Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

REDUCING SIMULATION MODEL COMPLEXITY BY USING AN ADJUSTABLE BASE MODEL FOR PATH-BASED AUTOMATED MATERIAL HANDLING SYSTEMS – A CASE STUDY IN THE SEMICONDUCTOR INDUSTRY

Sebastian Rank
Christian Hammel
Thorsten Schmidt

Professur für Technische Logistik
Technische Universität Dresden
01062 Dresden, GERMANY

Germar Schneider

Infineon Technologies Dresden GmbH
Königsbrücker Straße 180
01099 Dresden, GERMANY

ABSTRACT

Usually simulation studies of automated material handling systems of semiconductor fabs are extremely time consuming. This is due to the high detail of models used for investigations which are partly provided by transportation system suppliers. These models only provide poor possibilities for adjustment and are computationally expensive. The article will address these issues by proposing an adjustable supplier-independent base simulation model. It allows easy building, adjusting, and running simulation models of path-based systems without deeper programming knowledge. A use case of Infineon's Dresden fab revealed simulation results with an accuracy in the same range as the supplier models while disregarding a few details and thus showing significant time savings in modeling and adjusting the system as well as running simulation studies. This can be done by choosing an appropriate level of abstraction.

1 INTRODUCTION

Efficiently planning and running an automated material handling system (AMHS) mostly requires forecasting of its behavior. Right decisions regarding the application of dispatching strategies or the number of vehicles depend on the information about expected queue lengths, bottlenecks, tool utilizations, and so on.

The semiconductor sector with its extremely expensive manufacturing equipment is one of the most capital-intensive industries in the world (Peng et al. 2012). Considering the increasing level of automation of handling systems, it is notably important to know about their systems' behavior because even small deficiencies can lead to massive losses (Terwiesch and E. Bohn 2001, Arzt and Bulcke 1999). However, experiments with the real system are rarely possible because the system does not exist at all or it could be damaged. In addition, real system experiments may be non-economical due to long evaluation periods or interferences with operational activities. Consequently, model building activities and running experiments with the model come into focus.

In general there are two options available for analyzing systems and models. It is either possible using mathematical/analytical methods like queuing theory or discrete event simulation (Arnold and Furmans 2009). This paper focuses on rather complex systems and thus the simulation based approach is the appropriate choice (Arnold and Furmans 2009, Law and Kelton 2000, Thesen and Travis 1989). Discrete event simulation (DES) is widely accepted, established, and applied in industrial domain (VDI 3633 2010, Kuhn et al. 2010, Fonseca et al. 2003 – see also chapter 2). One of the systems where this approach is already applied is the AMHS of Infineon's intended pilot line of power semiconductors based on 300 mm technology. While investigating different setups in terms of layouts, ramp levels, control strategies, etc. it became obvious that using supplier provided models is insufficiently productive. The often highly detailed models *may* lead to a high degree of result accuracy but lack of adjustability and runtime performance. On the other hand, currently, there are only insufficient approaches to easily model, adjust, and visualize an AMHS of a semiconductor fab in a generic simulation model. The objective of this research is to overcome this issue.

The discussion is structured as follows: An overview of related work is given in section 2. Approaches to simulate AMHS are examined and objectives to determine the correct level of model complexity are discussed. In section 3 our newly developed adjustable base model for path-based systems will be introduced. Based on this, in section 4 a case study with Infineon's Dresden fab will be presented and a comparison with a validated supplier model will be conducted. Subsequently in section 5 the findings will be discussed, considering result accuracy, runtime performance, and adjustability against the background of different abstraction levels of various models. This is followed by a summary in chapter 6 with outlooking remarks.

2 RELATED WORK

A semiconductor AMHS is considered to be a system (usually) consisting of a track system, transportation vehicles, buffers (e. g. stockers), and tools. Thus, an AMHS includes a transportation and a production system. Wafers are transported in front-opening unified pods (FOUP) typically with maximum capacity of 25 wafers (Gartland 1999).

2.1 Simulation of AMHS in the semiconductor industry

Simulating automated material handling systems in the semiconductor industry received and receives noteworthy attention—especially with the switchover of the wafer diameter from 200 mm to 300 mm and the implementation of corresponding new transportation and automation concepts (Han et al. 2005, Agrawal and Heragu 2006, Brain and Lin 2007).

Comprehensive literature reviews of simulation studies can be found in Tyan et al. (2004) and Montoya-Torres (2006). Among others, the authors describe in their survey how the focus moves from "simple" scheduling and dispatching problems to highly sophisticated and fully automated fab models.

Some more recent AMHS simulation studies concentrate on e. g. finding optimal numbers of vehicles in a bay, optimizing delivery, and retrieving times or finding algorithms for balancing track utilization (Qin et al. 2013, Lin and Huang 2014, Kiba et al. 2009, Hammel et al. 2012).

Nearly all AMHS of semiconductor fabs show similarities in terms of their structure (see introducing paragraph of section 2) and the basic process logic: Wafers get recurrently processed on tools and are moved by the AMHS. So it is an intuitive approach designing adjustable simulation models or suitable frameworks. This allows reduced modeling efforts and increases model accuracy (Mackulak et al. 1998). In this regard first discussions go back to Nadoli and Pillai (1994). The authors describe some general aspects of designing AMHS simulation models. Tyan et al. (2004) present an integrated modeling framework for a simulation based automated fab. It allows modeling the manufacturing process as well as the AMHS in spine layout. Similar research can be found in Pillai et al. (2004) or Nazzal and Bodner (2003).

In Mackulak et al. (1998) the vehicle routing logic as part of the AMHS is put into a generic simulation model. In a use case of a clean room vendor, they observed significant time savings due to model reuse. Schulz and Stanley (2000) report on investigations with a (Infineon specific) generic model which combines conveyor (inter-bay) and monorail/hoist systems (intra-bay). In either case it is not clear whether the generic model can be used in general.

A comprehensive generic simulation modeling framework for simulating semiconductor fabrication lines is reported in Kim et al. (2009). The researchers describe a layout modeling software called "AutoLay" and an associated generic simulation software called "AutoLogic". Both are proprietary tools. AutoLogic is capable getting extended due to its module-based structure. The framework has been employed at Samsung for several simulation projects and revealed model building time reductions from two weeks to half a day.

The next logical step is not only to provide a generic simulation model but to develop automatically, iteratively new model initializations and analyze their outcomes. So simulation experiments can be rapidly created which supports the decision making process and the improvement of the underlying system. Some recent research and literature on this topic can be found in Lim and Seo (2014) and Wagner et al. (2014). The latter approach is planned to adopt the generic model proposed in this paper.

2.2 Level of abstraction

The abstraction level of a model is understood as a figure, describing the amount of information a simulation model contains. The higher the quantity of information in a model the lower the level of abstraction (Law and Kelton 2000, Benjamin et al. 1998). Furthermore, by definition, complexity specifies the systems' dynamics which arises from system elements interaction (Adami 2002). For simplification—even if it is not entirely correct—in the scope of this research the terms "highly detailed model" and "a complex model" are connected to a model of low abstraction level and vice versa.

In the context of this paper there are mainly two reasons why the level of abstraction should be discussed and carefully chosen. First, a highly detailed simulation model of an AMHS goes along with a high complexity and makes both model building times and runtimes expensive (see most of the sources stated above and e. g. Nazzal and McGinnis 2007). Secondly, a pre-programmed generic model is supposed to be provided. As this model is intended to be applicable and adjustable for a wide range of semiconductor AMHS a common ground has to be found which inevitably results in model simplification and abstraction.

However, from our experience a highly detailed model is asked by project partners quite often. This is due to the widely spread assumption of getting more precise results by models with a lower level of abstraction (e. g. Kiba et al. 2009, Pillai et al. 2004 or Astrup et al. 2008). To some extend this might be true but from some point the gradient of gaining result accuracy decreases rapidly or even gets negative—figure 1 illustrates this (Benjamin et al. 1998, Chwif et al. 2000, Robinson 2007, Astrup et al. 2008).

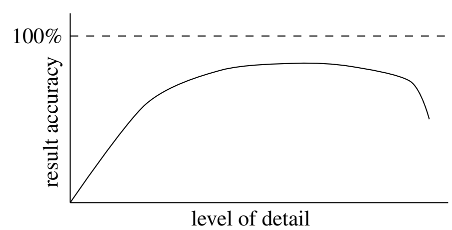

Figure 1: Dependency of result accuracy and model abstraction level—similar to Chwif et al. (2000)

Even if the majority of researchers pronounce something like "as simple as possible but not simpler" or "the smallest possible number of parameters for adequate representation of data", model complexity of automated material handling systems is increasing rapidly (Thesen and Travis 1989, Astrup et al. 2008). Chwif et al. (2000) call this effect the "include all syndrome" and discuss possible reasons like "nowadays we have computational power to deal with huge and complex simulation models".

There is no general definition of what a complex AMHS model is (Chwif et al. 2000) and of course, the decision of how much detail should be implemented in a model strongly depends on the questions that need to be answered with the model. So for instance, if the objective consists in roughly estimating the queue length in front of a tool, a less detail model is needed than if the aim is to investigate the fabs throughput under the condition of varying lot sizes. Nevertheless, the literature provides some guidelines to support choosing the right abstraction level and to appropriately classify them (e. g. Thesen and Travis 1989, Robinson 2007 and Jimenez et al. 2008). The proposed model in section 3 refers to this. Its "appropriate" level of abstraction is found by trial and error as suggested in Benjamin et al. (1998).

3 BASE MODEL

There are several DES tools available at the market (for an overview see e. g. Swain 2013). The herein presented base model utilizes AutoMod. Since AutoMod provides a compiled model, it is one of the fastest commercial DES tools and hence the de-facto standard in the academic and industrial semiconductor domain (see sources in section 2.1 and especially Hallenborg and Demazeau 2008, Muller 2013).

As presented in section 2.1, there have been several attempts in the past developing or providing more or less generic simulation models. From our point of view the approaches lack in different functionalities, capabilities, and features or are subjected to restrictions—mainly they are highly vendor specific, only allow particular layouts, use proprietary tools in addition to the simulation tool, or do not model equipment properly. So the newly presented model will meet the following primary and secondary requirements.

First of all, (1) besides AutoMod no other proprietary software or tools need to be applied—apart from a text editor to manipulate text files. Hence model building and adjustments in terms of e. g. defining objects, drawing layouts, or implementing code for new strategies is done in AutoMod. The model is (2) supplier independent and intended to be layout- and data-driven. This means it can be adjusted without deeper programming knowledge by simply configuring predefined parameter files and by using the AutoMod provided graphics based layout editor for placing tracks and equipment. Therefore, (3) the model is generic and able to run different simulation instances without changing code. Furthermore (4) the base model is highly flexible: It is able to simulate segregated MHS as well as unified ones (see figure 2). It also covers different storage concepts (e. g. stockers or side track buffers).

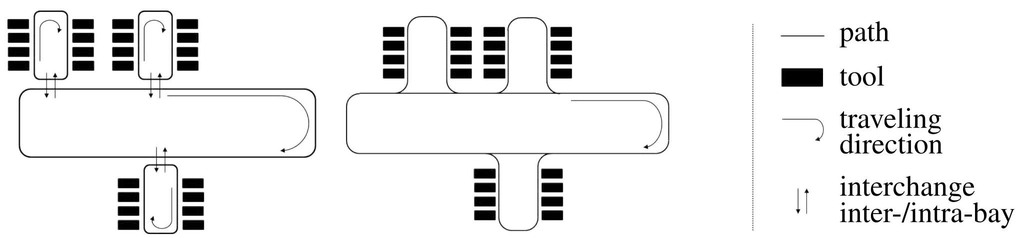

Figure 2: Different transportation system paradigms: segregated MHS (left) vs. unified MHS (right)

In order to track interferences between production processes and material handling processes, (5) both systems are integrated in the generic model. The production process allows batch processing. The AMHS covers path-based systems like overhead hoist transportation systems or guided floor conveyors.

In addition to the features (1) to (5) the proposed model is supposed to be as fast as possible by choosing the *right* level of abstraction (see section 2.2) and is reliable. The associated source code is intended to follow the principles of good programming. Referring to Sommerville (2010), the model/program has a neat code layout, uses meaningful names, is well commented, and uses the syntax in a way that robustness and readability is ensured. This also guarantees further implementations of code in order to extend and adjust model functionality.

As mentioned, it is not easy to state a specific model abstraction level. However, to give an idea of the herein presented model's complexity Jimenez et al. (2008) is cited. Accordingly, the level of detail of the AMHS model can be described approximately as an integrated model with "a few, but not necessarily all components" (*class E*) and some facets "not represented accurately" (*class 5*). Since the process model triggers the AMHS to execute transportation tasks the overall wafer fab model is classified as *case (d)* in *quadrant 4*—for explanation in detail see Jimenez et al. (2008).

The structure of the AMHS simulation model is illustrated in figure 3. As AutoMod supports hierarchical construction, there are three different systems/model levels: virtual model, sub model, and layout. In accordance to AutoMod language, "model" and "system" are hereinafter used synonymously.

The main part of the base model's code is located in the outlined "virtual model". This top level system contains collections of functions, methods, and procedures as well as global variables and constants. The

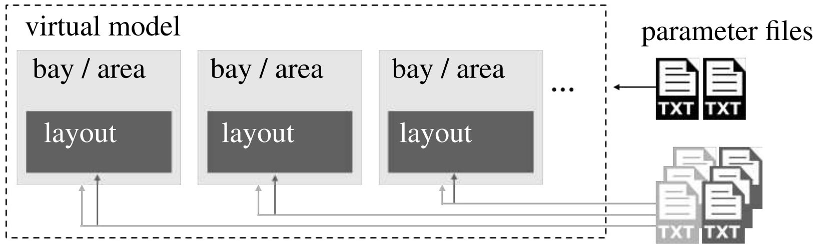

Figure 3: File and code structure of the simulation model

virtual model contains all basic functionality and main logic, e. g. reading input files, vehicle dispatching routines, congestion avoidance strategies, workload creation, or the basic processing and transportation routine. This concept allows a lean and easy way to maintain source code since all sub models (light gray colored) are embedded into the virtual model and inherit its mentioned functionality and logic. A sub model represents a logical or physical domain/territory, for instance a lithography bay, a wet chemistry bay, a inter-bay transportation system, or a stocker area. In order to e. g. run more precise experiments with certain logical parts of physical areas of the fab they may also be implemented as separate sub models. The simulation model can have as many sub models as necessary. Each sub model has furthermore the ability to locally overwrite the parents' source code to for example implement sophisticated routing strategies.

Usually each sub model holds another model (dark gray illustrated in figure 3) which represents and visualizes the layout in terms of transport system layout and tool layout. It is the physical representation of the fab, so for example rails, paths, tools, load ports, hoists, and so on.

As can be seen from figure 3, several text files are read by the program. The main reasons for extracting parameters from source code are speed (a simulation model does not have to be compiled between different parameter sets if they are read from external sources) and the intended adjustability of the model. The most important ones and their descriptions are listed in table 1. Each file contains a header, explaining its content, structure, and mandatory data.

Building a model and running simulations with the proposed base model is as simple as follows: (1) On sub model level, as many path mover systems as wanted need to be defined and (2) their corresponding track layout has to be drawn. For representing equipment (e. g. tools and stockers) (3) control points should be placed and if a visualization is desired (4) queues as representation of the equipments' load ports and the actual processing/handling unit might be placed. The connection of the different layout elements is made via an unambiguous nomenclature. To parametrize the model (5) input files have to be filled with data (see also table 1). In case of different logics to be implemented, (6) the source code needs to be extended at the right position. To ensure a high level of adjustability, usually all functionality which might get replaced or customized, is capsuled in functions.

The base model does not require AutoMod internal work lists (list of locations where vehicles look for workload) and park lists (list of locations where idle vehicles can park). By default the following behavior applies: If a vehicle drops a job/FOUP or is sent to a specific location it switches to idle mode and stays at its destination. If a vehicle blocks another one it moves to the next location on its current path. After a specific amount of movements (e. g. 5) without load a vehicle is sent to a park location (see table 1). Vehicles select their route in accordance with the AutoMod shortest path algorithm.

In contrast to the most suppliers provided simulation models the base model's general wafer fabrication procedure is rather simple. Without modifications or extensions it works as follows: According to the production schedule (see table 1) FOUPs are cloned to their initial location—which not necessarily needs to be the first process step. Thenceforward, they run through the rest of their product specific process steps and tools respectively. Jobs which need to be transported to their next location and vehicles with remaining load capacity are kept in lists—henceforward called *job list* and *vehicle list*. Each time a vehicle finishes

Table 1: Overview of files read by the virtual model and sub model

file name (virtual model)	parameter / content
Load_Input	• production schedule: times and venues the different products are initialized and their lot size
V_ProductProcessing	• product specific sequences of equipment types • necessary auxiliary parts (e. g. reticles, masks, carrier substrates, ...) • product specific tool processing times (distribution, mean, standard deviation) • tool and product specific hand-over-times (vehicle → load port, load port → tool and vice versa)
V_NoOfVehicles	• number of vehicles per sub model (dedicated vehicles are possible) • capacities of the vehicles
V_ActiveSystems	• systems/sub models which are initialized at simulation start • sub systems' stocker (reference to the stocker) and the hand over times • declare a system as an inter-bay transportation system

file name (sub model)	parameter / content
V_ToolQueueCapacity	• list of tools whose capacity differs from 1 • list of tool load ports whose number differs from 1 • capacity of stockers
V_VehParameter	• vehicle velocity for e. g. curves, straight, switch passing • vehicle acceleration (loaded and empty) • vehicle deceleration (loaded and empty)
V_VehParkLoc	• locations an idle vehicle explicitly is sent to
V_VehStartLoc	• location vehicles are beamed to at initialization of simulation

a transportation job, picks up a FOUP, or an equipment is done with its handling process, the subsequent algorithm is triggered.

If the number of items on both the job and the vehicle list is greater than one, then run through the job list and find a job whose next equipment has remaining capacity—if allowed, also look for an alternative equipment. If no appropriate equipment can be found, the job is sent to storage (e. g. a stocker) or stays at the current location if the storage is full. If an appropriate equipment is found, it is checked if it is a batch tool and the job has to wait in a storage until the batch size is reached. In all other cases the nearest available vehicle on the vehicle list is ordered to handle the delivery.

If the fab layout is a segregated one (see figure 2) and the found equipment is located in a different bay, the delivery process is executed as follows: The job is sent to a location which allows transitions between an inter-bay and an intra-bay transportation system (usually a stocker). A vehicle of the inter-bay systems then delivers the job to the transition location of the destination bay where a third vehicle finishes the transport.

Moreover, jobs are prioritized in the base model by their initial times. This provides a running model without changes but as all other functions, it may be substituted by more complex rules.

4 USE CASE DEMONSTRATION

Due to confidentiality reasons, the following use case cannot be described in detail. Object of study is the fab of Infineon Dresden and the production of a specific product with 106 manufacturing steps in sum. The processing route ranges from single of a kind tools to 42 redundant tools. The system's layout is a segregated one involving 10 different bays. A simplified layout with all activated bays is illustrated in figure 4 (the given number of tools are arbitrary and do not reflect the actual number). The number of active vehicles per bay varies from 1 to 15 and is stated in figure 4. The inter-bay system contains 30 vehicles. All vehicles have a transportation capacity of 1 FOUP—apart from the ones on the inter-bay system with a capacity of 2 FOUPs. The workload varies per experiment.

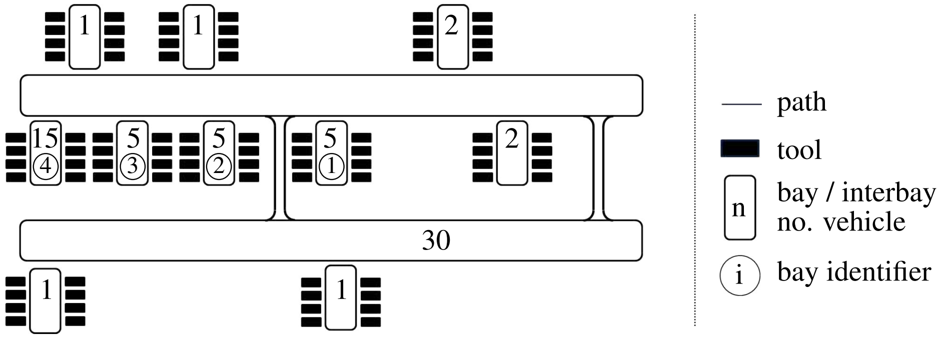

Figure 4: Layout of the demonstrator and use case, respectively

On one hand, a slightly modified simulation model provided by the AMHS supplier Muratec ("Murata Machinery, Ltd."—henceforward denoted as Muratec model) as a reference is applied. On the other hand, the same system is modeled from scratch on basis of the base model presented in section 3 (henceforward denoted as our model). Building and initially parametrizing our model took the authors approx. 10 hours. Although the two models basically represent the same system, they differ in their level of detail. The most important aspects are the following ones.

In contrast to the base model, the Muratec model implements a self-made routing algorithm which bases on a detailed from-to matrix. In this connection, the Muratec model also uses a mixture from AutoMod internal work lists and park lists and a quite specific self-made dispatching logic on top. Furthermore Muratec implements different traffic jam avoidance strategies which are partly based on cost factors. The Muratec model accurately performs the process of handling FOUPs and corresponding vehicles at load ports. This logic is detailed to an extent that for each handover process the subsequent vehicle's distance is checked and in case of being less than 7 m, some alternative strategy applies.

All in all, our model manages the whole fabrication processes with roughly a third of the code of the Muratec model which is a strong indication of a higher level of abstraction used for the base model.

The Muratec model is considered to be valid. Its simulation results are in strong accordance with real world measured data. Hence in section 5, the result accuracy of our model is assessed with respect to the outcomes of the Muratec model.

5 USE CASE RESULTS

All experiments have been conducted on a 64 bit Windows 7 machine with an Intel i5 2500k processor and 8 GB of RAM. The AutoMod version used was 12.4 build 23.

Apart from intrinsic model differences, the Muratec model and our model are initialized with identical parameters and workload (see section 4). To avoid confusion, only results based on a particular workload are presented in this article. There are no additional major findings while varying the workload. The warm-up phase was a quarter of the tracking period and is excluded from results. The tracking period's length was chosen "appropriately" and has been constant for each simulation run. Each model has been simulated 10 times with different seeds. Furthermore, the evaluation will concentrate on bay ①, ②, ③ and ④ (see figure 4) because the Muratec model does not allow deeper analysis of the rest of the bays.

A sloppy implementation in the Muratec model leads to incorrect counting of some transports. All subsequently stated values are based on the corrected results.

Table 2 shows the overall finished transportation cycles executed by the active vehicles. Therein are included tool to tool transports, stocker to tool transports and vice versa. Stated in table 2 are the lower and upper limits of the results of the different initializations.

Table 2: Transportation jobs completed—interval given by 10 seeds

	bay ①	bay ②	bay ③	bay ④	bay ① to ④
Muratec model	[957, 995]	[1 368, 1 565]	[1 129, 1 230]	[6 462, 6 945]	[9 977, 10 651]
our model	[1 074, 1 151]	[1 344, 1 432]	[1 235, 1 258]	[6 997, 7 132]	[10 718, 10 895]

From table 2, it becomes obvious that regarding the total finished transportation jobs both models show nearly the same results. It should be emphasized again that the base model and hence our model has implemented only very basic control strategies (see chapter 3) in contrast to the Muratec model.

Table 3 gives the average number of finished transportation cycles and the coefficient of variation (COV) on basis of the 10 seeds. It allows some deeper research to which extent the models differ in their transportation load.

Table 3: Transportation jobs completed—mean (and coefficient of variation) from 10 seeds

	bay ①	bay ②	bay ③	bay ④	bay ① to ④
Muratec model	981 (0.001)	1 499 (0.030)	1 186 (0.030)	6 867 (0.020)	10 534 (0.018)
our model	1 115 (0.020)	1 377 (0.020)	1 245 (0.007)	7 073 (0.005)	10 810 (0.005)
deviation (basis: Muratec model)	14 %	8 %	5 %	3 %	3 %

First of all, all COV are found to be ≤ 0.03. So the results of the different simulation runs only differ slightly which suggests robust outcomes. Thus, an evaluation of the particular mean values is natural. Across all four bays the deviation is only 3 %. The maximum relative deviation between the Muratec and our model can be seen at bay ①, the lowest deviation at bay ④. It is worth mentioning that the results are strongly indicating a connection between the number of tools bundled in a bay and outcome accuracy. In bay ① basically only 3 equipments are employed whereas 17 tools are active in bay ④.

A small downside of the base model can be seen in figure 5. Therein one simulation run was randomly picked and the histogram of the hourly finished jobs of bay ② and bay ③ are shown. It can be seen that the distribution of the hourly finished transportation jobs differs a little. Especially for bay ③ of our model the shape of the histogram varies and the density is shifted to higher values.

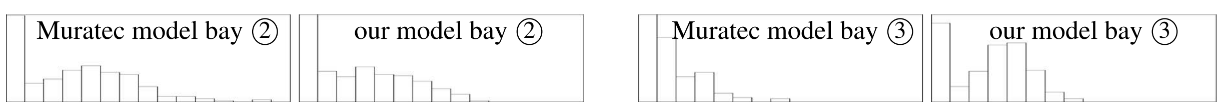

Figure 5: Histogram of hourly finished transportation jobs—x-axis: transports per hour; y-axis: frequency

Nevertheless, the main conclusion one can draw from the gathered results is the ability of the newly introduced base model predicting transportation workload of wafer fabs quite well, even though there are minor differences at hourly level.

In order to evaluate the performance of the base model in connection with forecasting equipment behavior, 3 tools of an arbitrary seed are randomly chosen and their received FOUPs and utilization are compared—table 4 illustrates the results.

According to table 4, the highest relative deviation is observed at tool 3—the lowest values 5 % are detected for tool 1 and 2. The order of difference between the Muratec model and our model concerning the equipment behavior is approx. the same as of the results of the transportation load (see table 2 and 3).

Additionally, in two other aspects, the models' outcomes resemble one another: Firstly, when searching for bottlenecks, the same tools come into focus—what is basically obvious because of identical equipment

Table 4: Jobs received to process by equipment—total (and utilization)

	tool 1	tool 2	tool 3
Muratec model	342 (0.625)	501 (0.281)	801 (0.448)
our model	359 (0.662)	525 (0.292)	722 (0.402)
deviation (basis: Muratec model)	5 %	5 %	10 %

sequences of the wafers. Even more remarkable is that these tools' maximum utilizations are 0.82 in our model and 0.86 in the Muratec model. Secondly, both models reach their maximum utilization level at the same workload level. So our model is able to reliably predict (the maximum) utilizations and hence the (maximum) throughput. Table 5 reveals some further investigations in this respect. Results shown are the mean cycle times and absolute throughputs of a simulation series. Similar to previous results our model and the Muratec model produce outcomes which only vary minorly: The observed cycle time's difference is 1%, the deviation of throughput is 5%.

Table 5: Cycle time and throughput—normalized to 1 (basis Muratec model) due to confidentiality reasons

	cycle time	throughput
Muratec model	1.00	1.00
our model	0.99	1.05

As mentioned in the introductory paragraphs, runtime performance of simulation models is a well-known issue. Besides adjustability the herein presented base model addresses this topic. To give an idea of its computational performance, the CPU time of simulation runs with different workloads have been measured and compared. The findings based on our and the Muratec model are shown in table 6.

Table 6: Computational performance of one simulation run—CPU time measured in seconds

	workload ("wafer starts per week")			
	low	medium	high	very high
Muratec model	1 896 s	4 353 s	6 138 s	6 487 s
our model	332 s	577 s	648 s	729 s
ratio (Muratec / our model)	5.7	7.5	9.5	8.9

From table 6 it becomes obvious that our model is always faster than the Muratec counterpart. Time savings go up to nearly 10 times the CPU time. Additionally, in either case apparently runtimes correlate with the number of wafer starts per week—the higher the workload the slower the model. From the particular results this relationship was found to be less impactful in case of the proposed model.

In order to further increase the computational performance, some modifications at the base model have been evaluated. The findings have not been tested in detail. Hence quantitative assessments can not be given. Nevertheless, defining proper park locations for vehicles has a strong impact on performance. This is due to the fact that in AutoMod vehicle activity is generally expensive. In this connection no blocks (regions with limited vehicle capacity to avoid for example collisions at intersections) have been placed. Thereby, some minor CPU time improvements could be observed with the admission of rarely (approx. 5 per simulation run) appearing collision warnings of AutoMod. Furthermore some studies report significant runtime enhancements when not modeling an integrated model but separately create a transportation system and an equipment system and couple them somehow (e. g. Jimenez et al. 2008). This idea has also been discussed in an early phase of the base model. It was discarded due to dramatically poor results accuracy.

6 CONCLUSION AND OUTLOOK

We introduced a novel base model from scratch for building, adjusting, and simulating automated material handling systems with a path based transportation system. Compared to the most suppliers provided models, for the base model a higher level of abstraction was chosen. In a use case of Infineon's Dresden wafer fab we could show that the base model produces results with an accuracy in the range of results obtained from highly detailed and validated model. However, the base model is up to 10 times faster in terms of CPU time and only needs a fraction of time for establishing a particular model.

The base model is highly adjustable due to extracting parameters into files which get read by the program. Additionally the encapsulation of functionality (e. g. dispatching strategy) allows an easy replacement and extension. Hence, it is supplier independent and applicable to a wide range of systems including segregated track layouts as well as unified ones or even automated guided vehicle systems as the differences may be incorporated into the overwritable functions and parameters.

Based on the findings so far, it is hard to assess if the base model performs equal for other wafer fabs. Especially for fabs following a foundry concept, more research is needed because their workload and control strategies differ from integrated device manufacturers—like presented in section 4—or memory fabs (Tung et al. 2013).

Furthermore, we will continue to find the *right* level of detail in order to either get more accurate results or improve runtime. Nevertheless, the presented study reinforces the theory of losing performance if the model complexity is too high (see section 2.2). A descriptive example is the Muratec model's traffic jam avoidance strategy. It is always active although, most bays' track layout are unsuitable to avoid jams at all.

As mentioned in section 2.1 there are also efforts ongoing to implement the herein presented base model into a framework to automatically create simulation runs.

ACKNOWLEDGMENT

The work has been performed in the project EPPL (`www.eppl-project.eu`), co-funded by grants from Austria, Germany, The Netherlands, Italy, France, Portugal – ENIAC member states and the ENIAC Joint Undertaking.

REFERENCES

Adami, C. 2002. "What is Complexity?". *BioEssays* 24 (12): 1085–1094.

Agrawal, G., and S. Heragu. 2006. "A Survey of Automated Material Handling Systems in 300-mm SemiconductorFabs". *IEEE Transactions on Semiconductor Manufacturing* 19 (1): 112–120.

Arnold, D., and K. Furmans. 2009. *Materialfluss in Logistiksystemen*. Berlin, Heidelberg: Springer.

Arzt, T., and F. Bulcke. 1999. "A New Low Cost Approach in 200 mm and 300 mm AMHS". *Semiconductor Fabtech* 10:19–26.

Astrup, R., K. D. Coates, and E. Hall. 2008. "Finding the Appropriate Level of Complexity for a Simulation Model: An Example With a Forest Growth Model". *Forest Ecology and Management* 256 (10): 1659–1665.

Benjamin, P., M. Erraguntla, D. Delen, and R. Mayer. 1998. "Simulation Modeling at Multiple Levels of Abstraction". In *Proceedings of the Winter Simulation Conference*, edited by D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannon, 391–398.

Brain, M., and P. Lin. 2007. "Impact of Speed on the Performance of Vehicle-free Transport". In *International Symposium on Semiconductor Manufacturing, 2007. ISSM 2007*, 1–3.

Chwif, L., M. Barretto, and R. Paul. 2000. "On Simulation Model Complexity". In *Proceedings of the Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 449–455. Orlando.

Fonseca, D. J., D. O. Navaresse, and G. P. Moynihan. 2003. "Simulation Metamodeling Through Artificial Neural Networks". *Engineering Applications of Artificial Intelligence* 16 (3): 177–183.

Gartland, K. 1999. *Automated Material Handling System (AMHS) Framework User Requirements Document: Version 1.0*, Volume Technology Transfer # 99073793A-TR. Austin: International SEMATECH.

Hallenborg, K., and Y. Demazeau. 2008. "DECIDE: Applying Multi-agent Design and Decision Logic to a Baggage Handling System". In *Engineering Environment-Mediated Multi-Agent Systems*, edited by D. Weyns, S. A. Brueckner, and Y. Demazeau, Number 5049, 148–165. Berlin, Heidelberg: Springer.

Hammel, C., T. Schmidt, and M. Schöps. 2012. "Network Optimization Prior to Dynamic Simulation of AMHS". In *Proceedings of the Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 172:1–172:11. Berlin.

Han, Y., D. Park, S. Chae, and C. Lee. 2005. "Full Fabrication Simulation of 300mm Wafer Focused on AMHS (Automated Material Handling Systems)". In *Systems Modeling and Simulation: Theory and Applications*, edited by D.-K. Baik, Number 3398, 514–520. Berlin, Heidelberg: Springer.

Jimenez, J., G. Mackulak, and J. Fowler. 2008. "Levels of Capacity and Material Handling System Modeling for Factory Integration Decision Making in Semiconductor Wafer Fabs". *IEEE Transactions on Semiconductor Manufacturing* 21 (4): 600–613.

Kiba, J.-E., G. Lamiable, S. Dauzre-Prs, and C. Yugma. 2009. "Simulation of a Full 300 MM Semiconductor Manufacturing Plant with Material Handling Constraints". In *Proceedings of the Winter Simulation Conference*, edited by M. D. Rossetti, R. Hill, and B. Johansson, 1601–1609. Austin, Texas.

Kim, B.-I., S. Jeong, J. Shin, and J. Koo. 2009. "A Layout- and Data-Driven Generic Simulation Model for Semiconductor Fabs". *IEEE Transactions on Semiconductor Manufacturing* 22 (2): 225–231.

Kuhn, A., A. Wagenitz, and K. Klingebiel. 2010. "Praxis Materialflusssimulation". In *Jahrbuch der Logistik 2010*, edited by H. Wolf-Kluthausen, 206–211. Korschenbroich: free beratung.

Law, A. M., and D. W. Kelton. 2000. *Simulation Modeling and Analysis*. Singapore: Mcgraw-Hill.

Lim, D.-E., and M. Seo. 2014. "A Generic Simulation Framework for Efficient Simulation Analyses for Semiconductor Manufacturing". *International Journal of Control & Automation* 7 (2): 75–84.

Lin, J. T., and C.-J. Huang. 2014. "A Simulation-based Optimization Approach for a Semiconductor Photobay with Automated Material Handling System". *Simulation Modelling Practice and Theory* 46:76–100.

Mackulak, G. T., F. P. Lawrence, and T. Colvin. 1998. "Effective Simulation Model Reuse". In *Proceedings of the Winter Simulation Conference*, 979–984. Los Alamitos, CA, USA.

Montoya-Torres, J. R. 2006. "A Literature Survey on the Design Approaches and Operational Issues of Automated Wafer-transport Systems for Wafer Fabs". *Production Planning & Control* 17 (7): 648–663.

Muller, D. 2013. "AutoMod: Modeling Complex Manufacturing, Distribution, and Logisitics Systems for over 30 Years". In *Proceedings of the Winter Simulation Conference*, 4037–4051. Piscataway, NJ, USA.

Nadoli, G., and D. Pillai. 1994. "Simulation in Automated Material Handling Systems Design for Semiconductor Manufacturing". In *Proceedings of the Winter Simulation Conference*, 892–899. Orlando.

Nazzal, D., and D. Bodner. 2003. "A Simulation-based Design Framework for Automated Material Handling Systems in 300 mm Fabrication Facilities". In *Proceedings of the Winter Simulation Conference*, edited by P. J. Sanchez and S. Chick, 1351–1359. New Orleans.

Nazzal, D., and L. McGinnis. 2007. "Expected Response Times for Closed-Loop Multivehicle AMHS". *IEEE Transactions on Automation Science and Engineering* 4 (4): 533–542.

Peng, C., F. Erhun, and E. F. Hertzler. 2012. "Capacity Planning in the Semiconductor Industry: Dual-Mode Procurement with Options". *Manufacturing & Service Operations Management* 14 (2): 170–185.

Pillai, D., E. Bass, J. Dempsey, and E. Yellig. 2004. "300-mm Full-factory Simulations for 90- and 65-nm IC manufacturing". *IEEE Transactions on Semiconductor Manufacturing* 17 (3): 292–298.

Qin, W., J. Zhang, and Y. Sun. 2013. "Multiple-objective Scheduling for Interbay AMHS by Using Genetic-programming-based Composite Dispatching Rules Generator". *Computers in Industry* 64 (6): 694–707.

Robinson, S. 2007. "Conceptual Modelling for Simulation Part I: Definition and Requirements". *Journal of the Operational Research Society* 59 (3): 278–290.

Schulz, M., and T. D. Stanley. 2000. "Simulation Based Decision Support For Future 300mm Automated Material Handling". In *Proceedings of the Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1518 – 1522. Orlando.

Sommerville, I. 2010. *Software Engineering*. 9th edition ed. Boston: Addison-Wesley.

Swain, J. J. 2013. "Simulation Software Survey Simulation: A Better Reality?". *OR/MS-Today* 40 (5): na.

Terwiesch, C., and R. E. Bohn. 2001. "Learning and Process Improvement During Production Ramp-up". *International Journal of Production Economics* 70 (1): 1–19.

Thesen, A., and L. E. Travis. 1989. "Simulation for Decision Making: An Introduction". In *Proceedings of the Winter Simulation Conference*, edited by E. A. MacNair, K. J. Musselman, and P. Heidelberger, 9–18. New York: ACM.

Tung, J., T. Sheen, M. Kao, and C. H. Chen. 2013. "Optimization of AMHS Design for a Semiconductor Foundry Fab by Using Simulation Modeling". In *Proceedings of the Winter Simulation Conference*, 3829–3839. Piscataway, NJ, USA.

Tyan, J. C., T. C. Du, J. C. Chen, and I. H. Chang. 2004. "Multiple Response Optimization in a Fully Automated FAB". *Computers & Industrial Engineering* 46 (1): 121–139.

VDI 3633 2010. *Simulation von Logistik-, Materialfluß- und Produktionssystemen*. Berlin: Beuth Verlag.

Wagner, T., A. Gellrich, C. Schwenke, K. Kabitzsch, and G. Schneider. 2014. "Automated Planning and Creation of Simulation Experiments with a Domain Specific Ontology for Semiconductor Manufacturing AMHS". In *Proceedings of the Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 2628–2639. Piscataway, NJ, USA.

AUTHOR BIOGRAPHIES

SEBASTIAN RANK is a research assistant at Technische Universität Dresden. He is member of the scientific staff at the chair of Technische Logistik (Logistics Engineering). Having received his Diploma (similar to M. S. degree) in Economics and Engineering (Wirtschaftsingenieurwesen) from Technische Universität Dresden in 2011 his research interests include applications of AMHS simulation as well as autocorrelation in logistics systems. His email address is sebastian.rank@tu-dresden.de.

CHRISTIAN HAMMEL is a research assistant at Technische Universität Dresden. He is member of the scientific staff at the chair of Technische Logistik (Logistics Engineering). Having received his Diploma (similar to M. S. degree) in Applied Mathematics (Technomathematik) in 2007 from Technische Universität Dresden his research interests include the application of analytical methods to optimize existing AMHS as well as to support the design phase. His email address is christian.hammel@tu-dresden.de.

GERMAR SCHNEIDER is a manager at Infineon Technologies Dresden GmbH and in charge of "Thin Wafer Handling". He is project leader for automation projects within the Eniac-JU funding project EPPL. He received a Diploma and a PhD in analytical chemistry and has worked for 18 years in different positions at Infineon Technologies. In the recent years he filled management positions in processing, manufacturing, and maintenance departments and was head of an automation and factory integration group. His email address is germar.schneider@infineon.com.

THORSTEN SCHMIDT heads the chair of Technische Logistik (Logistics Engineering) at the Technische Universität Dresden. He holds degrees in Mechanical Engineering from Technische Universität Dortmund and Industrial Engineering from the Georgia Institute of Technology. His research interests focus on machinery and design of facility logistics and production systems. His email address is thorsten.schmidt@tu-dresden.de.