

A SIMULATION TOOL FOR COMPLEX ASSEMBLY LINES WITH MULTI-SKILLED RESOURCES

Evangelos Angelidis
Daniel Bohn
Oliver Rose

Department of Computer Science
University of the Bundeswehr Munich
Neubiberg, 81547, GERMANY

ABSTRACT

The focus areas of our research are simulation and optimization of complex assembly lines for heavy machinery (airplanes, turbines, industrial machines etc.). These production facilities have several specific characteristics: many isolated project networks with precedence constraints and thousands of multi-mode activities, time-bounds for activities and projects, many priority rules, limited numbers of multi-skilled resources with individual shift regimes, internal and subcontracted personnel, and resource locking rules. Formally, it is defined as a Multi-Mode Resource-Constrained Multi-Project Scheduling Problem with activity splitting. A promising way of dealing with problems in this domain is simulation-based optimization. In this paper, we introduce a specific custom-built simulator designed for this problem domain. The tool supports a variety of real-world extensions and dedicated behavior which usually comes at enormous runtime and development cost when it has to be built into a commercial off-the-shelf simulation tool.

1 INTRODUCTION

This paper presents a specific custom-built simulator designed to support solution approaches for scheduling problems in complex assembly lines found in industrial environments. Taking into consideration the ever increasing global market demands for reduced production costs and on time delivery, especially in assembly lines with workforce planning. This research focuses on small series or even unique items such as turbines, planes or industrial machines. The problem description contains various products with different production plans, precedence constraints and thousands of activities. Many elements, such as activities and products, have time bound requirements. The activities do not have a fixed processing time but many possible modes, in which their duration is closely linked to the amount and skills of the resources assigned to them. In addition, the manufacture has limited multi-skilled resources with individual scheduling and finite buffers. The scheduling of this type of assembly production lines is a very complex task.

It is defined as a large combinatorial order problem and is often mentioned in the literature as a Multi-Mode Resource-Constrained Multi-Project Scheduling Problem (MMRCMPSP) with activity splitting (Buddhakulsomsiri and Kim 2006). Unfortunately this is NP-hard and real-life problems of this size cannot be solved with classic approaches in short runtimes (Pappert, Angelidis, and Rose 2010). The research in this area is mainly based on small or medium sized problems, which are often solved by exact approaches. These strategies are not suitable for our large real-life problems, especially with respect to short runtimes. In Pappert, Angelidis, and Rose (2010) is a detailed explanation of this scheduling domain problems and its plausible solution strategies.

An alternative way of finding efficient solutions for real-life problems in short run time is the Simulation-based Optimization approach (Shapiro 1996; Law and McComas 2002; April, Glover, Kelly, and Laguna

2003; Pappert, Angelidis, and Rose 2010). The basic idea of this approach is to simulate a scenario, analyze it and create a new one based on the outcomes of the simulated scenarios. The goal of these iterations is to generate a feasible solution within only a few steps. We created a Simulation-based Optimization Platform (SBOP) (Angelidis, Naumann, and Rose 2012; Angelidis, Bohn, and Rose 2012) which aims to offer the possibility to create various solution strategies for different assembly line scheduling problems. This platform is designed to support most simulation tools as long as it is possible to write a model generator and control them from outside.

In this paper, we introduce a specific custom-built simulator designed for this problem domain. The tool supports a variety of real-world extensions and dedicated behavior which usually comes at enormous runtime and development cost when it has to be built into a commercial off-the-shelf (COTS) simulation tool. The simulation tool is a further development of the prototype tool, presented in Angelidis, Pappert, and Rose (2011).

The paper is organized as follows: In Section 2 we will extend the discussion on the Simulation-based Optimization Platform. Section 3 is devoted to the simulation tool, the inner model and provides an overview of a tool extensions. In the last Section, summary is drawn and future research is discussed.

2 SIMULATION-BASED OPTIMIZATION PLATFORM FOR COMPLEX ASSEMBLY LINES

In 2010 a framework for Simulation-based Optimization of assembly lines was presented in Pappert, Angelidis, and Rose (2010). That framework aimed to offer the possibility to create various solution strategies for problems in this area while reducing complexity and development time. In SBOP the previous concept was adjusted by making broader applications and as a result created a new tool (Figure 1). The new platform is based on OSGi (OSGi Alliance 2012) as a runtime environment. OSGi was selected because it offers many specifications, associated implementations and it is state of the art in software development. The platform components can communicate per event (sync or out of sync) or via direct service call between each other. The existing remote interface specification is also crucial to this research due to the fact that it allows the extension of this platform with remote interface in short time and with minimal effort. The platform has a flexible and easy modifiable meta-data model for working with several companies of different size and assembly approaches. It uses for persistence a scalable, high-performance database, the mongoDB (10gen 2013). Web technology and cloud services provide a user-friendly interface for uploading factory plans, triggering optimization runs with different parameters and visualizing the optimization results.

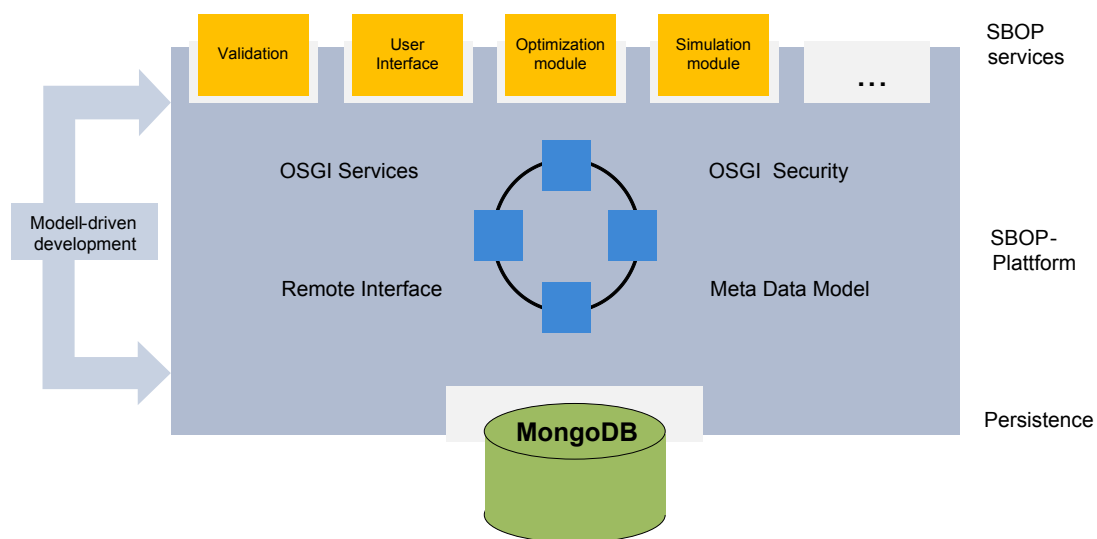


Figure 1: Simulation-based Optimization Platform based on OSGi

The general optimization mechanism in SBOP is a cycle of the following steps (illustration in Figure 2):

- creation of a scenario based on initial setups or results of previous optimization steps,
- simulation of this scenario,
- data preparation of the simulation results and
- evaluation of simulation results.

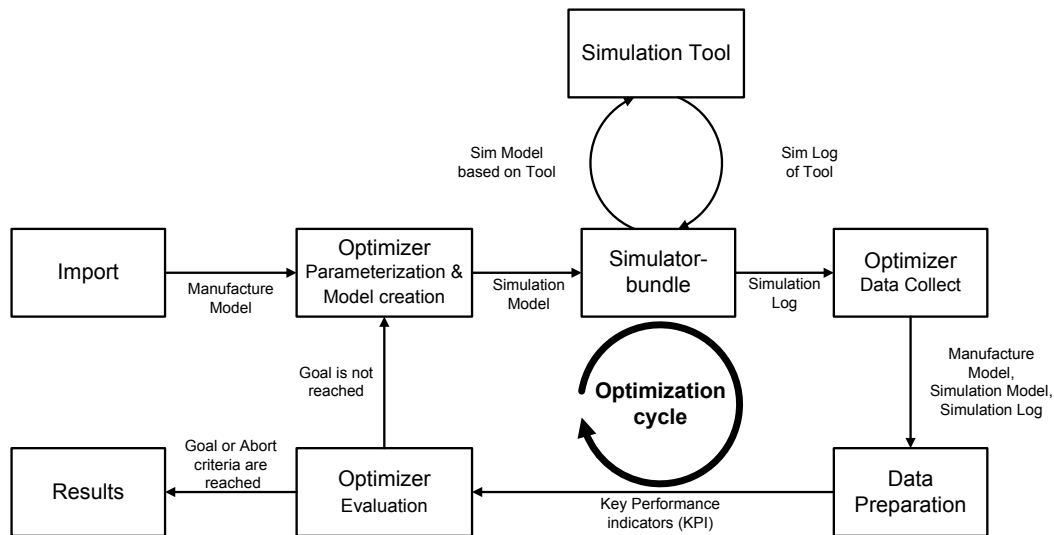


Figure 2: General cycle of Simulation-based Optimization.

This cycle is terminated when the goal or an abort criteria is reached. As a central element within SBOP, the optimizer module is responsible for the creation of scenarios (which we want to simulate) and the evaluation of the simulation results. The structure of the optimizer module itself is divided into three almost independent sub modules. This segmentation allows easier maintainability, exchangeability and upgradeability.

First part of the optimizer is the parameterization, which includes the actual optimization algorithms. In the parameterization the attributes for the upcoming scenario are determined. These attributes are for example:

- activities' mode,
- earliest start and latest end dates of activities, production plans and customers' orders,
- dispatching rules of activities,
- schedule plans of resources,
- resources' behavior over time,
- strategies of resource allocation, etc.

For the initial optimization cycle of an optimization run, default values are being used for these parameters. The parameterization of later scenarios is a result of the chosen optimization algorithm and is based on the information gained in previous runs.

The second part of the optimizer module is the model creation. Based on the factory model and on the attributes from the parameterization, a new scenario is created. This scenario is stored in the data base, where the simulation module can load it.

The evaluation is the third part of the optimizer module. After processing the simulation raw data in the data preparation module, an evaluation of whether another optimization cycle is necessary or not is made. Depending on the optimization target, several key performance indicators are checked, for instance, measurements presented in SEMI (2004), production costs, slack of orders and products, balanced workforce, violation of due dates, resource utilization, etc. Based on these criteria it is decided whether the simulation result and therefore the latest scenario is acceptable or not. In addition the abortion criteria, defined as maximum number of optimization cycles or elapsed time, is checked. If a target or abortion criteria is met, the optimization is terminated. In the instance that neither a target criteria nor an abortion criteria is met, a new optimization cycle starts.

We implemented various optimization strategies using SBOP, and tested them successfully on real and generated production data of complex assembly lines of heavy machinery. For instance

- finding best values of various indicators (slack, due date, utilization, etc.) using random walk,
- extension of the three-pass heuristic approach presented in Majohr (2008),
- minimizing of slack and due date violation, and balanced workforce based on heuristic using self-organization (Angelidis, Bohn, and Rose 2012),
- minimizing of due date violation using the extended Critical Path Method presented in Angelidis, Naumann, and Rose (2012),
- minimizing of production plan costs using genetic algorithms, etc.

One of the most important advantage of the SBOP is, that all implemented optimization module can be combined with each other. So we can create easily new optimization methods, while we reduce development cost and time.

3 SIMULATION TOOL

Over the last years we worked with several general purpose simulation tools and we created for almost all of them suitable simulation modules for the second step of the optimization cycle in Figure 2. During the project, we realized that many of them are not efficient. The results using general purpose simulation tools are:

- **Model transformation and tool runtime:** In the second optimization step the time spent transforming our current meta-model into the specific simulation model, transforming of the specific simulation log into our current meta-model, and waiting for the results of simulation runs is far longer than the time the optimizer takes to prepare new scenarios. So the time of the solution search mostly depends on model transformation and simulation speed and thereby the choice of the right simulation tool.
- **Dynamic change of internal meta-model:** The platform is based on a model-driven development. The model often changes according to the new manufacturing characteristics. The effort of the customization of the new simulation model is unknown and the development costs are often high.
- **Implementation of libraries:** Functionality which is not provided out of the box can be added using scripts or own libraries which contain user created behavior. Numerous hours are spent to enhance the capabilities of these simulators to a point where it almost seems like reimplementing the simulator's core.
- **Unrealizable features:** It was not always possible to implement all needed behaviors of the real system in all simulation tools.
- **Unused active functions:** Furthermore these simulation tools try to provide a large spectrum of features. So the overhead is slowing down the simulation resulting in a solution which is costly due to the working hours spent and slow due to the interpretation of scripts, graphic routines and support for unused functions.

- **Model validation and tool comparison:** It was not always easy to validate simulation models or compare models of different tools (black box behavior). For instance, the way how the internal event list is sorted, is unknown.
- **Control by external tool:** In some cases it is quite difficult to automate the start of the simulator or to control it from outside.

The results are slow runtime, high development time/costs and sometimes unrealizable modules. For these reasons we present a specific custom-built simulator designed for this problem domain.

It uses the same internal meta-model as the platform, so time will not be spent on model transformation and its easily changeable. We implemented all the needed features and we activate them only if they are required. The tool supports a variety of real-world extensions and dedicated behavior. It can be used as an standalone application or as a library for a simulation bundle in the platform. This is a further development of the simulation tool, which is presented in Angelidis, Pappert, and Rose (2011).

3.1 Simulation Model

The current simulation model is an extension of the internal model that is presented in detail in Angelidis, Pappert, and Rose (2011). However the complexity of the problem is amplified while using:

- **Multi-skilled resources:** Activities do not have a fixed processing time but have various modes (high resource variation) that describe their execution characteristics. Activities processing time used to be dependent only on the amount of the resources assigned to it. In the current model the processing time is dependent on the amount of the required skills of the resources. Although the optimizer choose the mode for every activity, the simulation tool have to choose the available resources according a resource-qualifications table (Figure 3). More realistic models can be simulated. For the resource selection many rules are implemented, like:
 - choose first low cost resources,
 - choose first resources with a high number of availability,
 - choose first resources that probably will not be needed in the nearest future,
 - choose first resources with low utilization,
 - choose first resources with many qualifications.

Another advantage of this realization is the reduction of many alternative production paths. As explained in Angelidis, Pappert, and Rose (2011), these alternative paths come into existence because of the possibility to execute a production process in similar processing facilities. Now with this approach the table takes on the responsibility and the complexity of the production plan decreases.

- **Internal and subcontracted workers:** The simulation model has resources defined as internal workers, who have a group of subcontracted workers. These can be used when the internal scheduled workers are currently not available. This allows the optimizer to evaluate some manufacturing cost factors and analyze the due date violation over the resource utilization.

3.2 Simulation Tool Extension

We implemented the simulation core in such a way that the behavior extension is being easily realizable. It follows the main implemented extensions.

3.2.1 Restart from Specific Time Point

The simulation speed influences the optimization runtime and its improvement is an important task in the development process of the tool. A mechanism was developed that allows performing execution not only from the beginning, but also from specific time points. Therefore, when a new simulation model is created

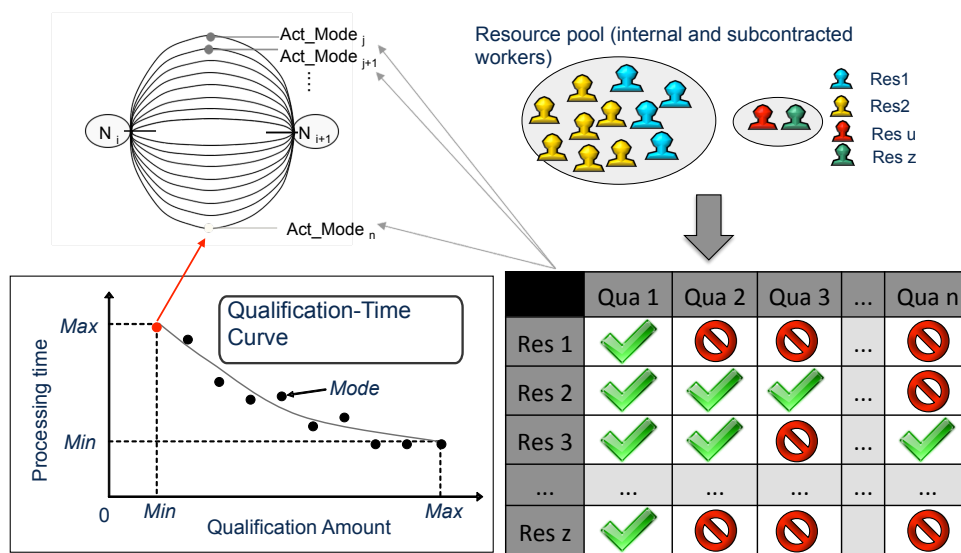


Figure 3: Extended activity mode definition.

according to the previous model, the tool can begin from the point of time where the first change appears, rather than starting from the beginning of the model. This method should reduce simulation runtime.

Modern simulation tools propose different mechanisms of the simulation runtime optimization. One of the most popular is to create snapshots during simulation. They allow to save system states during the simulation. As a result, the platform can use one of the saved states to simulate a model from a specific time point for the next optimization iterations. Snapshots distribution can be based on simulation progress, on time, on events and on other parameters. For example, simulation milestones can be created every x steps or minutes of the simulation progress. Disadvantage of this approach is that a simulation can be started only from a snapshot before the first difference and many unnecessarily saved snapshots in the simulation log.

In our approach the optimizer informs the simulation module where the first difference between the models is, so that runtime is less then the normal one. This is not always the case, for example between small models where the difference take place early. We use this extension successfully on huge manufacture models where the runtime is enough and the differences appear near the end. This is applicable for optimization heuristics that found a robust solution and easily modify the next model.

3.2.2 Resource Allocation Forecast

One of the common characteristics of the activities of our manufacture models that we found, is the huge processing time (approximately 70% between 18-72 hours). Considering also that almost all activities require a processing facility (which are few and precious), there is a result due date violations because of long waiting times to process many activities. A promising way to try avoiding it, is the following. Before the simulation core tries to distribute resources to ready to start activities, it has to analyze the remaining processing time of all already processing activities. If these remaining times are less than a given amount of time and their successors have a higher priority than the ready to start activities, then suitable resources will be reserved for these successors until the next simulation event. It is a powerful way to help the activity schedule, while most of the general purpose simulation tools, analyze only the current state ignoring the nearest future.

3.2.3 Resource Behavior

The behavior of the resources over the time is based on the description of (Angelidis, Pappert, and Rose 2011). All resources have shift plan, break plan and breakdown plan and all of them influence the state of the resource at the same time. The various distributions of resources after a shift change is implemented as well, as it is described in Angelidis, Pappert, and Rose (2011). In this paper the resource allocation took place at the initial start of an activity. The activity kept the same resource units until the end. In our model we extended it, because of the qualification table. All resources act equally as in the last model, except of the workforce. When they are on a nonschedule time or downtime they are released from the activity. The activity can start when it found new available resources according to the qualification table. The strategy from the old paper uses it as opportunity to simulate work conditions where the worker is responsible for the total process of the activity. The new strategy allows the processing of activities without the person being important. We implemented also rules that make the workforce behavior more realistic, such as not using available resources, when a schedule event (for instance shift change) will occur in less than a given amount of time.

3.2.4 Resource-Qualification Table with Dynamic State

As mentioned before in the new model, modes require qualification and not resources directly. There is a resource-qualification table (Figure 3) that allows the simulation core to select resources for every mode. This table is not static and can be changed according to many parameters that the optimizer defines (products near a due date, high utilization, high costs, etc.). It is a powerful extension to analyze the workforce behavior under special circumstances.

3.2.5 Priority Rules

The simulation core tries to start the activities by checking for available resources. For that reason dispatching rules will be used to determine the order in which the activities will be started. The rules analyze activities' data, part of the network and the simulation current state sorting activities according their criticality. We created a dispatching rule framework for creating new rules based on our meta-model. We implemented more than forty well known rules presented in Haupt (1989), Pinedo (2005), Pinedo (2008). The simulation tool has an adjustable internal list of dispatching rules to order the activities. If the values are equal, then the comparison tries the next rule. If all rules within the list cannot prioritize one activity over another we use the unique id of the activities as last deciding factor. The calculation of the prioritized values has a strong influence on the runtime of the simulation. Some activities have static values for some rules and we calculate them once (first in first out (FIFO), shortest/longest processing time (SPT/LPT), earliest/latest due date (EDD/LDD), etc.). On the other hand some rules are depended on network analysis or performance indicators and the values have to be calculated every time again, as critical ratio (CR), least/most total successors (LTS/MTS), etc.

It is important to notice that the networks have dummy activities (Angelidis, Pappert, and Rose 2011). They are used to create complex network constraints or synchronize activities. These activities have an execution time equal to zero, do not represent a real job, can always start without delay and for almost all rules the dummy activities have equal or undefinable values. For these reasons, the tool first finds all real possible successor activities (not only the immediate while ignoring the dummy successors) and then calculates a min, max or average value for the dummy activity. As illustrated in the first part of Figure 4, for calculating the value of *Dummy1* the *Act2* is needed and for *Dummy2* the *Act3* and *Act4*. Alternative the tool can remove the dummy activity from the order in which the activities will be started. All possible real successor will be add in the list and it will be sorted again. When a successor of the dummy starts then all the dummy's successors will be removed. This way help us to choose the higher prioritized activity that can start at that moment. Almost all the simulation tools that we used in the past, had problems to realize these features. Many of them choose a dummy without being able to check the real successor activities.

Another characteristic of our model is the various alternative paths of the production plan. Priority rules that are based on a network analysis as CR, LTS/MTS, minimum slack time, have various possible values. For that reason, the tool analyzes all alternative paths until the end and calculates their min, max or average value. As illustrated in the second part of Figure 4 two possible paths are defined ($Act20 - Act21 - Act23$) and ($Act20 - Act22 - Act23$). Such methods calculate more realistic values but are very time consuming. Every time an alternative path is chosen, all values change and the activities must be calculated again.

Another way to prioritize activities of production plans with alternative path characteristics is the following. Let us assume in Figure 4 that the predecessors of nodes a and b are executed. Instead of sorting a list of activities $\{Dummy1, Dummy2, Act21, Act22\}$, we created a list with lists as elements, where these elements contain the alternative successors of a node $\{\{Dummy1, Dummy2\}, \{Act21, Act22\}\}$. The internal lists have own rules and the highest prioritized activities of the internal lists will be compared with each other. That way custom rules can be created where only the prioritized elements of the alternative path compete with others activities for the resources. Such customizations are mostly unrealizable with general purpose simulations.

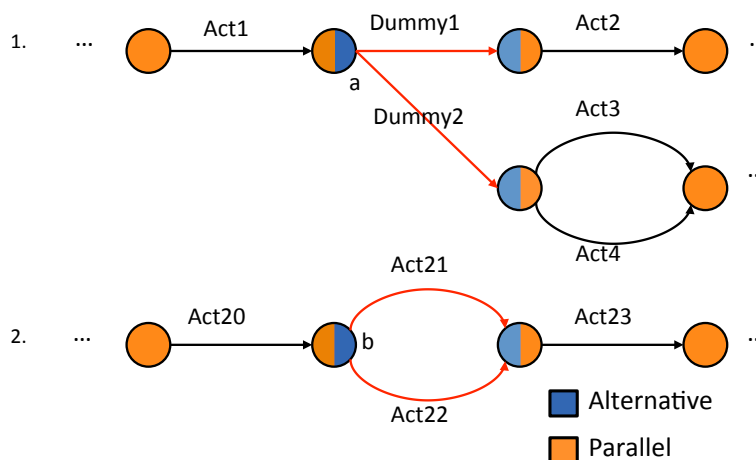


Figure 4: Example of production plan for better understanding of the dispatching rules.

4 SUMMARY AND FUTURE WORKS

We successfully extended a simulator designed specifically for the simulation of complex assembly lines which can be used by the Simulation-based Optimization Platform or as a standalone tool. The tool supports a variety of real-world extensions and dedicated behavior. At the moment is used for real manufacture models with up to 20000 activities and 20 resources with schedule plans. Currently we finished a completely new internal cycle of the simulation tool, which needs less runtime and combines the events of the resource schedules with the events that start the activities. That helps us to analyze and to distribute the available resources on a better way.

It follows a demonstrative example based on real data. A manufacture model that produces industrial machines. It has 16 internal and 6 subcontracted resources. The model contains 5 customers' orders with totally 7 products, with totally 4000 activities. The search space, according only on the amount of the

modes of every activity is $1.328061E+825$. The SBOP uses a genetic algorithm to minimize the production cost. In Table 1 is shown the runtime of every simulation model. In the simulation model are used all kind of dispatching rules. The experiment run on a standard PC (Intel Core i7-3770T @ 2.50GHz , 8GB).

Table 1: Measurement time of simulation tool (seconds).

Measurement Point	Amount	Avg	Min	Max
Read sim model from DB	110	3.412	2.772	7.214
Simulation runtime	110	3.329	1.97	5.127
Write sim log in DB	110	8.956	5.428	11.427

In the nearest future we plan to implement two more things according to the behavior of the resources. A resource setup for the processing facilities and a flow diagram that describes the behavior of the resource states is provided.

In the future the extension of 3.2.1 can be used to connect the simulation tool with an interactive Gantt Chart, where the user modifies the model. By giving new parameters or moving the executed activities a new simulation model will be created and simulated from the specific point.

ACKNOWLEDGMENTS

We would like to thank Siarhei Kazuk for his implementation work and Sebastian Carl for his discussions (especially about his idea of the resource-qualification table) during the project.

REFERENCES

- 10gen 2013. “mongoDB technology”. Accessed July. 22, 2013. <http://www.mongodb.org>.
- Angelidis, E., D. Bohn, and O. Rose. 2012. “A Simulation-based Optimization Heuristic Using Self-Organization for Complex Assembly Lines”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspace, R. Pasupathy, O. Rose, and A. M. Uhrmacher. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Angelidis, E., A. Naumann, and O. Rose. 2012. “An Extended Critical Path Method for Complex Assembly Lines”. In *Proceedings of the 2012 Industrial Engineering Research Conference*, edited by G. Lim and J. Herrmann.
- Angelidis, E., F. M. Pappert, and O. Rose. 2011. “A Prototype Simulation Tool for a Framework For Simulation-based Optimization of Assembly Lines”. In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. Creasey, J. Himmelspace, K. White, , and M. Fu, 2383–2394. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- April, J., F. Glover, J. P. Kelly, and M. Laguna. 2003. “Simulation-Based Optimization”. In *Proceedings of the 2003 Winter Simulation Conference*, 71–78. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Buddhakulsomsiri, J., and D. S. Kim. 2006. “Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting”. *European Journal of Operational Research* 175 (1): 279–295.
- Haupt, R. 1989. “A survey of priority rule-based scheduling”. *Operations-Research-Spektrum* 11 (1): 3–16.
- Law, A. W., and M. G. McComas. 2002. “Simulation-Based Optimization”. In *Proceedings of the 2003 Winter Simulation Conference*, 41–44. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Majohr, M. F. 2008. *Heuristik zur personalorientierten Steuerung vom komplexen Montagesysteme*. Ph. D. thesis, Dresden University of Technology.
- OSGi Alliance 2012. “OSGi technology”. Accessed January. 18, 2013. <http://www.osgi.org>.

- Pappert, F. M., E. Angelidis, and O. Rose. 2010. "Framework for simulation based scheduling of assembly lines". In *Proceedings of the 2010 Winter Simulation Conference*, 1690–1968. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Pinedo, M. L. 2005. *Planning and Scheduling in Manufacturing and Services (Springer Series in Operations Research and Financial Engineering)*. Springer.
- Pinedo, M. L. 2008, July. *Scheduling: Theory, Algorithms, and Systems*. 3rd ed. Springer.
- SEMI 2004. "EMI E10-0304E: Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)".
- Shapiro, A. 1996. "Simulation Based Optimization". In *Proceedings of the 1996 Winter Simulation Conference*, edited by J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, 332336. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

EVANGELOS ANGELIDIS is a Ph.D. student at University of the Bundeswehr Munich, Germany. He is a member of the scientific staff of Prof. Dr. Oliver Rose at the Chair of Modeling and Simulation. He received his M.S. degree in Computer Science from Dresden University of Technology. His research focuses on the simulation and optimization of complex assembly lines. His email address is evangelos.angelidis@unibw.de.

DANIEL BOHN is a Software Engineer at Bosch Software Innovations. He is also a Ph.D. student at University of the Bundeswehr Munich, Germany. He is a member of the scientific staff of Prof. Dr. Oliver Rose at the Chair of Modeling and Simulation. He received his M.S. degree in Computer Science from Dresden University of Technology. His research focuses on the simulation and optimization of complex assembly lines. His email address is daniel.bohn@unibw.de.

OLIVER ROSE holds the Chair for Modeling and Simulation at the Department of Computer Science of the University of the Bundeswehr Munich, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of IEEE, INFORMS Simulation Society, ASIM, and GI. His email address is oliver.rose@unibw.de.