. That conveyor is equipped with two sensors, one at the entrance and the other at the exit. When a product is between the entrance and the exit, we don’t know exactly its position. We only know the position of a product when it is facing a sensor. It’s the spatial uncertainty. We call the points where the position is known the observation points.

Furthermore, we only know the position of a product at the moment it is facing a sensor. It’s the temporal uncertainly.

To know the state of the system anywhere and anytime, we propose to utilize a real time simulator synchronized with the real time system via the MES (Fig. 1).

The role of the real time simulation is to be an observer of the system. It predicts the state of the system when this state is unknown.

This real time simulation is a deterministic simulation. If a stochastic event occurs in the real system, it will be detected later at an observation point. Then the simulator will take into account the stochastic event to adapt itself to the real system.

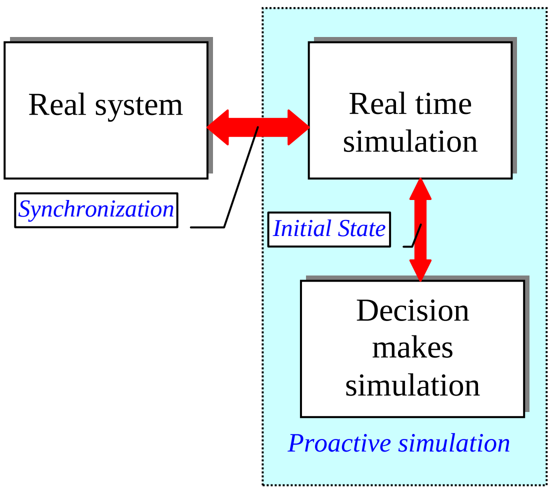

Fig. 1. Using a real-time simulator.

C. The proactive simulation

Then appears the proactive simulation concept: We try to make a simulation in another simulation. In the concept of simulation decision support, the actor of the decision is a human actor.

The next step is to insert simulation in the control loop of an automatic production system. The proactive simulation requires two conditions:

- The capture of initial state must be very quick.
- The simulation must be made in a very short time.

Of course the proactive simulation can take account stochastic events. Then the simulation speed becomes a strong criterion in stochastic systems because it needs several simulations to construct confidence interval to make a good decision.

III. General architecture of the system

A . System without simulation help

This assembly line was built for educational purposes by the Institut Universitaire de Technologie of Nantes (Fig. 2). 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”. A transporters storehouse (an accumulation conveyor) allows the free transporters storage. The 42 transporters are equipped with electronic tags. A typical use of the line could be described in these four steps:

1. An empty transporter leaves the storehouse to reach Station 1. All the information related to the products that have to be made on the transporter are written on the electronic tag.
2. On station 1, the Cartesian robot puts a product on the transporter. The tag is updated.
3. The transporter travels from station to station along the central network. Each time it reaches the entrance of a station, the following operation in the process planning of the product is compared to the operations the station is able to perform. If there is a match and if the station is available (i.e. no failure, no full batches etc.), the product goes into the station.
4. When the process planning is over and the product is off the transporter (generally put off by the Cartesian robot), it goes back to the storehouse.

The stations 1, 3, 4 and 5 have a buffer with a FIFO priority rule.

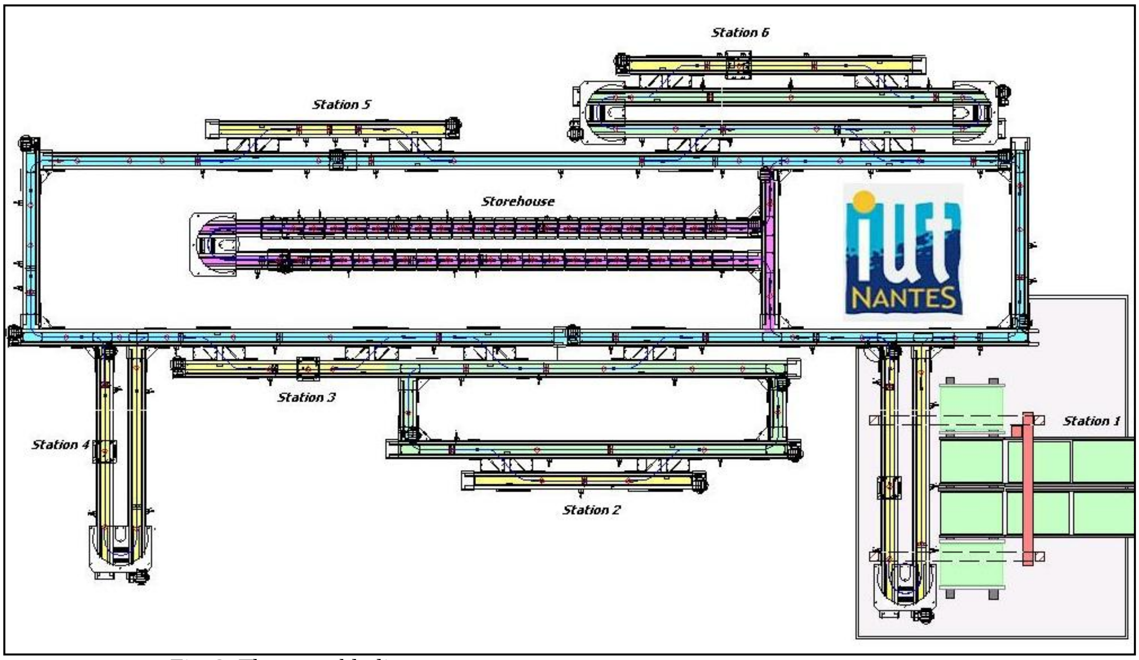

Fig. 2. The assembly line

To help use this line, the MES was developed using INTOUCH software and the Access database. INTOUCH is a module of the Wonderware MES tool.

It is obvious that the command of this system is relatively easy, but due to the complexity of this system, the lack of decision support may lead the operator to dead ends. That is why this system was chosen.

B. Introducing simulation in the architecture

To insert decision support in the system, four different sections are defined:

- The real system (the assembly line)
- The MES (INTOUCH & MySQL)
- The decisional center
- The simulation device

B.1. The decisional center

This center plays several roles in our architecture. The first one is to decide when a simulation has to be done, and which parameters should be used (time of simulation, total time before application of the decision, etc.). A second one is the analysis of the simulation results and of the production data in order to make a choice. This center could be of two kinds: either automatic or composed of one or several administrators.

An automatic center could enable the system to have advanced local scheduling rules. Fig. 3 shows station 6 of the assembly line at a specific time.

The station is settled for operation 10 (corresponding to the operation transporters labeled 1 have to perform). The local scheduling rule is "Clear a Fraction" [Kumar, 1990]. As there are no transporters 1 left on the small loop, the station decided to start and perform operation 30 (needed by the products on the transporters labeled 3). If the settling time is important, this rule is not necessarily the best one. Indeed, the last transporter 1 is arriving at the station. If a simulation had been led before letting transporter 3 get in the work station batch, the decisional center would have had the information, and would have certainly made the transporters 3 wait for transporter 1 to be treated before getting inside the work station.

Such a center could enable the system to be relatively autonomous, but plenty of decisions have still to be made by a human administrator. Indeed, these short rules have to be programmed by advance, which requires being able to model it.

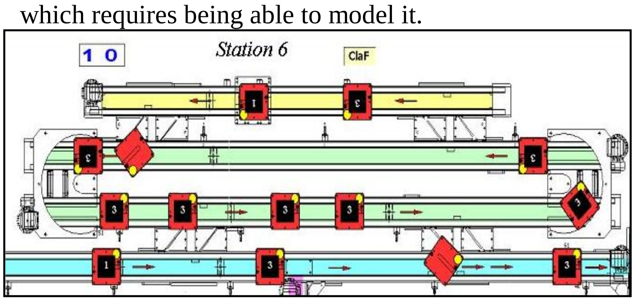

Fig. 3. Station 6 in production.

The other kind of center could enable administrators to apply their knowledge in scheduling and to take into account unforeseen events in the administration of the line. To illustrate these events, let us consider a station breakdown. To be able to end the production, the operations feasible on the station must be reassigned to another station (possibly several). Two different choices can be made: either the operations are assigned to the station which is the closest architecturally speaking, or to the station the less loaded. Obviously, such problems cannot be treated by the automatic center, because of the subjectivity of some criterions. Two simulations will then be run to determine which solution is the best.

All this being said, it is obvious that an automatic center is not sufficient, but it seems possible to make automatic decision center and administrators live together, as long as the automatic decisions are transparent for the user.

B.2. The simulation device.

The simulations will be run on Arena, a product of Rockwell Automation. Its specific Graphic User Interface is a first advantage of such software as it enables operators to have a concrete vision of the simulation without using lots of CPU time. Furthermore, it is relatively easy to establish multiple communications between Arena and the other software of our structure (support of the ODBC connections for example).

The proactive simulations are based on the simulation model of the assembly line, originally made to create the supervision application on INTOUCH before the real system was built.

The complete architecture is shown on Fig. 4, detailing the relationship between all the components.

As the initial state is recorded in separate text files, the administrator can run as many proactive simulations as he wants, therefore test as many choices as he decides to.

This flexibility is interesting as it allows the administrator to change his mind during the decision time (add a $n+1^{th}$ choice to test as the first n are not fully satisfying for example).

Proactive simulations and real-time simulation may be ran on separate computers: the Arena SIMAN engine, which calculates the behavior of the model, cannot be launched twice on the same CPU (except for multiprocessor architectures).

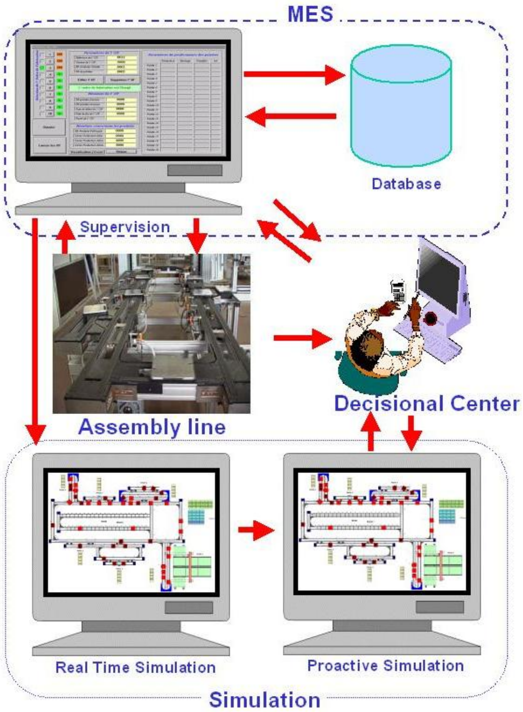

Fig. 4. Detailed architecture.

IV. REAL TIME AND REAL-TIME SIMULATION

As shown previously, the real-time simulation is the key of our architecture. Indeed, its accuracy guarantees the exactness of the proactive simulations and thus the precision of the decisions made. But, as accurate as the model is, it is difficult to take into account as insignificant events, as a transporter blocked in a turn for example. As a matter of fact, it seems essential to link the model with the real system.

Checkpoints are placed on the network at the exact place of the sensors on the line. In this manner, each time a transporter comes through a sensor, the simulation is adjusted: either the simulated transporter is late, and thus an infinite speed is assigned to the transporter until the next checkpoint, or it is in advance, and it is blocked on the checkpoint until the real one comes through.

Several problems can be solved in this way, but some dark points still exist. For example, if a major failure appends on a transporter, and that an operator decides to take it away from the line, it is very difficult to reproduce this behavior in the simulation.

Furthermore, the administrators of the line could want to take into account stochastic problems. For example in the presented line, one or more station could control the product on the transporter. The result of the control cannot be simulated. Indeed, when the real system returns the result of a control, the real-time simulation has to return the same result.

To reach these aims, a permanent communication between the line and Arena has to be established. Beyond the technical problems (all PLCs do not have an Ethernet communication module), making this communication work is a huge challenge. First, it influences the programming of the line. Each time a transporter comes through a sensor, the PLC has to send a message to Arena. That means that the simulation is not an added part of the original architecture anymore, but the links between the different parts are more complex. This could be a problem if the proposed architecture was to be settled on an existing manufacturing line.

A second challenge is about the communication language. Reference [Kouiss & Najid, 2004] outlines the technical realization of such architecture. Obviously, Arena and the PLCs do not have a common language, and they do not even communicate on the same network. As a matter of fact, the MES will be used to enable the communication, as it is made to allow both communication means. Once more, this links the programming of the different parts.

At last, Arena can only receive messages at predefined times of a run. Indeed, during the processing step of an entity (processing of the active entity through the model as far as it can until the entity is either blocked, destroyed, or begins some sort of time delay), Arena cannot deal with the received messages. Considering the comparison between the calculations times of Arena and the evolution speed of the line, it remains plenty of time to add this kind of functions. The proposed solution is to set a system of Questions & Answers: either Arena will seek information from the MES, or the MES will keep on sending the message until Arena responds. The first solution seems to be the best, because it avoids double sends and network overloads. A response of Arena might be considered to avoid the loss of packets in the Ethernet communication.

However, the model will never be the exact image of the line, as the transport between checkpoints is still simulated. As a matter of fact, the proactive simulation will give slightly incorrect answers, as it does not take into account the link with the real system. The size of the assembly line, the speed of the transporters and the distance between consecutive checkpoints allows us to make this approximation, as the simulated times are quite short.

Another cause is the stochastic results: if the real-time simulation can get the result of a control (for example) on the PLCs of the real system, the proactive simulation is not able to (by definition). These simulations are often run on a short (sometimes very short) time horizon – the administrator wants to check the behavior of the system for the next couple of hours or days, rarely more. As a matter of fact, the stochastic events must have append frequently and have a short duration – compared to the duration of the simulation.

V. CONCLUSION.

In this article, we proposed architecture to enable proactive simulation on an assembly line. The simulations can be driven by the pilot of the line, or be launched automatically and be transparent for the user. This technology may enable advanced global and/or local scheduling rules. This architecture needs a large coupling between a real-time simulation, the proactive simulation, a MES tool, a decisional center and the real system.

Lots of problems related to the establishment of such architecture in a manufacturing context are mentioned, and solutions are brought. The assembly line being currently under construction, it is not fully implemented yet. As a matter of fact, unexpected problems will certainly occur, which may lead us to use different solutions.

Several points still need to be enlightened. For example, the automatic proactive simulations will be driven, and their results analyzed, by a decisional center. This center still needs to be designed. The decisional center shall be the center of our future researches, as the interface between the administrator and the proactive simulation is also not designed.

REFERENCES

[Ait Hssain, 2000] Ait Hssain, A., Optimisation des flux de production. Méthodes de simulation, Editions Dunod.

[Davis, 1999] Davis W.J., Simulation: Technologies in the new millennium, Winter Simulation Conference'99.

[Kouiss & Najid, 2004] Kouiss K. & Najid N., Un Couplage MES-Simulation pour le pilotage d'un système de production, MOSIM'04.

[Kumar, 1990] Kumar P.R. & Seidman T.I., Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems, IEEE Transactions on Automatic Control, vol. 35, n°3, pp. 289-298, March 1990.

[Law & Kelton, 1982] Law A.M. & Kelton W. D., 1982. Simulation Modeling and Analysis, McGraw-Hill, New York.

[Mebarki, 2001] Mebarki N., Castagna P. & Gauduel R., Apport de la simulation comme outil d'aide au pilotage des systèmes de production - exemples d'application, MOSIM'01.

[Pujo, 2004] Pujo P., Pedetti M. & Ounnar F., Pilotage Proactif des lignes de production KANBAN par modélisation DEVS et simulation temps réel, MOSIM'04.