Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

KNOWLEDGE-BASED EVENT CONTROL FOR FLOW-SHOPS USING SIMULATION AND RULES

Mark Aufenanger
Wilhelm Dangelmaier
Christoph Laroque
Nando Rüngener

Heinz Nixdorf Institute
Fuerstenallee 11, University of Paderborn
33102 Paderborn, GERMANY

ABSTRACT

The requirements on production systems and their planning and control systems are constantly growing. Systems have to be flexible and provide viable solutions at the same time. Different planning and control approaches, such as optimization, simulation and combination of techniques etc., that attempt to solve the scheduling problems are available. Mathematical solutions which can be found in literature didn't solve the real-world problems in an appropriate way. Current knowledge based solutions did not give any value about decision reliability as well as their decision attributes are not differentiate enough. We are developing a new rule based approach by using a combination of simulation and a knowledge generation within a dynamic production planning and -control for flow-shops. Ideas of how knowledge can be trained by simulation are presented. Furthermore which kind of rules and attributes can be used and how decisions about the rule selection can be made are shown.

1 INTORDUCTION

Today, production systems are confronted with many types of uncertainty. Internal uncertainties like job processing times, machine breakdowns, etc. and external uncertainties as urgent jobs, unknown new jobs, and so on (cf. (Aytug et al. 2005) and (Schneeweiß 2002)). Furthermore manufacturing systems are getting more and more complex by themselves. There are no longer only a few different specifications of a product, there are hundreds of different specifications which can be chosen by the customers. The requirements on production systems and their planning and control mechanisms are constantly growing. To be competitive, these two problem areas (uncertainty and complexity) have to be handled in an adequate way. Therefore manufacturing systems have to become flexible to handle the complexity on the one hand. On the other hand planning and control techniques / methods are necessary to manage the flexible system efficiently are more and more essential. Apparently (dynamic) production scheduling and re-scheduling is one area, which is very critical for business success. By using an adequate schedule / scheduling dynamically (depends on system type), a company can reduce cost and gain flexibility at the same time. Hence the key to success is to become flexible and cost efficient at the same time.

At the first sight optimization models should be used for the purpose of generating or regenerating schedules. But they are not capable of this aim because of the real-time demand and the problem complexity. Hopp and Spearman (2007) emphasizes that most real-world problems violate the assumptions made in the classic scheduling theory literature. Because of the NP-hard character of the real-world problems, they are simplified. Optimization problems for example are able to generate efficient / optimal solutions but only for small problems (15 jobs and 10 machines) and not in real-time. Heuristics are able to generate good – not necessary optimal – solutions as well, but for real-world problems they also take an expensive amount of time to solve the problem. So they cannot be used as real-time control systems too (cf. (Holthöfer 2000)). Furthermore optimization- and heuristic-systems are generating solutions for a given time horizon, they did not take unavailable material or something else into calculation as well as the system development during the processing time. Günther and Tempelmeier (2005) suggest, that optimization models, which are able to map problems in an adequate way, gather only a low acceptance for practical experience. Mostly, quick heuristic approaches working with dispatching rules are used. The rule and attributes are not detailed enough to handle Situations. For example Situation 1 and Situation 2 like the same situation for the control system because of the missing attribute machine utilization. We think that an intelligent use of dispatching rules -they can be used locally and generate quick solutions- to generate efficient solutions could be very promising. In our opinion most times it is better to use a good solution instead of waiting too much time for the optimal one.

In this paper we present the concept of a reactive (re)scheduling approach for a flow-shop system, for incoming jobs by the use of different dispatching rules as well as differentiated attributes. Since no rule dominates the others, we are selecting rules dynamically, depending on the current status of the production system (machines and jobs). Selecting the rules is done by a knowledge based method. The training examples will be generated by the use of simulation and the knowledge will be generated by the use of a classification system. The simulation system (d³FACT insight) for generating training examples system we are developing since the last three .

The remainder of this paper is structured as follows. In the Chapter 2 we present the state of the art in the area of selecting dispatching rules in combination with artificial intelligence approaches. In Chapter 3 we show the researched flow-shop mathematically and in chapter 4 we show the assumptions made for our approach. We presents the concept of our approach in chapter 5. Chapter 6 provides the conclusion and a forecast of work that has to be done next.

2 STATE OF THE ART

Wu and Wysk (1989) are among the earliest to describe the problem of dispatching rule selection within a flexible manufacturing system. They present a scheduling algorithm which employs discrete simulation in combination with straightforward part dispatching rules in a dynamic fashion. The approach divides the time horizon into shorter intervals and at the beginning of each interval a variety of rules are simulated. Afterwards the best performing rule is selected for the next time period. Wu and Wysk point out, that within their approach, decisions are made locally and may not lead to global system performance. Their approach is lacking a few things. They did not select rules really dynamically a every system status change (they are using intervals) and they did not create good global solutions.

A few years later Piramuthu et al. (1993) have presented a framework for incorporating machine learning capabilities in intelligent scheduling. They developed pattern-directed method for heuristic acquisition and refinement. Thus a characterization of how different dispatching rules perform under different operating conditions by using a simulation model of the investigated manufacturing system can be made. Furthermore they apply a learning algorithm on the generated data to develop a decision tree. The generated tree is used to select a dispatching rule whenever a significant chance in system state is identified. But by the use of decision tree, Piramuthu et al. are not able to show how sure a selected solution is and there are problem if attributes are unavailable when generating a decision. Furthermore a detailed system status differentiation is not given.

Manivannan and Banks (1991) present a simulation and knowledge-based Real-Time Control (RTC) system for flexible production systems. They point out that a RTC system has to be capable of the following 5 points:

1. Reacting to the problem instantaneously
2. Evaluating several alternatives policies
3. Providing optimum or near optimal solutions
4. Learning from previous problems
5. Providing faster and more accurate solutions

Manivannan and Banks present a framework for accomplishing RTC. The aim of the system is to provide an integrated environment for the controller to evaluate various control policies using simulation. Data for the simulation is collected from the manufacturing cell The main issue of their approach is the synchronization of the events between the simulation system and the real-system. They describe the need for this event synchronization of RTC and an approach to handle this problem. A temporal knowledge base has been designed to synchronize the events and their time of occurrence in both the manufacturing cell and the simulation model. Furthermore a dynamic knowledge base has been implemented to store the results of previous on-line simulations results. Manivannan and Banks claim, that the dynamic knowledge base provides a response to a control problem in a short time period but by the use of “IF … THEN … ELSE …” rule selection and directly inserted machine numbers, their approach cannot be used for a system consisting of more than one machine.

Park et al. (1997) developed an adaptive scheduling policy for dynamic manufacturing systems: pattern-directed scheduling (PDS). They use a inductive learning method based on 6 parameters to select the appropriate dispatching rule in decision points. The rule selection method is imbedded in a decision tree which is generated by applying an inductive learning algorithm on the training example generated by the example generator. Park et al. show the superiority of the suggested PDS approach against the repeated application of a single dispatching rule by experimental studies dealing with a FMS. By the use of 6 parameters, different system situations could not be recognized deep enough and there was no solution presented for dealing with situations with not an suitable rule.

Harmonosky (1995) reviews different approaches and developments using simulation, with its look-ahead and what-if capabilities, as a tool in real-time scheduling. As further development Harmonosky et al. (1997) are presenting an approach for making a real-time selective re-routing decision. Making the re-routing decision is done by using steady-state system performance estimates from simulation models run a priori to any system disturbance. Their approach is to create a relatively simple tool based on simulated long-term system performance data. They make an evaluation of their system, based upon system performance measures of average flow time and average throughput. Furthermore they point out, that rerouting all jobs in the case of a disruption / failure leads to global system performance measures that are worse, than doing nothing. Harmonosky et al. not able to show how sure their system is about the respectively selective rerouting decision.

O'Kane (2000) presents a knowledge based application which is used to research reactive planning scenarios in a specific flexible manufacturing system. Concepts of "History logging" and expert system "learning" are proposed and adapted to provide decision making and control across a FMS schedule lifetime. O'Kane supposed that his approach represents a first step in the development of intelligent simulation systems for the analysis of the reactive scheduling problem in FMS. Amongst others, their approach did not give look-ahead solutions when making a decision.

Kwak and Yih (2004) present a decision tree based approach for the control of a testing and rework cell with parallel stations and a central input and output, with in a dynamic computer-integrated manufacturing system. Their competitive decision selector (CDS) observes the status of the system and the jobs. It makes its decision on job preemption and dispatching rules in real-time. The CDS consists of two different algorithms and combines the knowledge about the long-run performance and the short-term performance (by the use of data mining methods applied on simulation based generated training data) of each rule on the various status of the system. Their experimental results show, that the CDS dynamic control is better than other common control rules with respect to the number of tardy jobs. Using a decision tree leads to the problem of selecting a rule without giving a value of how sure the current selection is as well as there are problems if attributes are missing during the rule selection.

Wang (2007) characterizes today's possibilities of collecting and storing all kinds of data in manufacturing enterprises. He points out, that traditional data analysis methods are no longer the best alternative to be used. Wang identifies four reasons for the few research interest of data-mining-techniques in the manufacturing domain:

1. *The majority or researchers in the manufacturing domain are not familiar with DM algorithms and tools.*
2. *The majority of theoretical DM researchers are not familiar with the complex manufacturing domain area.*
3. *The few researchers who are skilled in both DM algorithms and manufacturing domain are not able to access, often proprietary and sensitive, manufacturing enterprise data.*
4. *It is difficult to evaluate the effectiveness and benefits while DM is implemented in manufacturing.*

His paper focuses on data mining definitions, techniques and procedures, which can be used to discover and extract knowledge form manufacturing data bases. He provides some practical example, e.g. using a decision tree in combination with data mining technology. The use of decision tree lead to the some problems presented within the approaches above.

Halevi and Wang (2007) present a knowledge based manufacturing system (KBMS). They emphasize the complexity within batch type manufacturing environments. Halevi and Wang are defining a system approach where management performance relies on decisions made at a too early stage in the manufacturing process. They point out that today decisions are made by engineers who are neither economists nor prediction planner`s experts. Within their approach it is no longer the engineer`s task to make decisions but rather to prepare a knowledge-based "road map". By using the road map, each user can generate a routine that meets his needs at the time of needs by using KBMS CAAP. The approach presented by Halevi and Wang is not able to make real system status dependent decisions and furthermore they did not take a knowledge refinement into account.

Mahl and Krikler (2007) describe a software application for capturing and re-using rule based knowledge concerning manufacturing machine services. The system is designed to support different kinds of manufacturing machines and manufacturing machine specific domains. Therefore they present a machine specific ontology.

The presented approaches showed, that combining data mining and simulation technologies is a good approach for doing good real-time / event based decision in complex and dynamic systems. But nevertheless the presented work lacks in some ways. Attributes and rules are not diversified enough for doing differentiated decisions. Furthermore when using a decision tree, it is hard to maintenance the tree if attributes were added or deleted. Also a decision tree has some problems with unavailable attributes when generating a decision as well as the rule selection methods did not give hints about the decision assurance. It is impossible to say if the system is sure in selecting the rule. Furthermore there are no ideal look-ahead functions to manage unsure situations.

3 RESEARCHED MODEL

Consecutively, we descried the flow-shop we are currently working with (This is the first step, our control system will be extended to be used within a flexible manufacturing system). All the jobs are running through the system in the same order, there are no machine setup times (they are an implicit a part of the job machining times), at one time there can only be one job at a machine, at every time there can only be one job operation be handled, non-preemptive operations, job machining times are given but unsure.

The system description formally:

- n = number of jobs
- s = number of stages
- j = number of one job ($1 \le j \le n$)
- i = number of one stage ($1 \le i \le s$)
- N = number of all jobs
- $p_{j,i}$ = machining time of job j at stage i, p element
- r_j = release time job j at the first stage
- $rt_{j,i}$ = remaining machining time of job j from stage i

Our approached aim is a makespan minimization, so this is used as the performance measure. According to this, the problem can be written shortly by the use of the α|β|γ notation (cf. (Graham et al. 1979)) as following:

$$F||C_{max}$$

4 ASSUMPTIONS

We think that there is an ability to learn from good past schedules before the active phase of a production system and to use this data for doing (re)scheduling decision event based (real-time) during the systems running phase. For doing so an automatically generation of learning data during the initial schedule creation it is necessary (No knowledge of experts should necessary, the system can be human independent but haven't too.). Furthermore a definition of system wide and diversified attributes and rules for doing differentiated decisions are necessary. Any more global attribute values are available for local decision points

The use of pre-generated knowledge leads on the one hand to better routing decisions during the active control phase (intelligent differentiated rule selection) and on the other hand decisions can be made event based and in real-time. No mechanism for ensuring decisions has to be used during the active system phase for sure decisions but by the use of simulation look-ahead unsure decisions can be tested.

5 FRAMEWORK

The following chapter presents the concept of our controlling system approach consisting of two system phases: Training- and control-phase (dynamic (re)scheduling). During the first phase attributes and rules are evaluated and the training examples for the knowledge based rule selection are generated.

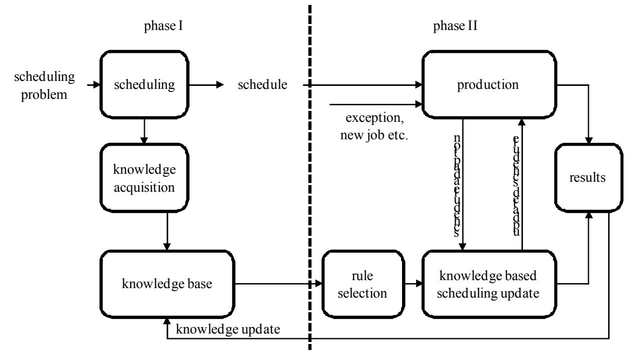

Figure 1: Entire system with components

Afterwards this knowledge is used during system phase two for doing dynamic (re)scheduling by selecting the rules system status dependent. Figure 1 shows a schematic system overview. Both phases will be explained in detail during the next two chapters.

5.1 Testing Attributes and Rules – Generating Training Examples

The system training phase will be done before operative work of the production system. Training will take some time, but this time will be saved during the active phase because of the ability to generate event based real-time (re)scheduling decisions.

There are some possible training methods for generating data like optimization, heuristics, or simulation for generating the training data. Within our approach we want to use simulation for the knowledge based system training. Primarily it is very applicable because of the system dynamic (e.g. unknown job release times as well as unsure job machining times). Secondly the simulation model / system can be used for a quick look ahead if there is a situation where the knowledge based rule selection cannot generate a feasible solution. A simulative based look-ahead can be taken (this takes some time) for generating a decision.

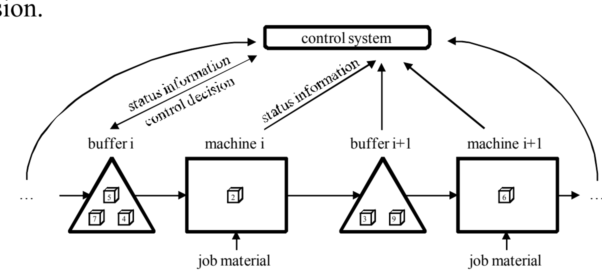

Figure 2: Flow-Shop decision scheme

We are differing of two types of attributes within our approach. The one presenting the local information and the one presenting global information. The local ones are belonging only to the decision points stage and the global ones holding information system wide. Local ones are essential for generating a solution, global attributes are used to ensure good system wide results in time (for example there could be urgent jobs a few stages backwards the decision stage which influence the current decision) as well as machinery (for example necessary material for a job is missing at the next stage). A short view of attributes we want start within our approach firstly to test the system could be for example:

- Job machining time
- Machine utilization whole system and next two stages
- Free buffer capacity next two stages
- Relative progress of all machining time of planned operations
- Relative progress of machining time of planned operations for job
- Relative progress of the number of all planned jobs
- Relative progress of the number of planned operations for job

Naturally the amount of attributes have to be expanded for describing differentiated system situations. Otherwise the best fitting rules cannot be selected. Therefore simulation based attribute test as well as foreman interviews can be taken.

Next the our function of combining rules and attributes is explained. For example if there are 3 possible jobs (j_2, j_3, and j_7), the simulation system selects job 3 as the best one for gathering the systems' aim (minimization of makespan). Afterwards the systems firstly gathers all current attribute values and secondly the simulation solution has to be mapped to a dispatching rule (we start by the use of standard dispatching rules e.g. SPT, LPT, EDD, LSO, MWR) selecting the same job. Thereafter the mapping system test which of the rules selects job 3 as well until all rules are processed. By doing so it could be possible that there are more than one appropriate dispatching rule for the situation which is not a problem. After mapping situation and rule, there is a new training example consisting of the attribute values and the dispatching rule(s). For that case, that no one of the rules can select the job there could be no training example generated, the job selecting rule could be randomized, or a simulation based look-ahead has to be taken during the systems active phase.

Because of these factors, the production system and its attributes are changing continuously. So it seems to be normal to use an approach, that dynamically selects rules, depending on the current system status and point out the selection reliability. To do so, we need a method that can select the best scheduling rule for the current situation (cf. (Piramuthu et al. 1994)). Our rule selection mechanism will be presented in the following chapter.

5.2 Material Flow Control – Dynamic Rule Selection

The second system phase is called the (re)scheduling / control phase and runs while the production system is operative. Generally our approach can work as a scheduling system, integrated within a quick heuristic doing critical decisions or as a reactive scheduling system by dynamically routing the jobs through the production system.

When putting jobs onto machines, it has to be saved that there will be no future problem, e.g. all parts for the selected job are available at the next stage. To avoid such situations, global attributes are used within our approach. For example there are job material attributes for the next stages and if material is not available, then the job will not be selected. Furthermore if there are situations where no good decision can be made, the system will be able to manage that situation (see chapter 5.1). The simulation used for generating training examples is an appropriate solution here.

In general three common decisions have to be taken within a scheduling system: Selecting a job out of a queue, selecting a machine for one job, or selecting a job out of a queue and a machine for the selected job. Because this paper works with the flow-shop (Figure 2), we primarily do the selection of a job out of a queue decision. After the final implementation and testing, we extend the system for managing flexible manufacturing systems (FSMP), so all three decisions have to be taken by our approach in the future.

An efficient rule selection tool is necessary to generate good decisions. Candidates are Naïve Bayes Classifier, Decision Trees, Baysen Networks, and Clustering Methods (Detailed information's about the different approaches can be found in Mitchell (2007) for example.). We have selected the Naïve Bayes Classifier because it can process thousands of attributes within a short time, maps the problem best, and writes out the reliability of the generated decision. It works by the use of statistical classifications with knowledge about the apriori-probability of solutions $P(L)$ and conditional probability $P(M/L)$ of attributes under assumption of solutions. The probabilities are calculated from a representative case base. On this probability base, the solution probability $P(L/M_1 \ldots M_n)$ is calculated by the use of Bayes (cf. (Görz et al. 2003)) Against the naïve assumption, Naïve Bayes Classifiers also work well with dependent attributes (cf. (Friedmann 1997)). If the classification system is not trained well, the whole control system is not able to generate good results. Because of this, methods for avoiding an overfitting of the rule selection method has to used. There are some standard methods in the literature which can be used.

Because there can be situations where the rule selection system is not be able to select an appropriate rule solutions for managing such situations have to implemented. Possible solutions could be the exclusion of inapplicable rules an making a random decision between the others. Because this will normally not lead to good results a simulation based look ahead within defined borders is the most promising method for us. By doing so, the simulation will take current production system status and make a short look-ahead for the next two stages.

Knowledge refinement works indirectly the whole time. When a decision is made, the result will be put within the training example base and when reaching the define point (for example every night), the classification mechanism will be trained again (Alike an incremental learning mechanism.).

6 CONCLUSION

In this paper we highlighted complexity and uncertainty within today's production systems and pointed out the problems of current control mechanisms. Afterwards we presented the concept of our approach of a dynamic production control mechanism by the combination of simulation and knowledge based dispatching rule selection. Foremost we point out that more and better (differentiated) attributes are necessary to specify an increased number of system situations. Only by doing so, appropriate decision can be done. Secondly we suggest that decision have to be done event based (in real-time), therefore we present the concept of classification rule selection. As well as we pointed out the advantages of using a classification system instead of a decisions trees. In the next development steps we will finish the system implementation and do performance and result tests (for the flow-shop environment). The first few results suggest, the knowledge based control system will become applicable. Following the performance test, the system will be extended to (re)schedule flexible FSMP production systems.

REFERENCES

Aytug, H., M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research* 161:86–110.

Friedman, N., D. Geiger, and M. Goldszmidt. 1997. Bayesian network classifiers. *Machine Learning* 29:131-163.

Görz, G., C.-R. Rollinger, and J. Schneeberger, (eds.). 2003. *Handbuch der künstlichen Intelligenz*. 4 th ed. Oldenbourg: Wissenschaftsverlag.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals Discrete Mathematics* Bd. 5:287–326.

Günther, H., and H. Tempelmeier. 2005. *Produktion und Logistik.* 6th ed. Berlin: Springer.

Halevi, G., and K. Wang. 2007. Knowledge based manufacturing system (KBMS). *Journal of Intelligent Manufacturing* 18:467–474.

Harmonosky, C. M. 1995. Simulation-based real-time scheduling: review of recent developments. In *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, 220–225. Arlington, Virginia, United States: IEEE Computer Society.

Harmonosky, C. M., R. H. Farr, and M. Ni. 1997. Selective rerouting using simulated steady state system data. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 1293–1298. Atlanta, Georgia, United States: IEEE Computer Society.

Holthöfer, N. 2000. *Regeln in einer Mengenplanung unter Ausbringungsgrenzen*. Heinz Nixdorf Institut, Universität Paderborn, 66.

Hopp, W. J., and M. L. Spearman. 2007. *Factory physics: foundations of manufacturing management.* 3rd ed. Boston: Irwin/McGraw-Hill.

Kwak, C. and Y. Yih. 2004. Data-mining approach to production control in the computer-integrated testing cell. *IEEE Transactions on Robotics and Automation* 20:107–116.

Mahl, A. and R. Krikler. 2007. Approach for a rule based system for capturing and usage of knowledge in the manufacturing industry. *Journal of Intelligent Manufacturing* 18:519–526.

Manivannan, S., and J. Banks. 1991. Real-Time Control of a manufacturing cell using knowledge-based simulation. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B. L. Nelson, W. D. Kelton, and G. M. Clark, 251–260. Phoenix, Arizona, United States: IEEE Computer Society.

Mitchell, T. M. 1997. *Machine learning*. New York: McGraw-Hill.

O'Kane, J. F. 2000. A knowledge-based system for reactive scheduling decision-making in FMS. *Journal of Intelligent Manufacturing* 11:461–474.

Park, S. C., N. Raman, and M. J. Shaw. 1997. Adaptive scheduling in dynamic flexible manufacturing systems: a dynamic rule selection approach. *IEEE Transactions on Robotics and Automation* 13:486–502.

Pawellek, G. 2007. *Produktionslogistik. Planung – Steuerung – Controlling*. München: Hanser.

Piramuthu, S., N. Raman, and M. J. Shaw. 1994. Learning-Based Scheduling in a Flexible Manufacturing Flow Line. *IEEE Transactions on Engineering Management* 41:172–182.

Piramuthu, S., N. Raman, M. J. Shaw, and S. C. Park. 1993. Integration of simulation modeling and inductive learning in an adaptive decision support system. *Decision Support Systems* 9:127–142.

Schneeweiß, C. 2002. *Einführung in die Produktionswirtschaft*. 8th ed. Berlin: Springer.

Wang, K. 2007. Applying data mining to manufacturing the nature and implications. *Journal of Intelligent Manufacturing* 18:487–495.

Wu, S. D., and R. A. Wysk. 1989. An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing. *International Journal of Production Research* 27:1603–1623.

AUTHOR BIOGRAPHIES

MARK AUFENANGER studied Business Computing, at the University of Paderborn. Since 2005, he is a research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM. He is mainly interested in simulation of logistic systems, artificial intelligence and knowledge based systems. His email address is <Mark.Aufenanger@hni.upb.de>.

WILHELM DANGELMAIER studied Mechanical Engineering at the University of Stuttgart, Germany. Since 1981, he was director and head of the Department for Corporate Planning and Control at the Fraunhofer Institute for Manufacturing. There he was responsible for the development of SIMPLE++ (today eM-Plant owned by UGS\Siemens). In 1991, Dr. Dangelmaier became Professor for Business Computing at the Heinz Nixdorf Institute; University of Paderborn, Germany. In 1996, he founded the Fraunhofer Center for Applied Logistics. His principal interests today are models and tools for distributed production systems. His email address is <Wilhelm.Dangelmaier@hni.upb.de>.

CHRISTOPH LAROQUE studied business computing at the University of Paderborn, Germany. Since 2003 he has been a Ph.D. student at the graduate school of dynamic intelligent systems and, in 2007, received his Ph.D for his work on multi-user simulation. Since then he is team leader of the “simulation & digital factory” at the chair of Business Computing, esp. CIM. He is mainly interested in material flow simulation models and the “digital factory”. His e-mail address is <Christoph.Laroque@hni.upb.de>.

NANDO RUENGENER studied Business Computing, at the University of Paderborn. Since 2006, he is a research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM. His email address is <Nando.Ruengener@hni.upb.de>.