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

AN OPTIMIZATION FRAMEWORK FOR WAFERFAB PERFORMANCE ENHANCEMENT

Daniel Noack
Boon Ping Gan
Peter Lendermann

D-SIMLAB Technologies
9 Jurong Town Hall Road
609431, SINGAPORE

Oliver Rose

Institute of Applied Computer Science
Dresden University of Technology
Dresden, 01062, GERMANY

ABSTRACT

A typical wafer fab requires numerous decisions for daily operations. Even small decisions on system configurations may have significant impact on the overall fab performance. One of the most critical performance measure is cycle time, where just one day of reduction could represent significant cost savings. In this paper we describe an automated simulation-based optimization method to improve fab configurations. We illustrated our method through a case study that involves optimization of multiple decision variables in a wafer fab. The objective of the optimization is to reduce the cycle time. We show the potentials of such an optimization through achieving an improvement of 15% in cycle time for a furnace toolset.

1 INTRODUCTION

There has been significant amount of work that uses simulation to improve fab performance, by evaluating and changing configuration settings of the fab operations. These work attempted to figure out the best configurations for decision variables. For example, authors tried to derive the best batching rules in (Akçali et al. 2000) and (Mittler and Schoemig 1999), the best recipe dedication policy in (Quek et al. 2007), optimal batch size in (FabTime 2008), and optimal lot size in (Potorad, Winz, and Kam 1999). All of these studies defined the relation between configuration settings for a certain decision variable and fab performance. They demonstrated that significant improvement can be achieved through optimization of some configuration settings in a wafer fab.

One common feature of these studies was that the optimal or improved configuration was found through a manual optimization process. The authors derived several possibly good configuration and then evaluated these configuration through computer simulation. Manual optimization, however, has the disadvantage that it requires highly qualified manpower and it takes a lot of time.

The recipe dedication problem presented in Quek et al. (2000) is a good example to illustrate the complexity and inefficiency of manual optimization. Typically, a fab has to handle a large number of product mixes. Some recipes can be run only on certain tools within a toolset, the respective products can therefore be processed on these tools only. In turn, if a tool is not qualified with that recipe, it cannot process certain products. A manual optimization of that would require the expert to generate suitable scenarios, start the simulation, wait for results, analyze the results, and decide the next course of actions or next set of configuration for experimentation. With expert involvement, the search space could possibly be reduced. In addition, the optimization process is very target-oriented. But the problem is that the search space will increase dramatically with increasing problem size. Equation 1 shows the maximum problem size *N* as follows:

$$N_{combinations} = \prod_{i=1}^{t} 2^{r_i} \qquad (1)$$

where *t* represents the number of tools in a toolset and *r* the number of available recipes for each single tool.

An example is a toolset with only 5 machines and 5 recipes each machine. The recipes are able to run on all tools. For one tool the number of combinations is 2^5. By multiplying the number of combinations of all tools you get the number of possible combinations for the whole toolset. There are already 2^{25} possible recipe configurations for the toolset.

Humans are creative and know many relations between fab settings and fab performance and therefore are able to generate new ideas out of this domain knowledge to solve certain problems . For example, a human is able to create a better dispatch rule to improve fab performance. However, even if an expert has to run only 1000 configurations (which is less than 1% of the search space) it will take several weeks already to do such an optimization. If changes such as product mix or introduction of new recipes happen on the shopfloor then the earlier efforts of optimization will go to a waste.

By integrating simulation-based optimization into a fab decision process the disadvantages associated with human involvement in the optimization process will be addressed. Automated simulation-based optimization is useful for repetitive tasks, like any automatic solution. Considering the recipe dedication problem described before, it is easy for a computer to deal with this problem. A computer can change the value of such a predefined decision variable, evaluate and improve the fab performance iteratively. It can run millions of simulation scenarios automatically and select the best scenario. Simulation-based optimization is widely accepted and provides huge problem solving potential (Law and Kelton 2000).

In this case study it is our motivation to figure out the potentials of solving fab operational optimization problems with simulation-based optimization. We evaluated the potential gain in fab performance by applying a simulation-based optimization method. It is our target to reduce the manual simulation effort by automating this procedure, delivering the results before it is outdated, and to improve the quality of the solution. An automatic optimization is able to run more iterations automatically. By using a target oriented optimization algorithm it will be possible to find a better solution in a huge search space. We also see the potentials of using simulation based optimization in dynamic manufacturing environments. Only such a system can keep the fab configuration up to date, even if more drastic changes such as changes in product mix occur.

The remainder of this paper is organized as follows: In Section 2 we describe the simulation-based optimization approach. Section 3 contains a description of the simulation model used in this case study. In Section 4 we give an overview of the optimization framework and the algorithm used. We present the experimental results in Section 5. In the last sections we conclude our case study and give an outlook of our future plans.

2 SIMULATION BASED OPTIMIZATION

This section provides a general description of a simulation-based optimization system. Furthermore the simulation and optimization tools as used for this case study are presented.

2.1 Overview of Simulation-Based Optimization

To improve fab performance, a simulation-based optimization approach is used. A detailed fab model is necessary to represent a realistic behavior in the simulation. Depending on the simulation model settings, the simulation generates the key performance indicators. The optimization evaluates these key performance indicators to find different model settings for the next simulation. By doing this, the optimization tries to improve factory performance. Changes in the model setting affects the factory performance. It is an iterative cyclic process.

2.2 D-SIMCON Discrete Event Simulator

In this case study the simulation model is running with D-SIMCON, a data driven discrete event simulator, see (D-Simcon 2008). It is highly customizable and allows a high level of detailing for critical tools and low level of detailing for non-critical tools. This capability makes it feasible to simulate a huge fab model with sufficient level of detail while the simulation time remains very fast. One year simulated time with a model as described below takes less than three minutes to execute.

2.3 D-SIMCON Optimization Framework

The D-SIMCON optimization framework is an independent routine that interacts with the simulation tool. It is designed to interact with any other simulation tool like Factory Explorer, see (Factroy Explorer) or AutoSched AP, see (AutoSched AP). It is easily extensible and configurable to address a wide range of fab operation optimization problems. It contains solutions for different types of problems like selection, permutation and integer problems. A typical selection problem is the recipe dedication where a decision on which tool should run which recipe for optimal fab performance is made. An example of a permutation problem is the lot sequencing for cluster tools. Changes in lot sequence affects the throughput of a cluster toolset. Typically two lots, with the same or different recipes, can be processed in parallel on a cluster tool. Some recipes need the same subset of chambers for processing while others require a different subset of chambers. If lots belonging to the former group of recipes are processed together, the capacity of the tool is reduced as wafers are processed in a sequential mode. As such, changing the queue sequence of lots with different recipes will have impact on the capacity of the toolset. An example of an integer problem is the determination of the optimum batch size. Different integer values for the batch size have different impact on the cycle time and capacity. By increasing the batch size the capacity also increases, assuming that the processing time is still the same. But an excessive batch size could have negative impact on the cycle time because it may take too much time to accumulate a full batch.

3 SIMULATION MODEL

In this case study the simulation model is running in D-SIMCON, described in Section 2 . The model focuses on the metallization process of a wafer fab, see (Gan 2006). A typical product flow enters the fab and loop several time before it leaves the fab. In the loop it gets processed on wet bench, furnace and etch tools, see Figure 1. Depending on the number of metal layers, the lots have to run on the main processing loop for up to 7 times. All lots are divided into 12 different product flows with up to 60 steps for each flow. The whole model contains about 40 tools in 15 groups. A high level detailing is available for cluster, wet bench and batch tools – up to the chamber and wafer level. There are 9 different recipes available for wet bench tools and 7 recipes for cluster tools to model different wafer flows inside a tool. Toolsets that are not of interest are modeled as delay processor. Such a delay processor is assumed to have unlimited capacity, and the delay is equivalent to the average cycle time of the toolset.

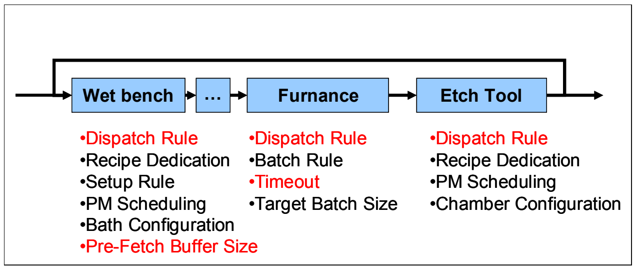

Figure 1: Simulation Model

Figure 1 summarizes the decision variables at each decision points. In this case study we optimize the following decision variables: (1) dispatch rules for all toolsets, (2) timeout at furnace, and (3) pre-fetch buffer size for wet bench.

Four variations of dispatch rules are used for all toolsets except wet bench. They are First-In-First-Out (FIFO), Smaller Step Number (SSN), Most Remaining Steps First (MRS) and Least Remaining Steps First (LRS). The dispatch rule SSN sorts the lots according to their number of completed steps in the product route. Lots with a low number of completed steps will be processed first. MRS sort the lots according to the number of steps in the product route until completion. It prefers the highest number of remaining steps. The dispatch rule LRS also sorts the lots according to the number of steps in the product route until completion. The only difference to MRS is that it prefers lots with the lowest number of remaining steps. All these dispatching rules affect the cycle time of the lots. Three dispatch rules are available for wet bench tools. They are Same Recipe (SR), Earliest Start (ES), and Earliest Finish (EF). The dispatch rule SR prefers lots that are of the same recipe as the last loaded lot's recipe. The ES rule gives higher priority to lots that are able to start the earliest, considering the no over-dipping and no overtaking constraint of wet benches. On the other hand, the EF rule prefers those lots that can be completed on the wet bench tool the earliest. The dispatch rules at the wet bench affect the cycle time of the lots and the capacity of the whole toolset.

Another decision variable is the timeout for furnace. It is the maximum time that a tool will wait for a full batch. If a timeout occurs, the batch processing will start immediately even if the tool is not fully loaded. This decision variable affects cycle time and toolset capacity.

The pre-fetch buffer size defines the volume of lots, an operator collects for a wet bench tool while it is processing. The target is to create a batch before processing of the old batch finishes. The operator is able to start this new batch without any losses in cycle time and capacity. The pre-fetch buffer size mainly affects toolset capacity and the cycle time of the lots.

```
MACHINE RECIPE   BATH_1 BATH_2
AMU6-01 EZ-1A    20,00  10,00
AMU6-01 EZ-2A    30,00  10,00
AMU6-01 BV-1A    26,67  7,00
AMU6-02 BV-2A    6,67   6,67
AMu6-03 BEM-1A   21,00  10,00
AMU6-03 BEM-2A   35,00  10,00
AMu6-03 BEV-1A   18,00  10,00
AMU6-14 EPW-1A   20,00  10,00
AMU6-14 BEW-0A   5,00   10,00
```

Figure 2: Input File

The simulation input files contain the whole model information for the data driven simulator. Basically all model information can be changed in the optimization by changing the values in the input files. Figure 2 shows the recipe dedication for wet bench tools in the simulation model. The optimization changes the recipes assigned to certain toolsets. Recipes can be added or deleted by adding or deleting lines with the recipe name and the machine name. The simulated behavior will then change accordingly.

```
PERFORMANCE_PARAMETER     VALUE
Fab_meanWIP:              109.6503
Fab_completedLots:        7992
Fab_releasedLots:         8114
Fab_meanCT :              6801.663
```

Figure 3: Output File

The simulation output file contains the key performance indicators. Performance indicators like mean cycle time and mean work in progress are available after the simulation, see Figure 3. On toolset level many parameters like utilization and wafer per hour are available too. The optimization evaluates the performance of one simulation by accessing the output file.

4 OPTIMIZATION

This section gives an overview of the general optimization framework and the optimization procedure. The objective function of the case study, the algorithm used for the optimization and the termination criteria are described in detail.

4.1 Overview Optimization Framework

The optimization framework is divided into different modules, see Figure 4. Each module can be modified easily due to the modular structure of the optimization framework. For example a user can change its objective function through a few mouse clicks, a developer can implement new optimization search strategies without influencing the stability of the software, etc. Due to this high customization possibility, the framework is applicable to a wide range of fab optimization problems.

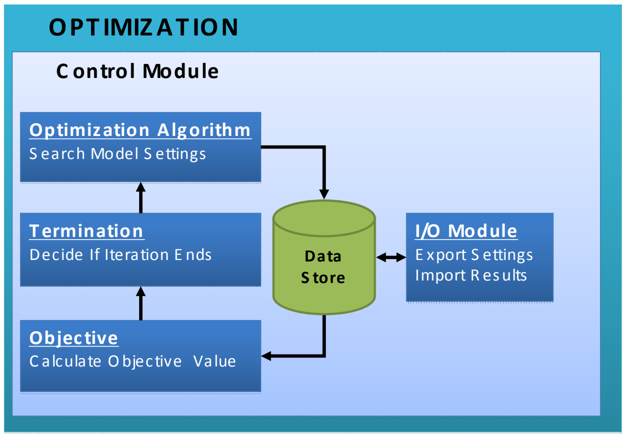

Figure 4: Optimization Framework

The input output module is the access point to exchange all relevant data with the simulation tool. The task of the data store is to save all simulation model settings and corresponding simulation results for all iterations. The values for all decision variables and decision points can be saved. Also, the key performance indicators are available for each previous simulation. The objective module contains the objective functions. Depending on its current objective it calculates the objective value from the key performance indicators. The termination module decides when the cyclic process will stop. After the optimization stops, the results will be presented to the user. If the optimization continues, new model settings will follow. In the optimization algorithm, the optimization strategy is implemented. It finds new simulation settings and store these settings in the data store.

4.2 Optimization Algorithm for the Case Study

The optimization begins with the current fab settings as a reference for further performance improvement. This reference setting will be replaced when an improved performance with refined fab settings is observed during the optimization process. The optimization algorithm is illustrated in Figure 5. In iteration 1 and 2 no improvement can be achieved while the settings in iteration 3 does improve the objective. The optimization discards the model settings in iteration 1 and 2. Iteration 3 is the new baseline for subsequent improvements. Further changes will be accepted only if they improve the objective from iteration 3. All changes are based on the previous best iteration.

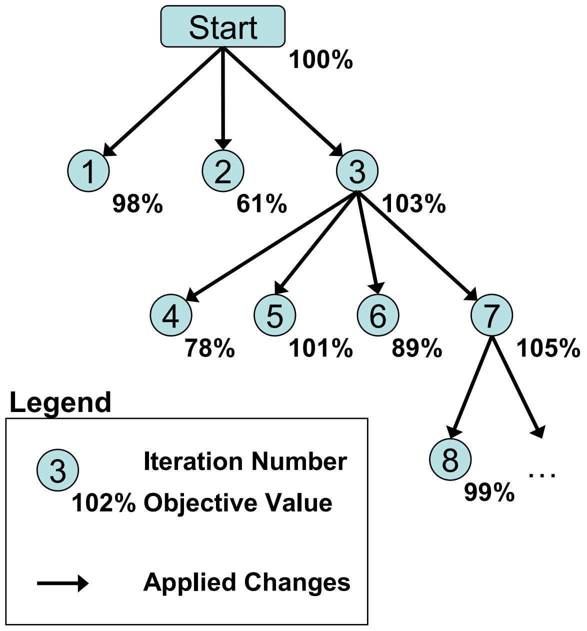

Figure 5: Optimization Progress

For all scenarios in this case study a random optimization algorithm is used to change the simulation model settings. It chooses decision point, decision location and the values randomly. The decision variables are dispatch rule, timeout and pre-fetch buffer size. Decision locations are toolsets such as wet bench, furnace and cluster tools. At the beginning of the optimization significant improvements can be observed. That means many decision variables, many decision locations will change between two iterations. For example the dispatch rule changes for wet bench, batch tools and furnace. The pre-fetch buffer size changes for wet bench and timeout changes for batch tools. At the end of the optimization only small improvements are made. Only one decision variable and one decision location are changed. The reason for altering the step width is to prevent the optimization algorithm from stopping at a local optimum.

The optimization will terminate after 100 iterations. The value 100 was chosen because the experiments show that after 100 iterations no changes or only minor changes occur. We use the same number for all scenarios, because all optimization scenarios have to be comparable among each other. All optimization scenarios have approximately the same computation time. The objective function of this study is fab cycle time reduction. Table 1 contains an overview of the optimization.

Table 1: Optimization Overview

Module	**Implementation**
Optimization Algorithm	Random Search
Objective Function	Minimum Fab Cycle Time
Termination	Fixed to 100 Iterations
Decision Variable	Dispatch Rules Batch Timeout Pre-fetch buffer size
Decision Location	Wet bench Furnace Etch Toolset

5 EXPERIMENTAL RESULTS

In this section we present scenarios to demonstrate that our simulation-based optimization is capable of improving the current fab settings. So the current fab settings will be applied in the simulation model. The cycle time will be improved by changing only the following decision variables; dispatch rules, batch timeout and pre-fetch buffer size. The optimization starts with current fab cycle time at iteration zero, see Figure 6. As can be seen, the cycle time was reduced iteratively while the optimization was running.

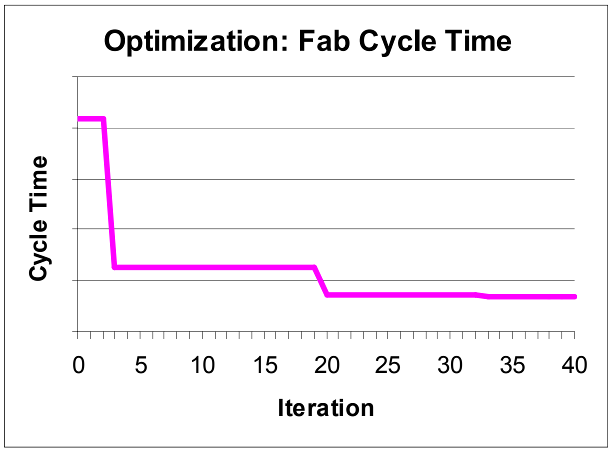

Figure 6: Optimization Iterations

Figure 7 depicts the cycle time reduction after optimization for each toolset. For wet bench we achieved a cycle time reduction of 1%, for furnace 15.4% and for cluster tool 7.4%. Further improvements have been achieved in terms of capacity for cluster tool. The average capacity of all cluster tools in the toolset increases by 1.5 wafer per hour.

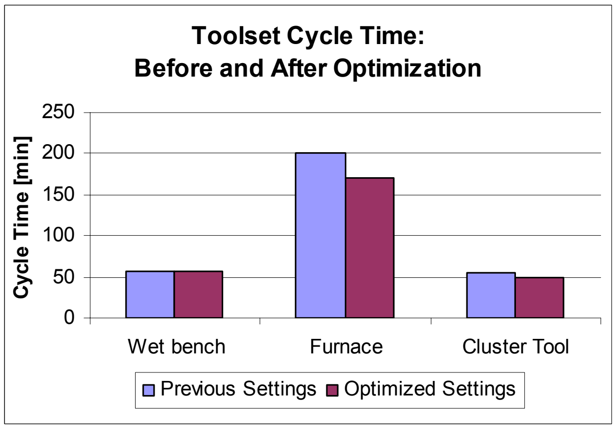

Figure 7: Optimization Iterations

The reasons for all the improvements mentioned above are changes of the simulation settings as determined by the optimization. For the wet bench the EF dispatching rule now replaces the SR dispatching rule on two toolsets. For the furnace the LRS rule, and for the cluster tool the SSN dispatching rule has been applied instead of FIFO. In this scenario the batch timeout and pre-fetch buffer size have only minor effects on cycle time.

The changes of dispatching rules mainly affect the toolset cycle time. By using the LRS rule, lots that will finish soon are given preference. They will finish earlier and therefore the cycle time is lower. The work in progress in the main product loop will also be reduced. Possible reasons for the cluster tool capacity improvements are the SSN dispatching rule and a different upstream behavior from the furnace and wet bench tool. By replacing the SR dispatching rule for the wet bench, a more continuous product flow with less variability for the cluster tool will be achieved.

6 CONCLUSIONS

In this case study we present our simulation-based optimization system. We show that the optimization algorithm is capable of improving cycle time by changing multiple decision variables. The optimization achieves improvements of more than 15% in cycle time for furnace toolset. It is a significant improvement in terms of time as compared to the manual experimentation, which took several weeks to complete. With the automated simulation-based optimization, we completed the optimization in less than five hours. To underline the advantages of simulation-based optimization, we will apply model changes and run the optimization again. We want to demonstrate that further optimization is capable to achieve further improvements. These model changes can be a significant tool downs, changing tool configuration or changing product volume.

7 OUTLOOK

It is our vision to improve the quality and extensibility of the D-SIMCON optimization framework, realizing a fully automated optimization process. More fab operation related problems such as lot sequencing, release strategy, preventive maintenance management, predictive operation management will be addressed using our optimization framework. In addition, more sophisticated dispatching, batching and tool allocation policies will be incorporated and parameterized. This increases the number of decision variables tremendously. The challenge will be to resolve the scalability and complexity of the optimization. The optimization algorithm have to be more intelligent to deal with complexity and deliver high quality results in time. One idea is the usage of an integrated knowledge base to support the optimization. The optimization will be able to use background information such as relations between decision variables and key performance indicators.

REFERENCES

AutoSched AP. 2008. <http://www.brooksoftware.com/pages/246_autosched_overview.cfm> [accessed June 1, 2008].

Akçali E., R. Uzsoy, D. G. Hiscock, A. L. Moser, and T. J. Teyer. 2000. Alternative Loading and Dispatching Policies for Furnace Operations in Semiconductor Manufacturing: A Comparison by Simulation. *Proceedings of the 2000 Winter Simulation Conference.* Piscataway, NJ: Institute of Electrical and Electronics Engineers.

D-Simcon. 2008. <http://www.d-simlab.com/page.php?page=semicon> [accessed June 1, 2008].

FabTime. 2008. <http://www.fabtime.com/ctbatch.shtml> [accessed June 1, 2008].

Factory Explorer. 2008. <http://www.wwk.com/fxover.html> [accessed June 1, 2008].

Gan B.P., P. Lendermann, P. T. Quek, B. v.d. Heijden, C.C. Chin, and C.Y. Koh. 2006. Simulation Analysis on the Impact of Furnace Batch Size Increase in a Deposition Loop. *Proceedings of the 2006 Winter Simulation Conference.* Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Law A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*, 3d ed., McGraw-Hill, New York.

Mittler M., and A. K. Schoemig. 1999. Comparison of Dispatching Rules for Semiconductor Manufacturing using Large Facility Models. In *Proceedings of the 1999 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Potoradi J., G. Winz, and W. Kam. 1999. Determining Optimal Lot-Size for a Semiconductor Back-End Factory. In *Proceedings of the 1999 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Quek P. T., B. P. Gan, S. L. Tan, C. L. Peng, and B. v.d. Heijden. 2007. Analysis of the Front-End Wet Strip Efficiency Performance for Productivity. In *Proceedings of the 2007 ISSM Conference*.

AUTHORS BIOGRAPHIES

DANIEL NOACK is a Research Engineer at D-SIMLAB Technologies. His focus is on simulation and simulation-based optimization. He is also a PhD student at Dresden University of Technology. 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 email address is <daniel.noack@inf.tu-dresden.de>.

BOON PING GAN is the Chief Technology Officer and Founder of D-SIMLAB Technologies Pte Ltd, Singapore. He is responsible for product development and execution of client projects in the semiconductor industry. He received a Bachelor of Applied Science in Computer Engineering and a Master of Applied Science from Nanyang Technological University of Singapore in 1995 and 1998, respectively. His email address is <boonping@d-simlab.com>.

PETER LENDERMANN is the Co-Founder and CEO of D-SIMLAB Technologies, a Singapore-based company providing simulation-based decision support solutions and services to Aerospace, Semiconductor Manufacturing and other asset-intensive industries. He has been engaged in the simulation community since the early 1990's when he worked in a multinational research collaboration at the European Laboratory for Particle Physics CERN (Geneva, Switzerland) and Nagoya University (Japan). In 1996 he joined a German consulting firm where he was responsible for business process re-engineering projects with numerous process manufacturing, aviation and automotive clients in Europe, Canada and China. In 2000 he joined the Singapore Institute of Manufacturing Technology where he led the simulation-related research activities until spinning them off into D-SIMLAB Technologies. Peter holds a PhD in Applied High-Energy Physics from Humboldt- University in Berlin (Germany) and an MBA in International Economics and Management from SDA Bocconi in Milan (Italy). He is also an Adjunct Associate Professor at the Department of Industrial and Systems Engineering at the National University of Singapore. His email address is <peter@d-simlab.com>.

OLIVER ROSE holds the Chair for Modeling and Simulation at the Institute of Applied Computer Science of the Dresden University of Technology, 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@tu-dresden.de>.