Evaluating and improving production control systems by using emulation

A. Pfeiffer, B. Kádár, L. Monostori

Computer and Automation Research Institute,
Hungarian Academy of Sciences
Kende u. 13-17, Budapest, POB 63, H-1518, Hungary
Phone: (36 1) 297-6115, Fax: (36 1) 4667-503,
E-mail: {pfeiffer, kadar, laszlo.monostori}@sztaki.hu

Abstract

The paper describes the possible roles of the simulation and emulation in production planning and scheduling as well as in production control systems through a case study for optimizing the internal logistic system of a given production system. The study focuses on the control of a transportation unit and automated storage and retrieval system. A complex simulation study has been carried out to be able to analyze, improve and evaluate the control system. In our work, we analysed the influence of the scheduling methods currently applied in the (emulated) production system. Parallel to these scheduling methods, new techniques have been tested as well.

Key words

production control, emulation, simulation, evaluating control systems, industrial case study

1 Introduction

Simulation is the art and science of creating a representation (model) of a process or system for the purpose of experimentation and evaluation [1]. With other words: building a model of a real system (or a system-to-be), conducting experiments with this model, and creating output result for decision making and implementation support.

Why should simulation be used in production planning and scheduling systems? Simulation experiments can be conducted for several reasons, but they actually have the same primary purpose; as it is described in [2], all simulations are conducted to be able to make wise decisions in some way. Wise decisions lead to increased efficiency and reduced costs, which are usually two of the main goals of a company. In other words, the main reason for using simulations is to support *decision making.* Some examples of what simulation can be used for are: prediction of system performance, evaluation of certain feature in the system, comparison between several alternatives, gaining knowledge of the system at different life-cycle phases, problem detection, and presentation of predicted results.

As mentioned in [3], the greatest overall benefit of using simulation in manufacturing environment is that it allows a manager or engineer to obtain a *system-wide view* of the effect of “local” changes to the manufacturing system. On the one hand, if a change is made at a particular workstation, its impact on the performance of *this* station may be predictable. On the other hand, it may be difficult, if not impossible, to determine the impact of this change on the performance of the *overall system* in advance.

The potential benefits of applying simulation in production planning and scheduling are as follows:

- Increased throughput, decreased times, reduced in-process inventories of parts, increased utilizations of resources, reduced capital requirements or operating expenses
- Better overview and understanding of the system and system-processes during the model building phase
- „Virtual” statistical data by analyzing results from simulation

Modular architecture is another possible way of cost reduction, by creating reusable simulation or control components using object-oriented tools.

2 Evaluating control systems by using simulation

Based on the possible combinations between reality and simulation, [4] describes four possible approaches to test control systems (Figure 1):

1. The *traditional way* to test control systems. Both control system and logistic system are in the reality. The control system is tested after installation.
2. *Emulation or soft commissioning* is a combination of a control system in reality and a simulated logistic system.
3. *Reality in the loop* is a combination of a simulated control system and a real logistic system.

4. *Off-line simulation*. Both control system and logistic system are simulated.

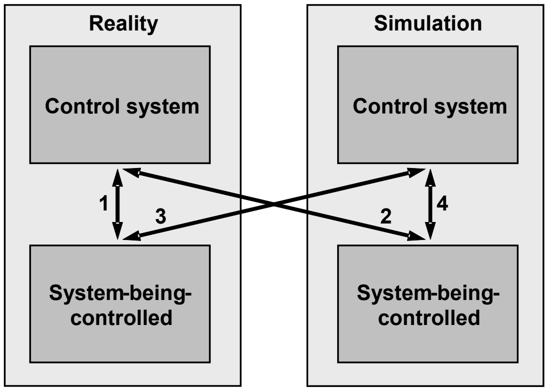

Figure 1: Approaches for Testing Control Systems [4]

As described in [4], most control systems are only completely tested at the shop floor. This is represented by combination 1 in Figure 1. It is difficult to verify and validate a control system before implementing and coupling it with the real system to be controlled. The testing takes place during the start-up phase of the system to be controlled. This is an expensive, risky and error-prone way of developing control systems. Emulation has been developed as a new improved way of testing control systems. Within emulation the real control system is connected to a simulation model that imitates the machines or the production systems. Emulation can reduce the developing time of control systems and thus shorten the time-to-market. It allows testing of control systems faster than real-time and under safe conditions. The conditions under which the tests are carried out can be better controlled, allowing us to study different scenarios the control system has to deal with. The effects of worst-case scenarios and machine break-downs can be easily studied by simulating them. Finally, emulation can be applied training process operators in an easy and safe environment.

3 The role of the simulation and emulation in production planning and scheduling systems

The possible roles of the simulation and emulation in production planning and scheduling as well as in production control systems is shown in Figure 2. To make the problem easier three main levels are defined. One is the level of the physical system, which can be an existing system or just a model of a planned one. The second is the level of the control and execution. This system controls the physical system level, i.e. propagates the scheduled tasks as *commands* to the physical system and receives *reports* from it. Mostly, this level does not have any complex planning or decisions-making function but it has a close connection to the resources at the lower level, this way the possibility to change the behaviour of the physical system. Any change in the state of the lowest level is described by events, and these events will cause reaction in the control system. The highest is the planning level where complex decision-making and scheduling processes are carried out. The plan is executed by the physical system under the control of the second level. The third level gets feed-back information about the plan from the second level. Systems at this level will provide statistical data about the plan, or just some information about the result of it.

Concerning logistic systems the third level is usually very complex. As described in [4], most of the time these systems are only tested at the shop floor after installation. For this reason, a lot of costly failures occur at the start-up stage. In order to eliminate the technical problems in the design phase, modelling and simulation of the whole system is needed. However, in order to model the above three levels in one framework huge compromise is needed. A good solution is to distribute the model of the systems, in such a way as the systems are separated in the reality.

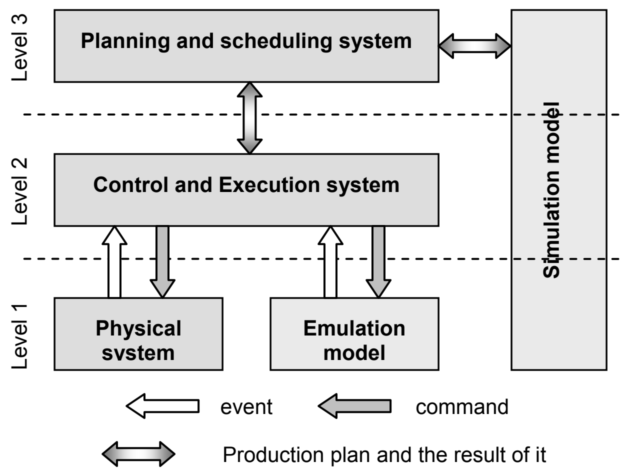

Figure 2: The possible roles of the simulation in PPS systems. How simulation and emulation can be used at the different levels and how can it work as a tool for evaluating production plans

During most simulation studies, a simulation model is developed, modelling the overall behaviour of the system, including planning and decision-making, control methods and the mirror of the physical system by modelling the resources. Most of the time this kind of simulation model (*simulation model* in Figure 2) is applied to test and validate production plans, to collect statistical data or simply to use optimization methods to find the optimal values of the selected system parameters. Simulation models are not so detailed, but have a very fast running-speed and wider time horizon (e.g. between 1 to 4 weeks). Thus it ensures a huge number of model runs, which gives a confidence by applying statistic-based parameter values,

or could have a role as an evaluation-function (*fitness function*) by searching the optimal value of a set of parameters [5].
In contrast to simulation, emulation only reflects the state of the underlying production system. Emulation (*emulation model* in Figure 2) is actually a simulation model without the control inside the model. This differs from the typical discrete event simulation, but the applied modelling techniques are the same [6]. Instead of validating production plans, emulation is applied for testing and evaluating control systems (level 2 in Figure 2).
In the following section, through a case study, it is described how the above methods can be applied.

4 Case study

4.1 Objectives

Our work is to find solutions for optimizing the internal logistic system of the selected section of a production system. The study is focusing on the control of the transportation unit (TRAM) and automated storage and retrieval system (ASRS). Therefore, a complex simulation study was carried out to be able to analyze and improve the control system.
The main steps of this study are as follows:

- create a simulation model of the selected section of the factory,
- locate the internal bottle-neck(s) in the material-handling processes,
- optimize the internal logistic system of the selected section of the plant.

4.2 The modelled system

The case study concerns a department of a factory, an open job-shop plant producing machine components. The production is organized around the ASRS that is the only temporary buffer in the system. A TRAM system is serving the ASRS and the workstations. Each container contains a various number of identical parts travelling together till the completion of their processing plans.
The machines are grouped in workstations, with a variable number of container docks and with different processing capacities. Typically, a workstation holds two containers: an empty container to be filled with the finished parts, and a full container with parts to be worked on.
Inside the workstation a part is taken by the human operator from the full container and loaded into the processing machines, processed and then unloaded and stored in the originally empty container. When this last container is full, the ASRS is prompted to take it away. Because the Tram has two containers docks, prior to pick up the finished containers it travels to the ASRS to bring the next container that is going to be processed in the requesting workstation. Therefore, once the Tram took the container with the finished parts, it unloads the next container without an additional movement.

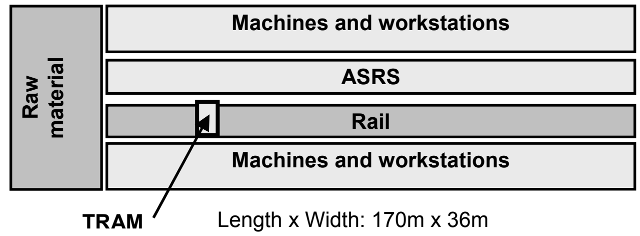

Figure 3 Topology of the inspected part of the plant

Finished pieces are stored in the ASRS and retrieved in a given number on a daily base, according to the assembly orders. The human operators are assigned to workstations and not to a single machine on the basis of their skills, shifts and preferences. Overall, the plant holds the characteristics of a classical open job-shop, with different alternatives to carry out a processing operation.

5 Model development

5.1 Selection of simulation tool

The first step is to choose the right simulation software tool for the given problem. The selected simulation package should have an open architecture and it should be easy to communicate with other software packages or real systems. The package should be able to deal with both standard communication protocols and user-defined communication protocols. Types of interfaces that can be used are for instance Dynamic Data Exchange (DDE), Dynamic Link Library (DLL), TCP/IP socket connections, ActiveX, OLE for Process Control (OPC), Distributed Components Object Model (DCOM). When needed the user should also be able to construct custom made interfaces.
Based on these criteria we have chosen *e*M-Plant 6.0 to implement the model of the inspected area.

5.2 Main steps of the development process

Because of the possible iterative steps between *creation of conceptual model, model implementation and testing*, three main steps have to be defined for the model (Figure 4). First, a classical simulation model was created focusing on the material handling system. After that the model was distributed to two separated model parts. As one of the results, a DDE connection has been established between these two models. By using this connection the developed string message based communication and control algorithms can be tested. The third step belongs to the further activities, i.e. to implement the improved control system and test it on the emulation. This will be the evaluation phase of the study (see combination 2 in Figure 1).

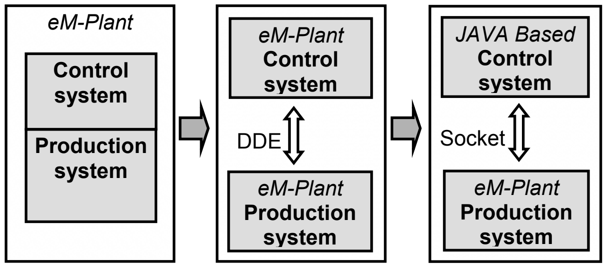

Figure 4: Main steps how to separate the controller and emulation and how to evaluate the control system

5.3 Preparing off-line simulation

In the first phase, both the emulation and controller parts were developed in *e*M-Plant, in the same simulation model, in order to be able to test the behaviour of the physical system (combination 4 in Figure 1). As described in [7], there are predefined *commands* and *events* available for the controller enabling the communication with the resources. Most of them are also implemented and used in our simulation model as string messages. By using eM-Plant, these messages can be handled asynchronously by applying *dynamic message lists*. The message processing component was built to be able to handle the incoming and outgoing messages parallel. This allows processes to send messages without waiting for an immediate answer. A process sends messages to the message processing component and continues to operate as normally, without having to wait for the other process. The message we use is a standardized string message:

<msg_ID|time|SensorID|sender|order|name|param1|param2|param3|param4|param5|param6|receiver>

At this state of the work, the only dispatching rule is to store the containers as near to the next process as possible. At the initializations phase all the slots of the ASRS are totally empty. Applying this setup the internal rack-serving algorithm of the ASRS was tested to discover the most frequent places in the store.
During the whole model development we focused on the message oriented communication approach between the controller and emulation parts, which made the separation of the controller from the classical simulation easier. When running controller and emulation in the same simulation environment it is easy to synchronise the two models, because the same *event controller* generates the events and both models have the same internal (simulation) clock. This behaviour could be applied very well in the development and testing phase.
At the end of this phase, the model (see Figure 5) had two main application-object libraries: emulation and control libraries. The emulation part is highly detailed, for example sensors on the track of the TRAM, or the acceleration of the TRAM are also implemented.

5.4 Synchronization of control and emulation

In the second phase, our communication interface was specified and implemented to be able to separate the model, which was divided to emulation and controller. The DDE communication was established between two *e*M-Plant licences running on two different computers. DDE is a widely applied protocol, it is supported on more hardware platforms, and it is often used in communication between hardware systems controlled with PLC-s and the outside computers. *e*M-Plant supports DDE communication allowing the user to establish a hot-link communication in a quite easy way. Despite of the several advantages of DDE, it also has disadvantages, namely in practice it has limitation in the number of the connected computers. This limitation depends on the structure of the messaging system.
The structure of the communication between emulation and the external controller is a standard string message based real-time communication, similar to the tested one in the first phase.

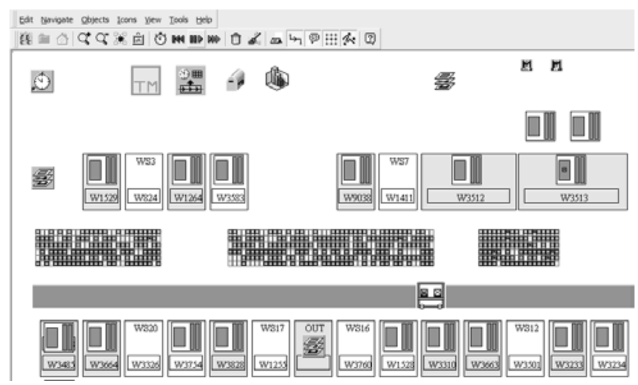

Figure 5: The emulated model of the production system in *e*M-Plant. The controller is a separated library in the model

As described in [4], synchronization is very important aspect in combing simulation, emulation and control. Two types of synchronization are distinguished; time and place ones. The synchronization of time is aimed at synchronizing the simulation clock of the simulated systems or to the internal clocks of the control system and emulated material handling systems. *e*M-Plant offers standard built-in features for real-time time progress in simulation models.
In our work we do not have to combine more simulation models controlled by more control systems yet, but the problem has also occurred while synchronizing the two models. The notification of events in the emulation will provide information to the controller. These *events* generating an answer in the controller, but these answers are not always sent immediately back to the emulation. The delayed *commands* are scheduled in the control system and if the internal clock gets to the defined time point, these commands will be released.
Our main objective is to make some suggestions how to improve the internal material-handling system (TRAM

and ASRS), so the other activities such as analyse and test different standard interfaces (e.g. CORBA or HLA) needed for the further research – to avoid the application of DDE and to be able to communicate to other, industrial applications – will be carried out in a short time as a part of the next phase.

6 Ideas on optimizing the TRAM and the ASRS

6.1 Some features of the currently used scheduler

In our work, we focused on the influence of the applied scheduling methods on the (emulated) production system. We have tested in the factory currently used scheduling methods as well as some new techniques too.
According to the current state at the factory the production plan is calculated without directly scheduling the TRAM. That means, the system calculates with a fixed transportation time and inserts this time into the plan after each operation on a machine. In this way the TRAM is not controlled by scheduled events, but it is operated by „direct" calls from the operators at the workplaces (the operators using their terminals to call the TRAM). However, this is an obvious and flexible solution in the current situation, but this kind of dispatching rule is responsible for the periodically occurring overbooking of the TRAM, as it is proved by the simulation. In the situation where the TRAM is called by several operators in a very short time, it becomes a bottle-neck in the system, because it is not able to carry out the needed transportations in time, moreover, the scheduled tasks for the workplaces will be delayed. There is no feed back information during the plan calculation to ensure if it is able to do the task, i.e. whether it is free at that moment.
Applying this control strategy, the simulation model represented the above described problems very well.

6.2 TRAM as a scheduled resource

If all of the resources in the system – workplaces, workers and also transportation resources – would be scheduled as simple resources, theoretically, it would be possible to calculate an optimal and feasible production plan.
In this case the TRAM also would have a schedule table which allows the control system to create a production plan without containing time periods with a higher transportations demand than the TRAM capacity. However, in the production plan there would be greater time-gaps between the operations on the machines – because the system would calculate with the data of the real schedule-table of the TRAM, instead of prefixed operation times. Consequently, this plan would be near to the optimal solution.
In the practical point of view, it is not always easy to realize such a strictly scheduled system, in fact, if there is a significant difference between the operation time of a machine and the TRAM. Important if both machine and TRAM are handled as the same kind of resources, that a normal process operation takes most of the time hours, on the other hand, the most extended and complex task for the TRAM takes only ten percent of it. „Sensibility" in the schedule of the transportation resource could query the effectiveness and fault-tolerance of a prescheduled production plan in such an environment comparing it to a distributed controller system [7]. This pre-scheduled system is lack of robustness and for this reason it had to be rescheduled several times in one shift to handle uncalculated disturbances occurring in the system.
Anyway, to schedule the TRAM and to create the plan for the simulation, some basic transportation times needed. So a possible way is, to let the control system calculate with the highest possible transportation times at the beginning, but as soon as it gets statistical data from the simulation runs it will modify the time for the given relation. The results of this *iterative* solution are written in 6.4.

6.3 Applying a more flexible schedule

As further activities we are working on another scheduling procedure for the TRAM. It is obvious to define not only exact time-points but time-windows (TW) during the TRAM schedule. The controller calculates the shortest possible operation time for the transportation, and also takes the fact that the TRAM has to reach the starting position of the operation in time into account (this is a kind of setup time). It requires the controller to always know where the TRAM is and what is its destination. For calculating operation times, permanently updated statistic data are needed, which comes from the emulation. It would be implemented so, that the TRAM would get only the TW (within the task has to be finished), the calculated duration of the operation and the two destinations (the two endpoints of the transportation task) from the scheduler. When collecting several TW-s it could allow the internal schedule-algorithm of the TRAM controller to optimize the current scheduled tasks with an objective function focused on finding the optimal i.e. the shortest possible routing for the resource. This also means the minimization of the total operation time.
Actually, the optimizations method could replace and sort the scheduled tasks if they can be replaced without „leaving" their own time-window. A very important thing has to be declared while applying this flexible scheduling system. We assume that the TRAM scheduler can create at least one feasible solution using the TW-s passed from the main controller-scheduler.

6.4 Finding an optimal place in the store

As it is described above, most workplaces do not have a buffer capacity for containers more than two slots. Between operations, temporary storage of the containers is needed, so the most important questions are *when* and *where* to store the containers in the ASRS. The answer to the *when* question depends on the schedule of the resources as discussed above. As other main possibility improving the control logic of the internal logistic system is to optimize the utilization of the store.

Comparing to common high-raise warehouses in this system, there are no exact in- and output points defined for the material flow in the store. For this reason the classical zone-strategy – applied in most warehouses – had to be modified.

The main idea to improve the utilisations level of the ASRS is to collect information about the resource, by monitoring the store-in (or store-out) operations. This was implemented by using data tables – at the *controller* side – representing the slots of the ASRS, where several data (e.g. number of the store-in operations at one defined slot, or product-types) are collected. This gives enough information to build *utilisations maps* of the storage system (see **Figure 6**).

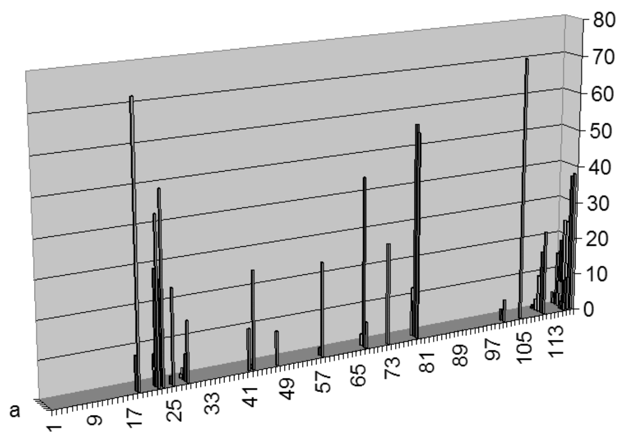

Figure 6: Frequently used slots in the store by applying the "next-to-workplace" strategy (200 orders and empty store at the initialization)

In order to discover product-specific zones, first the rack-serving-algorithm had to be defined. In the *control* model the ASRS selects that empty slot that is nearest to the workplace, where the next process will be carried out (*next-to-workplace* strategy). Three other slot selection strategies were tested: *half-way*, *1/3-way* and *2/3way* to the next resource (**Figure 7**).

No. of Orders	Next-to	Half-way	1/3-way	2/3-way
50	34675	34121	34365	34103
100	69177	67853	68653	67697
200	141885	139813	142023	139281

Figure 7: Distance in meters, ran by the TRAM. (Average value from ten simulation runs for each parameter setting.)

To make the store adaptive to changes, a monitoring system was developed and tested in the *controller*. If the number of the store-in operations exceeds a predefined value at a defined slot (e.g. dynamic priority for each slot), then this slot will be inspected. Inspection is a special operation for filtering the inactive orders in frequently used slots by moving the container to a slot with a lower priority.

According to the results of the simulation study, by using these methods the average time needed for a transportation operation has been reduced.

7 Summary

The study showed that, the flexibility and maintainability of the resulting combination of the developed control system and emulation model is exceptionally advantageous. Several experiments have been carried out, and the results show that designing new control systems or testing existing ones through distributed, interactive, object oriented simulation provides unique designing and testing features.

8 Acknowledgements

This work was partially supported by the National Research Foundation, Hungary, Grant No. T034632 and the 5th Framework GROWTH Project MPA (G1RD-CT2000-00298).

9 References

[1] T. Gogg, J. Mott, Introduction to Simulation, *Proceedings of the 1993 Winter Simulation Conference*, 1993, pp. 9-17

[2] A. Seila, Introduction to Simulation, *Proceedings of the 1995 Winter Simulation Conference*, 1993, pp. 7-15

[3] A. M. Law, W. D. Kelton, *Simulation modelling and analysis* (McGraw-Hill, 2000), pp. 669-672

[4] C. Versteegt, A. Verbraeck, The extended use of simulation in evaluating real-time control systems of AGVs and automated material handling systems, *Proceedings of the 2002 Winter Simulation Conference*, 2002, pp. 1659-1666

[5] Monostori, L.; Váncza, J.; Márkus, A.; Kádár, B.; Viharos Zs.J.: Towards the Realization of Digital Enterprises, *36th CIRP International Seminar on Manufacturing Systems,* June 03 – 05, 2003, Saarland University, Saarbrücken, Germany (paper accepted for publication).

[6] I. Mc Gregor, The relationship between simulation and emulation, *Proceedings of the 2002 Winter Simulation Conference*, 2002, pp. 1683-1688

[7] P. Valckenaers, M. Kollingbaum, H. Van Brussel, O. Bochmann, C. Zamfirescu, The design of multi-agent coordination and control systems using stigmergy, *Proceedings of the IWES'01 Conference*; Bled, Slovenia; March 12th - 13th, 2001.