Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

AUTOMATED GENERATION OF ANALYTICAL PROCESS TIME MODELS FOR CLUSTER TOOLS IN SEMICONDUCTOR MANUFACTURING

Robert Kohn
Oliver Rose

Dresden University of Technology
Institute of Applied Computer Science
D-01062, Dresden, Saxony, GERMANY

ABSTRACT

In this paper, we present an approach to automatically create an analytical process time model for cluster tools using real-world data. The proposed model combines advantages of simple throughput models and discrete event simulation models. We consider the effect of small lot size and the slow down effect occurring when simultaneously processed lots interfere with each other. Especially the use of Slow Down Factors depending on a certain recipe combination and start delay adequately mirrors sequential and parallel processing mode. We also describe a modeling method that automatically leads to parameterized models with high accuracy. This study presents evaluation results gained from models, which we create from and test against real-world data gathered from past equipment events. We discuss exemplary processing behaviors by means of three examples. We conclude that the proposed analytical cluster tool model is suitable to predict process times with respect to accuracy and prediction coverage.

1 INTRODUCTION

Semiconductor manufacturers intensified exploring new approaches in the field of operational resource scheduling, which is supposed to replace dispatching systems as state of the art. The application of lot scheduling on the shop floor, powered by optimization techniques, promises remarkable improvements in reducing lot cycle times as well as increasing lot due date compliance. Lot scheduling solutions require equipment models, which enable the simulation of lot schedules and thus their evaluation with respect to targeted objectives.

Modeling experts especially focus on cluster tools because of their wide spread use in manufacturing and their complex processing behavior. In practice we either face complex models based on discrete event simulation (DES) or those easier throughput models relying on simple formulae and bottleneck conditions. But both approaches do not fit the needs of operational scheduling. DES models perform too slow and simple throughput models lack the necessary details. The analytical equipment model seems to be a good trade-off between accuracy and speed. Regardless the underlying modeling approach, creating equipment models is very time consuming, especially when real-world data serves to parameterize the models.

In this article, we present an analytical process time model for a representative architecture of cluster tools as well as a method that automatically leads to parameterized equipment models using real-world data. The idea of automated modeling considerably saves time, but requires adequate outlier handling procedures. The described equipment model accounts for the effect of lot size as well as for slow down effects, which occur when simultaneously processed lots interfere with each other. We use the resulting model to predict processing times and completion dates, but not the makespan. We describe observed processing behaviors with the aid of three exemplary pieces of etching-equipment. As a result of this

study, we make a statement on how the suggested model/method is suitable for equipment simulation, especially with respect to accuracy of process time predictions. This study aims to lay the foundations for practically simulating cluster tools by use of an analytical model under real-world conditions.

1.1 Cluster Tool Description

As depicted in Figure 1, a cluster tool effectively combines a number of process chambers to a single machine, basically composed of mainframe and attached Equipment Front End Module (EFEM). The mainframe consists of a central wafer handling robot connected to the process chambers and a number of loadlocks offering access to the mainframe. The EFEM comprises a number of loadports and a wafer handler, which enables wafer transfer between loadports and loadlocks.

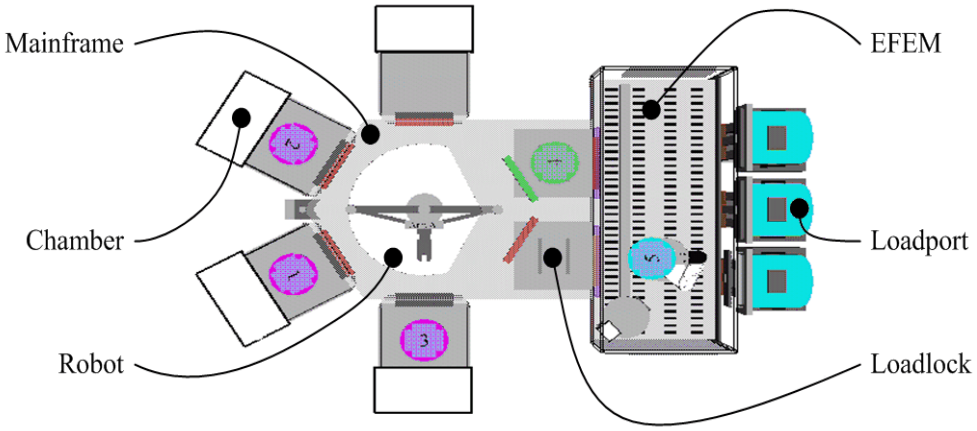

Figure 1: Cluster tool components

The procedure performed to process a lot follows a defined sequence of activities. After the carrier is put on the loadport, the EFEM's handler sequentially transfers the wafers in atmosphere to the loadlock, which then pumps to vacuum. Thereafter the mainframe's handling robot transports one wafer after another to a free process chamber. According to the recipe, a wafer possibly visits more than one chamber before it eventually returns to the loadlock. At the time, when the last wafer of a lot returns to the loadlock, the loadlock vents to atmosphere. Lastly the EFEM's handler transfers processed wafers back to the carrier. Finally the carrier is removed from the loadport in order to continue with next operation.

Obviously cluster tools were made to process lots in parallel, since they comprise multiple loadlocks and several chambers, which may offer different processes. But strictly speaking, the internal processing mode, either sequential or parallel, is firstly determined by the combination of lot's recipes performed and secondly depending on the internal wafer scheduling policy (see subsection 2.4). The recipe defines the internal wafer routing, meaning the sequence of specific chambers to visit. Different recipes commonly represent different process setups, either manifesting in different wafer routing sequences or at least in the same wafer routing sequences with differing chamber process times.

However, in the case of an unequal set of process chambers assigned for processing lots in parallel, the equipment operates in parallel mode in order to prevent idle chambers. In contrast, in the case of equal sets of process chambers lots competing for, the process mode is determined by the internal wafer scheduling policy applied. A fair scheduling policy, treating available wafers (from different lots) the same and therefore mixing them up during processing leads to parallel processing mode. In contradiction an unfair scheduling policy, commonly preferring the first available wafers, leads to sequential processing mode. We summarize that a cluster tool performs in sequential mode if the set of chambers (to be visited by competing lots) is the same and the scheduler treats the lots unfairly, otherwise in parallel mode.

Since we deal with real-world data, we focus on equipment events immediately beamed to the Manufacturing Execution System (MES) when fired. For our analysis aimed at producing results to parameter-

ize the model, we focus on the Raw Tool Time (RTT) of processed lots. The RTT is commonly defined as the time between start of processing the first wafer of a lot until finish of the processing of the last wafer. The pieces of equipment chosen in this article as examples, fire the RTT starting event before pumping the loadlock and the RTT finishing event after venting the loadlock (see Figure 2).

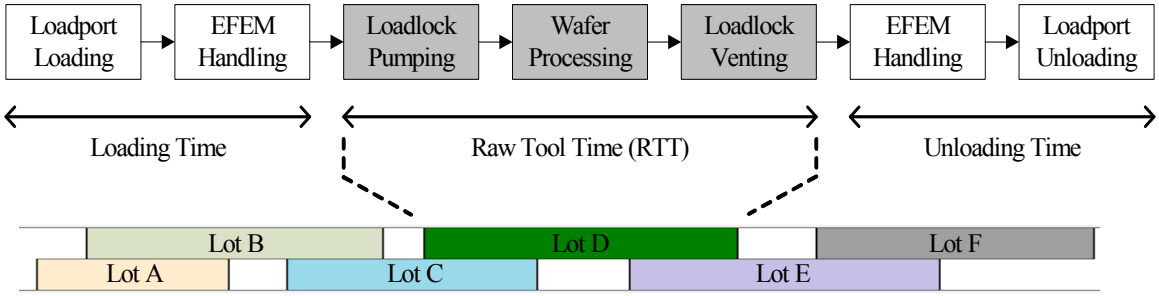

Figure 2: Raw tool time definition

1.2 Related Work

The origin of analytical cluster tool models is traceable to Wood (see Wood, Tripathi, and Moghadam (1994) and Wood (1996)), who first introduced an analytical modeling approach used for throughput modeling and process time estimations. Wood proposed a set of formulae and bottleneck conditions, which describe the cluster tool's throughput. Wood considered internal wafer handling and chamber processing times as well as different processing modes. Almost at the same time Perkinson (1996) presented an analytical cluster tool modeling approach, similar to Wood, used for throughput estimations.

Schmidt, Weigang, and Rose (2006) and Hosoe, Knanamori, and Yoshida (2007) also used analytical models for process time estimation, particularly focusing on the effect of lot size. Both identified and classified common lot processing behaviors with regard to lot size. In the field of automated modeling, Lange et al. (2008) published an approach for automated generation of equipment throughput models by analyzing internal equipment events, basically combining the approaches of Wood and Perkinson. We formerly presented a method in Kohn, Werner, and Rose (2010) to automatically estimate capacity-related equipment parameters as well as a process time model, taking the effect of lot sizes into account. We presented first thoughts on automated modeling based on the analysis of MES events.

According to the idea of using Slow Down Factors mirroring dynamic interrelationships inside a cluster tool, Niedermeyer and Rose (2003, 2004) presented a simulation-based analysis of process times. They observed the influence of recipe combinations and the impact of start delays, while stating first thoughts on how to use Slow Down Factors for cluster tool modeling. In a following effort, Unbehaun and Rose (2006, 2007) continued developing the idea (of using slow down factors) and presented a model as well as a method to predict process times at cluster tools. Both efforts using Slow Down Factors evaluated their results by use of DES models mirroring the ideal equipment behavior in real world.

2 RAW TOOL TIME MODELING

In this section, we present the analytical model used for predicting the RTT. Beside the model structure and its components, we describe the modeling steps that lead to an automatically parameterized model derived from real-world data. We picked up already presented approaches (see subsection 1.2) and modified the ideas; some were simplified, some were evolved. Simplifications and extensions both enter into a new RTT model. The resulting model enables RTT prediction in combination with the prediction method described in section 3. We assume that the RTT mainly depends on the performed recipe R. Thus the model relies on the default Raw Tool Time (RTT_{DEF}), which is then adjusted by additional factors according to process characteristics. The proposed model takes the effect of lot size (LS) into consideration, since small lot size results in reduced RTT. Therefore we introduce a Lot Size Factor (LSF) depending on the

lot size. In addition to that, the model takes account of possibly extended lot's RTT processed in parallel mode, referred to as slow down effect. The slow down effect occurs when lots processed in parallel compete for internal resources at the same time and consequently slow each other down. As a consequence the RTT of each lot is extended, while the overall makespan is reduced. Regarding that effect, we involve a Slow Down Factor (SDF) that depends on the recipe combination RC of simultaneously processed lots as well as on the start delay SD, arising when overlapping lots start consecutively. We predict the RTT using the following formula:

$$RTT(R, LS, RC, SD) = RTT_{DEF}(R) \times LSF(LS) \times SDF(RC, SD)$$

with the following parameters:

- The default Raw Tool Time RTT_{DEF} serves as reference without any disturbing impacts, only depending on the specific recipe R, which defines the process and consequently the wafer routing and chamber processing times.
- The Lot Size Factor LSF represents a corrective factor depending on the lot size, given in number of wafers. The LSF describes how much faster a process is performed in the case of small lot size. Thus the LSF ranges between a value higher than zero and one in the case of standard lot size.
- The Slow Down Factor (SDF) represents the already mentioned slow down effect. The SDF depends on the recipe combination (RC) as well as on the start delay (SD) and ranges between one and two.

2.1 Default Raw Tool Time Estimation

We use the RTT_{DEF} as the model's reference value that is independent from lot size and slow down effects, only depending on a certain recipe. The value RTT_{DEF} states an adequate estimation for the RTT, when both the lot size factor and the slow down factor were set to default, respectively to one. For the automated value estimation, we focus on historical jobs with maximal lot size and processed in single mode, without any other lots in parallel. We basically summarize jobs processed under equal conditions, which then lead after analysis to a value for the RTT_{DEF} for each recipe separately (see **Error! Reference source not found.**). Therefore jobs in history are collected, adequately filtered and categorized and then compressed by use of simple statistics. We describe the steps to proceed in the following.

Table 1: Default Raw Tool Times (example)

Recipe	RTT_{DEF} [min]
Recipe_A	15
Recipe_B	20
Recipe_C	30

1. **Job Filter** - In order to provide historical jobs for analysis, which meet the above mentioned conditions regarding lot size and process mode, the collected jobs are filtered. Jobs are filtered out, if they do not have the maximal lot size or were not processed with exclusive access to the machine. As a result there exists a subset of observed jobs suitable to represent the processing of a certain recipe without any disturbing interference.
2. **Job Categorization** - We assume that the recipe implies the course of visiting process chamber inside the tool, and consequently the resulting RTT. Since the RTT firstly depends on the lot's recipe, the remaining partial quantity of jobs is then categorized according to the recipe executed.

As a result we obtain groups of jobs, where each group is uniquely assigned to a certain recipe processed in history.

3. **RTT_{DEF} Calculation** - After filtering and categorizing the jobs, we assume that each group contains jobs processed under same conditions. In spite of that, there still exist differences in observed RTT within a group of jobs due to captured process failures or other exceptions. In order to decrease the influence of these outliers, the statistical median of observed RTT is calculated within each group, which finally represents the RTT_{DEF}.

2.2 Lot Size Factor Estimation

We use the LSF to approximate the effect of small lot size causing a shorter RTT, since the cluster tool separately processes the wafers in a single chamber each. In contrast to most approaches in literature that comprise multiple internal equipment parameter, for example internal wafer handling times and processing intervals (see subsection 1.2), we propose a more general approach. We use the LSF as a function of the lot size. The LSF ranges between zero and maximal one at default, in the case of carriers utilized to capacity. For LSF estimation we focus on jobs solely processed in single mode, because they were not accompanied by slow down effects. We state that the relationship between lot size and corresponding LSF is almost linear with offset for the focused type of cluster tool. The proposed estimation method handles outlier as well as the difficulty of possibly non-continuous data points, where the LSF function is automatically derived from. We describe a sequence of steps that automatically lead to a reasonable LSF function representing the elementary behavior of cluster tools with regard to lot size.

1. **Job Filter** - First the collected jobs are filtered to create a subset of jobs that is suitable to clearly represent the lot size effect for analysis. We focus on jobs that are not associated with the slow down effect. Therefore we exclude jobs that were not processed in single mode, and respectively show an overlap in RTT with another lot processed in parallel. The remaining set of jobs provides records of RTTs that exclusively depend on the recipe performed as well as on the lot size. Other influences, for example the slow down effect, are assumed to be neglected due filtering.

2. **Job Categorization** - In order to prepare a data set suitable for automated estimation of the LSF function, the remaining jobs are categorized by their lot size. Therefore jobs with equal lot size were grouped together. Consequently subsets of jobs emerge, where each subset embodies the processing of the same number of wafers.

3. **LSF Calculation** - After filtering and categorizing jobs collected from history, a representative LSF is calculated for each lot size specific subset of jobs. Since the jobs of each subset probably refer to different recipes, their RTT values differ, too. In order to neglect the effect of the recipe and to make the job's RTT comparable, the value of RTT is normalized. For that reason, each RTT value is divided by the previous calculated RTT_{DEF} (see subsection 2.1). Note, that the calculated quotients now represent the LSF assumed to exclusively depend on the lot size.

   We observed that despite filtering, categorizing and normalizing the values, statistical post-processing is necessary due to partly widely varying resulting LSFs per subset. This is why the statistical median of LSFs is calculated within each subset, reducing the influence of outliers and compressing the samples to a single LSF value for each lot size, as far as observed in the data set. The resulting LSFs range between a value higher than zero and maximal one. A LSF value higher than one indicates an erroneous model. Finally a data set is created, where each data point represents a LSF value related to a specific lot size.

4. **Interpolation and Outlier Removal** - The resulting data set is then used to estimate a functional relationship between LSF and lot size. In this context, two challenges impede the estimation of a reasonable lot size function. To begin with, the data set does not continuously provide LSF samples for each lot size, because the observed jobs do not cover the whole range of possible lot siz-

es. Furthermore, there are still remaining outliers, which may falsify the estimated function or even prevent a reasonable estimation of the lot size function, at worst. Therefore a multi-step estimation method is cyclically applied, where each cycle consists of three steps:

(a) The samples are linearly interpolated in order to handle missing samples for some lot sizes.

(b) The resulting interpolation is smoothed with moderate intensity. The smoothing effect is achieved by reassigning each value with the average of its neighboring values.

(c) The most distant data point compared to previously smoothed interpolant is removed.

The algorithm stops if the most distant point is within an acceptable range near the interpolation. Otherwise the algorithm continues with a new cycle, interpolating the remaining data points. This method leads to a reasonable LSF function most of the time, while effectively reducing the influence of outliers. We admit that a lack of data samples and a high number of outlier prevent a successful reconstruction of a reasonable LSF function. Figure 3 depicts the resulting interpolations at the end of the first three cycles.

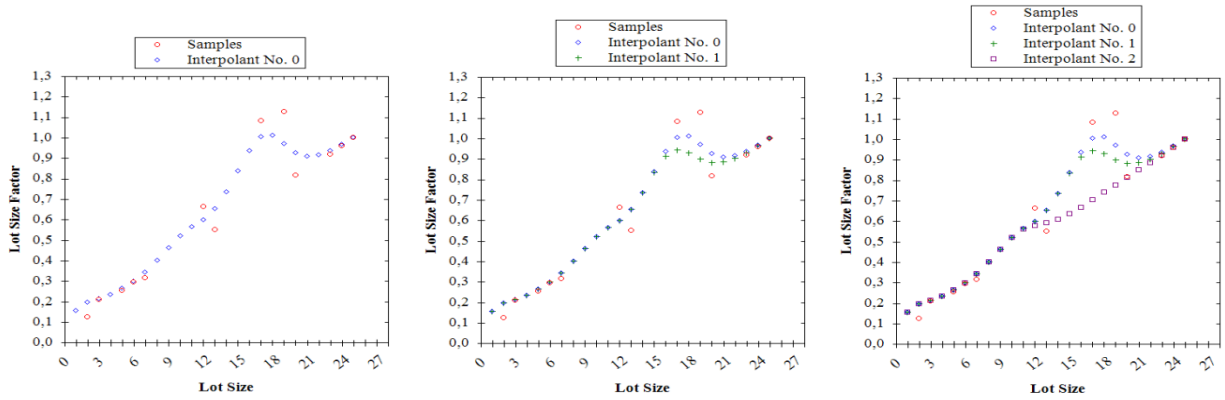

Figure 3: LSF interpolation and outlier removal

2.3 Slow Down Factor Estimation

We use the SDF to reflect the dynamic interrelationship between simultaneously processed jobs. Jobs processed in parallel possibly slow each other down, when competing for the equipment's internal resources, like wafer processing chambers or robots. The proposed concept of using SDFs involves two aspects of processing. On the one hand, a certain combination of lots, more exactly the combination of their recipes, determines the SDF in a specific situation (Niedermeyer and Rose 2003, 2004). On the other hand, the start delay between consecutively started lots impacts the value of the SDF. The start delay in this study represents the start time offset as a measure between successively occurring start events on an equipment.

Contrary to the studies presented by Niedermeyer and Rose (2003, 2004) and Unbehaun and Rose (2006, 2007), we derive SDFs from MES history instead of discrete event cluster tool simulation studies. The proposed method, described in the following steps, automatically leads to a recipe combination specific SDF function depending on the start delay.

1. **Job Filter** - We intend to isolate the slow down effect by choosing jobs with characteristics suitable for analysis. For that reason, the jobs were filtered according to two sorts of conditions. Firstly, to neglect the influence of small lot size, the jobs were filtered out that do not stand for a lot with the maximal amount of wafer possible. Secondly, we only focus on jobs that were simultaneously processed with exactly one other lot during their period of processing. Note that the studied type of cluster tool usually process lots continuously, meaning that a lot may interfere

with two different lots at the beginning and/or at the end of processing (see Figure 2). These constraints on jobs used for analysis remarkably simplify or even enable the deduction of the SDF characteristic from real-world data samples.

2. **Job Categorization** - The remaining jobs are now double categorized according to the two input parameter of the SDF function, respectively recipe combination and start delay. First the jobs belonging to equal recipe combinations were grouped together. For instance, every job of recipe A processed in parallel with a job of recipe B somehow, belongs to the same group. Secondly the resulting recipe combination specific sets of jobs were subsequently categorized to their start delay. In conclusion, there exist a number of subsets of jobs, where each subset relates to a specific start delay within a specific recipe combination. We assume that each subset comprises jobs describing equal processing situations with respect to the recipe combination and start delay.
3. **SDF Calculation** - After filtering and categorizing the jobs to groups as described, we calculate the SDF for each job. Analogous to the LSF estimation, the observed RTT values are normalized by calculating the quotient of RTT and previous calculated RTT_{DEF}. Thereby the job's RTTs change into SDFs after division. Next, the statistical median of SDFs is calculated within each subset of jobs, in order to infer a single value from the samples, while reducing the effect of outliers. As a result, there exists a single SDF for specific start delays, additionally depending on a certain recipe combination.
4. **Interpolation and Outlier Removal** - Analogous to the estimation of the LSF function, there is a need to handle the difficulty of non-continuous data samples with outliers. Both were addressed by use of the multi-step estimation method cyclically applied, analogous to the LSF estimation. A linear interpolation is created, which is then moderately smoothed, followed by the removal of the most distant outlier, compared to smoothed interpolation. As result, the SDF function depending on the start delay is obtained for specific recipe combinations. In contrast to the case of LSF modeling, in this case, the remaining data volume for each subset is considerably smaller due to double categorization. Thus, it is much harder to derive a reasonable SDF function for each recipe combination in presence of lack of data samples. That is why the resulting model may not contain a SDF function for every recipe combination observed.
5. **Finalization** - The proposed SDF estimation method goes along with additional error handling actions assuring a more failure tolerant prediction model. The extensive use of filtering and categorizing computation leads to lower volume data subsets. As a consequence, statistical methods or interpolations to derive reasonable SDFs are not always applicable. For instance, a considerable lack of job samples for specific recipe combinations prevents recognition of a reasonable relationship between SDF and start delay. Having the goal to provide a robust model, we simplify the use of SDFs. Thus, we use the SDF no more depending on a certain start delay, but only depending on the start delay's algebraic sign. Consequently the model provides two SDFs per low-sampled recipe combinations, one for positive and one for negative start delays. For those occasions, a simplified SDF function is derived and then used for prediction.

2.4 Slow Down Factor Interpretation

In this section, we discuss the manifestation of the two mentioned cluster tool's processing modes, sequential and parallel processing, on SDF functions observed as a result of our analysis. We present characteristic lot processing behaviors, recognized for three pieces of equipment. The chosen pieces of equipment represent three different types of equipment, but all of them with the earlier described cluster tool architecture (see subsection 1.1). Based on an extensive analysis, applying the proposed automated modeling method, among several tens of cluster tools, mainly two SDF schemes stand out, which represent the slow down effect between possibly interfering lots. On the one hand, we recognize the parallel processing mode, on the other hand we identify the sequential processing mode. We chose three pieces of

equipment suitable to demonstrate the SDF characteristics, where equipment A shows parallel processing mode, equipment B stands for sequential mode and equipment C merges both modes.

For the diagrams depicted in Figure 4, Figure 5, and Figure 6, we only executed the first three steps of the described SDF estimation method, without interpolation and outlier removal (step 4). This way the reader gets a better idea of existing outliers and missing samples. Note, that the RTT prediction is based on the interpolated function with decreased influence of outlier.

The diagrams show the calculated SDFs dependent on the start delay for a specific recipe combination. For sake of simplicity, we choose the four most frequent recipe combinations. Moreover the start delays range between minus one and plus one, since they are normalized by the value of the relating RTT_{DEF} in order to make different equipment models easier to compare. A negative start delay represents a job started first, while a positive start delay is assigned to the lot, which started later.

We chose the equipment A as an example for parallel processing at cluster tools (see Figure 4). The simultaneously processed lots slow each other down. The SDF is considered as axially symmetric to the Y-axis, where an increased start delay results in a lower value for the SDF. We come to the conclusion that the less the start delay is, the more the lots compete for internal resources and thus slow each other down, and vice versa.

In contrast, the equipment B represents an example for sequential processing mode (see figure 5). In this case, lots were processed in parallel with regard to equipment events, but do not slow each other down as observed at equipment A. In fact, the succeeding lot waits for the preceding lot to finish. The SDF equals one for all negative start delays, which refer to the first lot. Consequently the precedent lot is not affected by the succeeding one, and respectively preferred in processing. For positive delays, which represents lots that start second among simultaneously processed pairs, the SDF decreases with increasing start delay. One can say that lower start delays, meaning smaller proportions of completed wafers of the first lot, result in higher waiting times for the second lot. We summarize that the preceding lot is always preferred, while the lot starting second is strictly forced to wait.

The third example (equipment C) supports the assumption, that sequential and parallel processing mode coexists on a single machine (see Figure 6). Obviously, there is a need for recipe combination specific sets of SDF's, precisely because the combination of recipes predetermines the processing mode, either sequential or parallel.

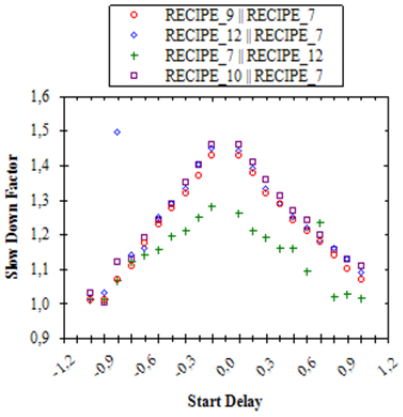

Figure 4: SDF Function (A)

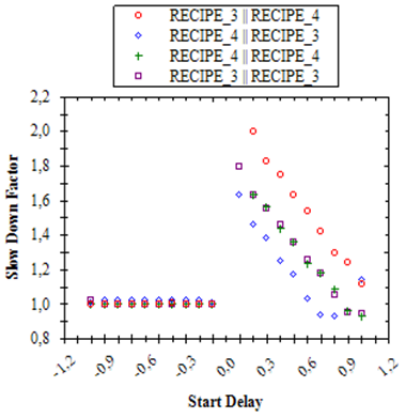

Figure 5: SDF Function (B)

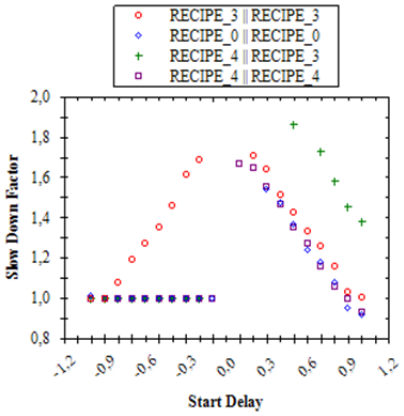

Figure 6: SDF Function (C)

2.5 Real-World Automated Modeling Challenges

Since we use real-world data records for automated modeling, we deal with multiple challenges arising in the real world. We face a high complex manufacturing system offering irregularities that compli-

cate the modeling process substantially. The following list provides some of the most common difficulties we deal with.

- **Process Stability** - Semiconductor manufacturers commonly integrate research and development activities into daily routine of production. Engineers try varying equipment setups, aiming to introduce improved processes. Those engineering activities often result in varying data records that show up as outliers during the modeling procedure, particularly in the analysis of processing times. In addition to engineering, technical difficulties at high complex equipment types also lead to outliers due to unexpected machine failures.
- **Equipment Events** - Semiconductor factories commonly comprise hundreds of machines from various suppliers at times with decades between their dates of construction. For this reason a specific equipment event, extracted from the MES, does not necessarily relate to the same physical activity at each type of equipment. In spite of efforts for standardization, unequal equipment event firing policies make it harder to create an equipment independent modeling approach.
- **Recipe Definition** - For our analysis we assume that the recipe unambiguously relates to a specific process, and accordingly to the way of processing on the equipment. However, a specific recipe on a certain machine may be processed in different modes, which are not consistently traceable in data records. For example, an equipment comprising several chambers is often capable of performing a certain process in multiple modes, which need differing number of chambers. As a consequence, these uncertainties lead to variations in data, which in turn lead to unpredictable processing behaviors particularly when the uncertainties occur frequently.
- **Temporal Changes** - We also assume that the equipment's processing behavior does not significantly change over time. But, we face a highly dynamic production environment, in which continuous changes take place. These changes range from mounting new hardware components to minor changes in process setups, and these modifications may considerably affect processing times.
- **Data Volume** - The success of the described modeling method significantly depends on the volume of data available for analysis. Due to the extensive use of data filtering and categorizing computations, the size of resulting data subsets possibly falls below a critical limit. An increasing level of detail in the model directly leads to a decreasing amount of data records for specific processing situations. In order to reasonably apply statistical methods, there is a need for a sufficient amount of data records. The challenge is to define a suitable level of model detail that allows for applying (statistical) outlier reduction methods.

3 RAW TOOL TIME PREDICTION

In this section, we present an algorithm deploying the proposed cluster tool model in order to simulate RTTs based on real-world data. Therefore, we apply the cluster tool models created without any additional corrections done by hand. We primarily use the proposed algorithm to evaluate both, the model's suitability as well as the prediction method's ability to predict RTTs. Moreover, the capability to simulate RTTs offers the possibility to quickly predict completion dates, possibly feeding look-ahead applications. We describe the algorithm used for prediction and consequently for evaluation as well as the results achieved with respect to accuracy. To evaluate the whole concept, comprising model and method, we compare simulated RTTs to RTTs collected in real-world. Therefore we simulate the processing for each job in history and then calculate the resulting relative errors for further evaluation.

Let's briefly describe the evaluation environment. The focused timeframe spans approximately 6 months of processing, equating to several thousand jobs per equipment. We deploy the same dataset for automated model generation as for evaluation with the aim to reduce negative effects that arise when ana-

lyzing a highly dynamic system over a long period of time. These conditions enable us to draw significant conclusions showing the potential of the described modeling approach under real-world conditions.

3.1 Prediction Method

We present an algorithm that simulates the processing, via predicting RTTs and completion dates, using the actual starting times (from history) and the proposed process time model. This algorithm, based on the prediction method presented by Unbehaun and Rose (2007), relies on real starting times to ensure that the start delays truthfully contribute to the prediction.

Unbehaun and Rose's approach underlies the understanding that each finish event is directly followed by a start event, resulting in continuous lot processing without gaps between process time slots. Based on the observations we made during real-world data analysis, we explicitly include these gaps into the prediction method. This modification is necessary to apply the method to the real-world data sets we use.

The algorithm takes into account that a lot's process may be subject to changing slow down situation due to starting and finishing other lots during a certain lot's process. More precisely the completion date is adjusted in the case of a changing recipe combination.

1. **Event Selection** - Within this simulation concept, there only exist two events triggering subsequent activities. On the one hand, the next event in time is the start of a lot from queue as given in the original data set from history. On the other hand, the next event is determined by the prior predicted completion date of a currently processing lot. Anyhow the simulation time T is set to the time of the next event. Depending on the type of the next event, the algorithm continues with step 2 or step 3, either starting a queued lot or finishing a currently processing one.
2. **Lot Start** - A new lot taken from the queue is started at the appointed time T and for that reason added to the list of lots in process. Then the contributory factors are determined as described in the model. The RTT_{DEF} is assigned according to the relating recipe as well as the LSF depending on lot size. Taking these factors into account the estimated completion date is calculated using following formula:
$$CD = T + RTT_{DEF} \times LSF$$
3. **Lot Finish** - The focused lot, for which completion date is predicted to current simulation time T, is removed from list with lots in process. Hence the finally simulated RTT is stored for analysis.
4. **Completion Date Adjustment** - Regardless of whether a new lot is started or a finished one is removed, the situation changed. Thus for every lot within the list of lots in process, the proper SDF_{NEW} is set according to the arising combination of recipes as well as to the start delay detected. A (re)calculation of SDF_{NEW} is necessary, because the previous SDF_{OLD} (if existent) might be no more valid since last event. With respect to a newly started lot, the (not yet existing) CD_{OLD} equals to the previous CD and the SDF_{OLD} is set by default to one. Finally, the completion date is revised according to the new situation using the following formula:
$$CD_{NEW} = T + \frac{CD_{OLD} - T}{SDF_{OLD}} \times SDF_{NEW}$$
5. **Termination -** The algorithm terminates if no more lots exist in the queue and no lot is in process at time T. If one of these two conditions is not fulfilled, the algorithm continues with the first step determining the next event.

3.2 Prediction Results

For evaluation, we simulate the processing of jobs in history and compare predicted RTTs with the corresponding originals. Therefore we apply the prediction method previously described (see section 3.1.). We

remember that the presented results refer to several thousand jobs simulated per equipment without exceptions.

We separately focus on the three model's components constituting the RTT calculating formula. By separated examination, we determine single factor contribution to the accuracy of simulation results. Table 2 shows the relative RTT errors relating to the 50%-percentile and the 90%-percentile. *DEF* stands for the prediction using RTT_{DEF} only. *LSF* labels simulation results achieved by use of RTT_{DEF} extended by LSFs. *SDF* is designated to the entire model including all factors as proposed.

Table 2: Relative RTT Error Percentiles

4	Default (DEF)		Lot Size Factor (LSF)		Slow Down Factor (SDF)	
	DEF50%	DEF90%	LSF50%	LSF90%	SDF50%	SDF90%
Equipment A	20%	46%	18%	43%	5%	32%
Equipment B	7%	45%	6%	42%	2%	32%
Equipment C	7%	43%	7%	42%	4%	34%

We assess considerable improvements in accuracy for the use of the proposed model, especially for equipment A (parallel mode). The median for the relative RTT error observed for the use of SDFs is strictly less than 5% (SDF50%). Compared to the exclusive use of a static value like RTT_{DEF}, simulating RTTs using SDFs leads to positively left shifted error distribution with decreased range. We observed small increase of prediction accuracy when involving LSFs. The reason for this is that approximately 5% of all lots relate to a remarkably smaller lot size only.

The diagrams depicted in Figures 7, 8 and 9 show the histograms of relative RTT errors simulated for *DEF, LSF* and *SDF*. These diagrams show that approximately 70% of predicted jobs, slightly varying for different equipment types, come along with a relative error smaller than 10% for *SDF*. Figures 10, 11 and 12 show relative RTT errors as bars for most frequent recipe combinations (RCs) processed. Each bar relates to the 50%-percentile at its lower edge and to the 90%-percentile on the top of the bar.

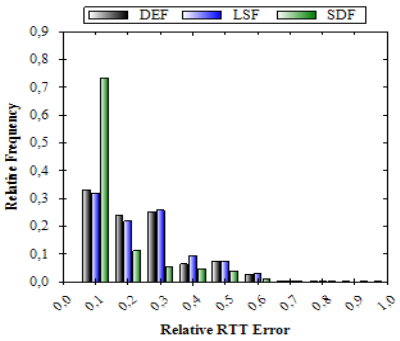

Figure 7: Error Histogram (A)

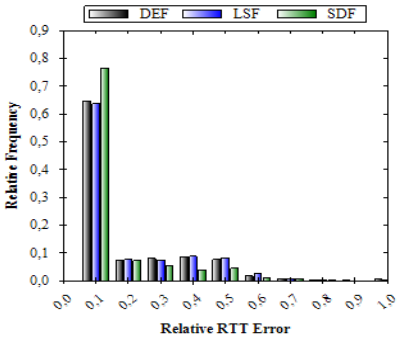

Figure 8: Error Histogram (B)

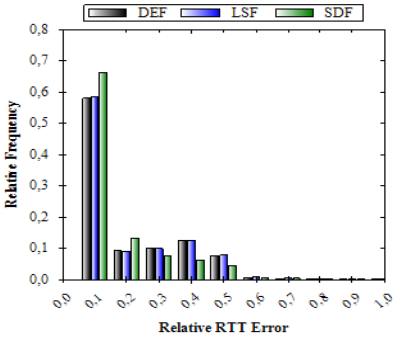

Figure 9: Error Histogram (C)

5 SUMMARY

In this paper, we present an analytical cluster model used for RTT prediction involving the effect of lot size and simultaneously processed lots interfering with each other. We state that the proposed analytical cluster model is suitable to predict RTTs with respect to accuracy and prediction coverage. Especially the use of Slow Down Factors depending on a certain recipe combination and start delay adequately mirrors sequential and parallel processing mode. Unfortunately we were not able to prove the approach's applicability to predict the makespan with desired accuracy. For further research, we intend to evolve the model in order to be capable of simulating an entire lot schedule using ordered job queues as input.

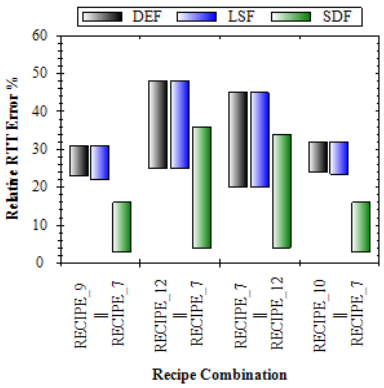

Figure 10: RC Errors (A)

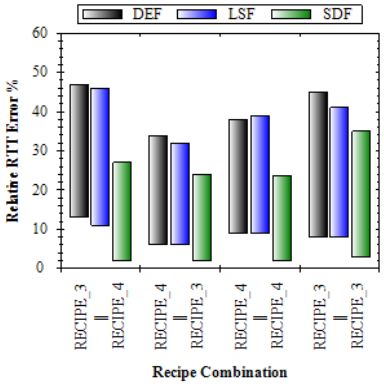

Figure 11: RC Errors (B)

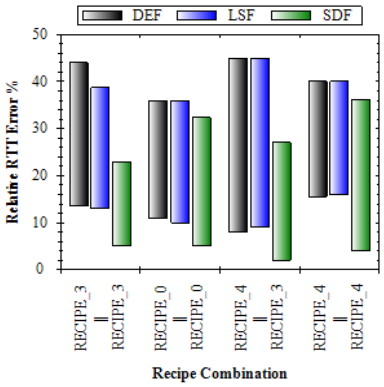

Figure 12: RC Errors (C)

Moreover, the idea of automated model parameterization using real-world data leads reasonable models, but depends on volume of data and quality. Preliminary experiments using regression to derive functional relationships from raw data showed promising results. We expect that fast and accurate lot schedule simulation, quickly performed, will enable optimized lot scheduling on operational level.

ACKNOWLEDGEMENTS

This work was supported by Grant 13139/2219 of the Sächsische Aufbaubank (SAB).

REFERENCES

Hosoe, H., N. Knanamori, and K. Yoshida. 2007. "The Methods of Data Collection and Tool Processing Time Estimation in Lot Processing" In *Proceedings of International Symposium on Semiconductor Manufacturing (ISSM 2007)*.

Kohn, R., S. Werner, and O. Rose. 2010. "Automated Semiconductor Equipment Modeling And Model Parameter Estimation Using MES Data." In *Proceedings of Advanced Semiconductor Manufacturing Conference (ASMC) IEEE/SEMI*, 11-16.

Lange, J., K. Schmidt, R. Borner, and O. Rose. 2008. "Automated Generation and Parameterization of Throughput Models for Semiconductor Tools." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 2335-2340. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Niedermeyer, H., and O. Rose. 2003. "A Simulation-based Analysis of the Cycle Time of Cluster Tools in Semiconductor Manufacturing." In *Proceedings of the 15th European Simulation Symposium*, 349-354.

Niedermeyer, H., and O. Rose. 2004. "Approximation of the Cycle Time of Cluster Tools in Semiconductor Manufacturing." In *Proceedings of Industrial Engineering Research Conference (IERC)*.

Perkinson, T. L., R. S. Gyurcsik, and P. K. McLarty. 1996. "Single-Wafer Cluster Tool Performance: An Analysis of the Effects of Redundant Chambers and Revisitation Sequences on Throughput." *IEEE Transactions on Semiconductor Manufacturing* 7(3):369-373.

Schmidt, K., J. Weigang, and O. Rose. 2006. "Modeling Semiconductor Tools For Small Lotsize Fab Simulations." In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. R. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1811-1816. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Unbehaun, R., and O. Rose. 2006. "The Use of Slow Down Factors for the Analysis and Development of Scheduling Algorithms for Parallel Cluster Tools." In *Proceedings of the 2006 Winter Simulation*

Conference, edited by L. R. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1840-1847. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Unbehaun, R., and O. Rose. 2007. "Predicting Cluster Tool Behavior with Slow Down Factors." In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1755-1760. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wood, S. C., S. Tripathi, and F. Moghadam. 1994. "A Generic Model For Cluster Tool Throughput Time And Capacity." In *Proceedings IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 194-199.

Wood, S. C. 1996. "Simple Performance Models for Integrated Processing Tools." *IEEE Transactions on Semiconductor Manufacturing* 9(3S):320–328.

AUTHOR BIOGRAPHIES

ROBERT KOHN is a PhD student at Dresden University of Technology. His focus is on simulation based resource scheduling by use of analytical equipment models and heuristic search methods in semiconductor industry. 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 University of Applied Sciences Stralsund, Germany. His e-mail address is robert.kohn@tu-dresden.de.

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 e-mail is oliver.rose@tu-dresden.de.