7th Int. IEEE Conference on Emerging Technology and Factory Automation (ETFA'99), Barcelona, pp. 1071 - 1080.

COLORED PETRI NETS FOR THE PERFORMANCE EVALUATION OF A SEMICONDUCTOR FABRICATION FACILITY

Armin Zimmermann*, Heinz Westphal†, and Stephan Gramlich**

*Technical University Berlin, Real-Time Systems and Robotics, Franklinstr. 28/29, Sekr. FR 2-2, 10587 Berlin, Germany, Phone: +49 (30) 314 73 112 Fax: +49 (30) 314 21 116 E-mail: azi@cs.tu-berlin.de

†University of Magdeburg, Institute for Automation Technology P.O. Box 4210, 39016 Magdeburg, Germany Phone: +49 (391) 67 12795 Fax: +49 (391) 11191 E-mail: heinz.westphal@e-technik.uni-magdeburg.de

**AMD Saxony Manufacturing GmbH, Wilschdorfer Landstr. 101 Mailstop E 21-AU, 01109 Dresden, Germany

Abstract - *The design, implementation, and operation of modern manufacturing systems is a complex task. Without powerful modeling and performance evaluation techniques it is practically impossible to efficiently design a manufacturing system. In this paper the application of Petri net modeling and analysis techniques is demonstrated using a real-life industrial example from the semiconductor fabrication field. It is shown how the performance can be evaluated and, using the results, system parameters are adjusted for an efficient production. A hierarchical colored model of the application example is presented. The detailed models of identical failing machines are aggregated to reduce the computational effort of the performance evaluation. Using the industrial example as a case study, this paper aims at presenting some theoretical methods and their application using a software tool during the design of manufacturing systems.*

1 INTRODUCTION

Modern manufacturing systems are characterized by automated processes and rapidly changing demands in the production output. Their design, implementation, and operation is therefore a complex task. The prediction and model-based optimization of the production process behavior is an important part of the design. This is especially the case if the resources are subject to failures, thus decreasing the production output. Without powerful modeling and performance evaluation techniques it becomes practically impossible to design a manufacturing system.

This illustrates the need for modeling methods, analysis techniques, and corresponding computer tools for manufacturing systems. In this paper the application of modeling and analysis techniques is demonstrated using a real-life industrial example from the semiconductor fabrication field [12]. We show how the performance can be evaluated and system parameters are selected for a more efficient production.

All techniques presented here are based on the modeling formalism of *stochastic Petri nets* [2, 3, 4]. They combine a simple graphical description and mathematical analysis methods in one framework. Manufacturing systems are one of the classical application areas of Petri nets. They can be used for the modeling, qualitative validation, performance evaluation, and control of production processes [11].

In a previous paper [13] the application example was modeled and analysed using a simplified uncolored Petri net model, which is briefly recalled in Section 2. This is advantageous for a first impression of the system behavior and a rough estimation of some performance measures. However, as the design process advances, more details of the planned system are available or have to be adjusted based on a performance evaluation. In Section 4 of this paper a more detailed model of the application example is presented, which is specified using a class of hierarchical colored Petri nets. This model type [15, 18] has been developed especially for the application area of manufacturing systems. Due to the refinement of the model, performance analysis becomes more and more computa-

tionally expensive. Different identical machines and their failure/repair behavior are therefore aggregated in Section 4.2. This leads to simpler models which are easier to evaluate.

Some background of the used performance evaluation techniques based on direct numerical analysis or discrete event simulation is presented in Section 5. They are applied to the industrial example afterwards to compute performance measures depending on selected system parameter settings. For the computations the software tool TimeNET [7, 16] has been used (see Section 6), in which the explained algorithms are implemented. This tool is available for personal computer platforms running the Linux operating system and can therefore be used in industrial environments, where powerful workstations are not always available.

Using the example described in Section 2 as a case study, this paper aims at presenting some theoretical methods and their application using a software tool during the design of manufacturing systems.

2 AN APPLICATION EXAMPLE

The Fab 30 wafer fabrication facility at Dresden, Germany, is AMD's European state-of-the art manufacturing facility for advanced microprocessor products [1, 12, 13, 14]. It will be staffed by approximately 1400 employees and will at full capacity produce 5000 eight-inch wafers per week. Semiconductor manufacturing system productivity is very important because of the enormous necessary investments and running costs. Due to the complicated production and test procedures and the large number of shared resources, the behavior of a semiconductor manufacturing system is very complex. The development environment will focus on getting products to market quickly, insuring that performance and cost goals are met or exceeded. The tools, methodology and technology necessary for these goals provide a very challenging environment for the design, simulation, optimization, and control of semiconductor manufacturing systems.

In addition to the wafer processing and inspection stations, the automated material handling system (AMHS) is an important part of the wafer production. The AMHS includes automated storage and retrieval systems (called *stockers*), monorail, and intrabay components. Wafer lots are stored in stockers between two operation steps. The material control system coordinates the actions of the AMHS components. The stockers are responsible for controlling material movements to and from the monorail and to and from the manual operator ports. A *monorail system* with tracks, switches, and monorail vehicles transports wafers between the stockers. On one monorail track the vehicles can only travel in one direction. Loops and switches in the track layout allow transport in both directions.

The whole fabrication layout is divided into different areas. In this paper we concentrate on modeling and evaluation of one selected area. The material transport between those areas is done by mass transfer systems. Only the wafers pass the border between areas, pod and cassette stay in the area. Each mass transfer system consists of two stockers (one stocker for each area) and a transfer station between the stockers.

From the point of view of the design process, the different wafer processing stations can not be optimized. There is only the choice of how many identical machines are installed, and possibly which types of machines. However, compared with the processing stations the AMHS is relatively inexpensive, but can have a substantial influence on the overall productivity. Stocker size and placement, monorail track layout, number of monorail vehicles, vehicle routing strategies etc. should therefore be optimized during the design. The goal is e.g. to maximize throughput, minimize work in process or some performance measures that incorporate all important cost/profit factors.

3 A FIRST UNCOLORED MODEL

During the early stages of the design process, a very global model without implementation specific details is sufficient. Figure 1 shows the *generalized stochastic Petri net* (GSPN) model of the considered part of Fab 30.

Dotted boxes contain model elements that belong together. Stockers and machines are organized in groups. For a certain processing step, a lot has to be transported to the corresponding stocker, from where it is brought to the machine. After processing it, the lot is placed in the stocker and is available for further steps. Lots arrive in stockers 102 or 103 (transition `In` fires), which are modeled together by `S102`. Each stocker can contain lots in different processing states, and is therefore modeled with several places. One place `c` counts the available places (capacity), and the others contain tokens that model lots in the corresponding processing states.

Transport operation one takes a `raw` lot to stocker 98 (`S98`). Each of the four transport operations has a start transition (immediate, named `t?s`), and a timed transition named `t?d` with an associated transport delay. To begin a transport, a monorail vehicle is needed (a token is in place `Vavail`), and a place in the destination stocker is free.

Groups of identical machines can be found at the bottom of the model. There are five groups: Equinox MP, Mira 1 Polisher POL, microscope inspection INS, semitool SNK, and Orbot inspection station

DEF. The number of available machines is given by the tokens in the places `avail`. Like for the transport actions, processing inside the machines is modeled with a start transition, an in-work place, and a timed `done` transition. Machines belonging to the groups named MP, POL, and SNK may fail (transition `fail` fires) and have to be repaired (transition `rep`).

4 A HIERARCHICALLY REFINED COLORED MODEL

Colored Petri nets [8] offer more advanced modeling facilities like distinguishable tokens and hierarchical modeling with respect to uncolored nets. The pure graphical description method of Petri nets is, however, hampered by the need to define color types and variables comparable to programming languages. This is often not well accepted by users without a strong background of computer science.

To solve this problem, a method for the modeling of manufacturing systems has been presented in [15, 17]. A class of colored stochastic Petri nets is introduced especially for the modeling of manufacturing systems. Two color types are predefined:

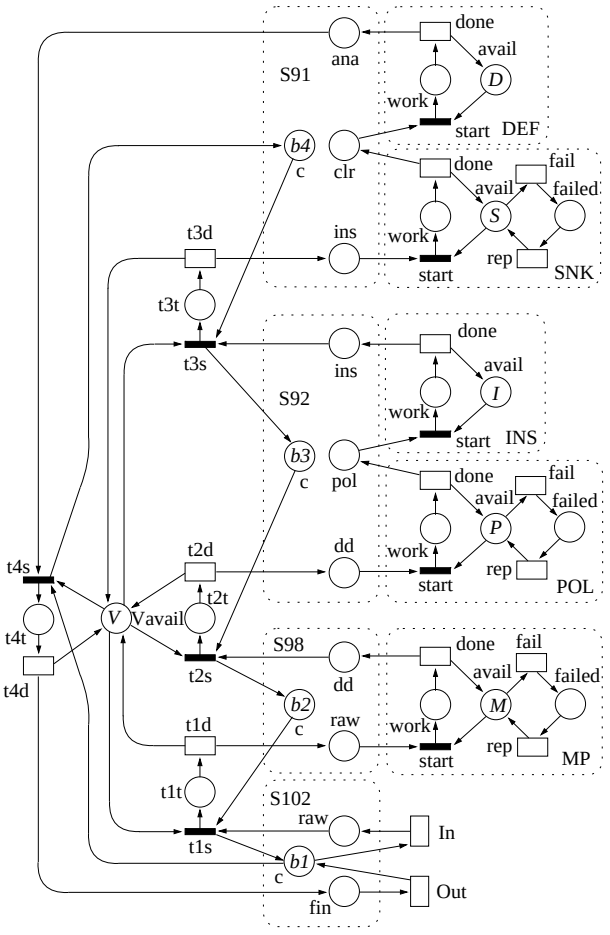

Figure 1: GSPN model of the wafer fabrication

Object tokens model work pieces inside the manufacturing system, and consist of a name and the current state. *Elementary tokens* cannot be distinguished, and are thus equivalent to tokens from uncolored Petri nets. Places can contain only tokens of one type. Textual descriptions needed in colored Petri nets for the definition of variables and color types can be omitted, and the specification of the types of places and arcs are implicitly given. The models are hierarchically structured, which is necessary to handle complex systems. Structure and work plans are modeled independently using this net class. This is important for the evaluation of different production plans, where the structural model is not changed. The structural model describes the abilities and work plan independent properties of the manufacturing system resources, such as machines, buffer capacities, and transport connections. Production sequence models specify the work plan dependent features of the manufacturing system. The application example is modeled with the dedicated Petri nets in the following subsections. During this process, a more detailed model than the first uncolored one is developed.

Firing delays are associated with transitions for the performance evaluation. We adopt the set of distributions as defined for *extended deterministic and stochastic Petri nets* (eDSPNs, [5]) here. They include immediate, exponential, deterministic, and more general transitions. The performance evaluation of models of this class is described in Section 5.

4.1 Modeling the Structure

Figure 2 shows the top level of the structural model. Places model buffers and other possible locations of parts. The places `S103pu`, `S98pu`, `S92pu`, `S91pu`, and `S102pu` correspond to places where a monorail vehicle stops at a stocker and exchanges the transported carrier with the stocker. Numbers in square brackets specify the capacity of each place. The stockers are modeled with the substitution transitions `S103`, `S98`, `S92`, `S91`, and `S102`. Through stocker `S103` new parts arrive, while finished wafers are removed by stocker `S102`. The other stockers have places `S98io`, `S92io`, and `S91io` to exchange carriers with processing units. The transport of carriers between stockers and machines or from a machine to the next is done manually and modeled with the transitions `MPman`, `POLman`, `POL2INS`, `INSman`, `SNKman`, `SNK2DEF`, and `DEFman`. The five groups of machines are modeled with `MP`, `INS`, `POL`, `DEF`, and `SNK`, together with their buffer places (`?w`) where carriers are located during processing. The Monorail system is modeled with the transitions `Move1`..`Move5`.

In principal, there are two different operations that can be performed: transport and processing of work pieces. The former corresponds to moving a token

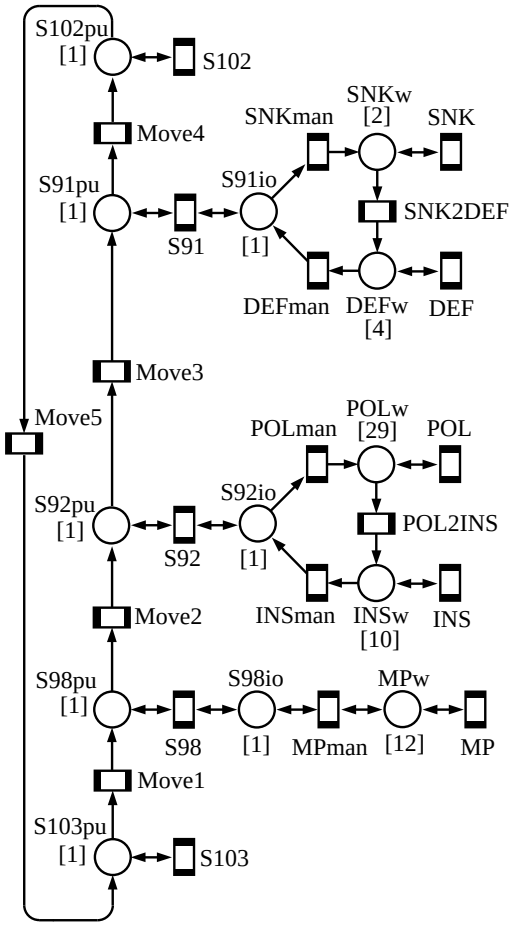

Figure 2: Main structural model

to another place, while the latter is modeled by a change in the color of the token that corresponds to the work piece. Transitions modeling machines specify processing steps which only change the token color. This is emulated by removing the former token form the place and instantly adding a token with the new color during the firing of the transition. Therefore many transitions and places are connected by arcs in both directions, which are conveniently drawn on top of each other.

Transitions with thick bars depict substitution transitions, which are refined by a submodel on a lower level of hierarchy. These transitions are e.g. used to describe the behavior of a machine with more detail during a top-down design.

Figure 3 depicts the submodel corresponding to transition `S98` in the upmost level of hierarchy (figure 2). Places `S98pu` and `S98io` form the interface to the surrounding model parts and are therefore drawn with a dashed line. Place `Store98` is the actual storage in the stocker. Transition `LoadUnload` models the loading and unloading of monorail vehicles that have docked at the stocker. `Pickup` corresponds to the place for in- and output of wafer lots to the machine side of the stocker.

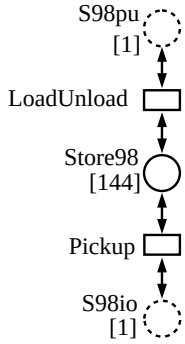

Figure 3: Submodel of stocker 98

4.2 Aggregation of Machine Submodels

Each substitution transition (depicted as ▯) is refined by a subpage that describes the behavior of the resource with more detail. Submodels from a library of standardized building blocks (*templates*) can be parameterized and instantiated while refining the model. Each template has a set of parameters such as processing times or buffer capacities. By associating values to the parameters, each of them represents a class of structurally similar resources.

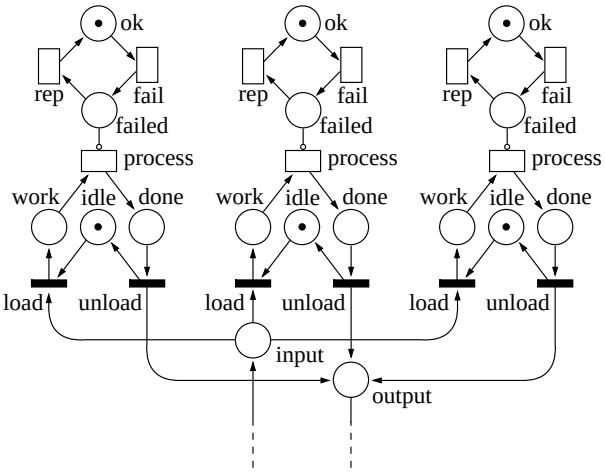

Figure 4: Detailed model of 3 failing machines

For the application example three groups of machines (MP, POL, and SNK) are subject to failures and repairs. This behavior is specified in a submodel of the substitution transition corresponding to the machine group. The first performance evaluations with the detailed model showed that the introduction of these failures substantially increased the necessary computational effort. The problem of complex reachability graphs for models of realistic size is known under the term *state explosion*. Figure 4 shows a detailed model of three identical machines with failures and repairs. For simplicity reasons this and the following models are uncolored GSPNs without hierarchical refinements. To reduce the model complexity, in a first step the failures were incorporated

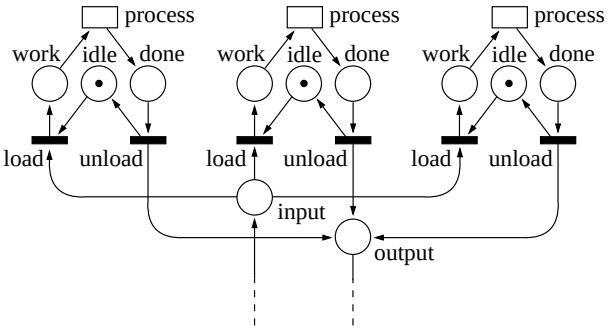

Figure 5: Simplified model of 3 machines with aggregated failure behavior

in the remaining model by adjusting the firing delay of each `process` transition. For this task, the failure/repair model was numerically analysed in isolation, and the steady-state probability of being in state `ok` was derived. Please refer to Section 5.1 for details on the numerical analysis techniques. The firing delay of each `process` transition was divided by the computed probability, thus keeping the throughput approximately equivalent. Figure 5 shows the resulting model.

Different combinations of this simplified model with surrounding model parts were numerically analysed and the results compared with the original model. In all cases the error was less than one percent, often much smaller. In a second step the three identical machines were aggregated to one model, which is shown in figure 6.

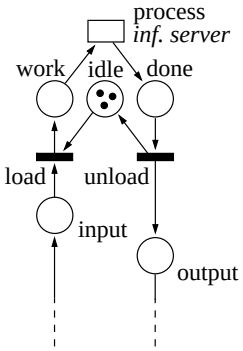

Figure 6: Second simplification: aggregation of machines

Only two slight changes are necessary: the capacity of the group of machines is now specified as 3 in place `idle`, because all three machines are modeled together. Secondly, the firing semantic of transition `process` is changed to *infinite server*. The model then behaves as if there is one transition (server) for each waiting customer (token in place `work`). In the case of a transition with exponentially distributed firing time as it is here, the mean firing delay is thereby divided by the actual number of tokens in place `work`. Thus with a less complex model the performance measures can be computed. It should be noted that the two last models are equivalent from a performance point of view, resulting in a total aggregation error of less than one percent. Because the performance evaluation of the whole model is carried out using simulation, this accuracy is sufficient.

4.3 Modeling the Production Routes

In addition to the structural model for each product a model of the production steps has to be defined. This set of models is described with the same type of dedicated colored Petri nets, with some slight differences. Each step can only be carried out by a resource that is available in the manufacturing system layout. The production routes represent paths through the structural model, hence the same places and transitions as in this model can be found here, possibly several times.

There is at least one work plan model for each part. In the case of the application example, the models have been divided into smaller parts like the one shown here. An independent work plan is e.g. specified for empty monorail vehicles, incorporating routing strategies. It is shown in the drawing area of the software tool screenshot in figure 11.

Figure 7 shows the part of the production sequence model that corresponds to the wafer processing at

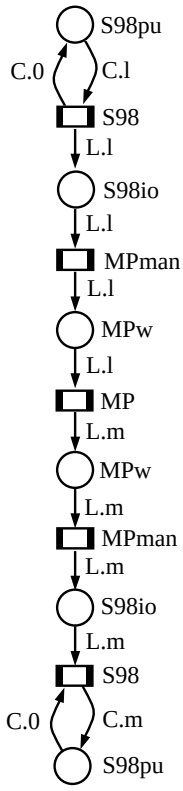

Figure 7: Workplan for the MP processing station

the MP machines. Transitions, places and their connecting arcs correspond to the net elements in figure 2. Model elements in production sequence models refer to their structural counterpart through the use of identical names. It is obvious that a substitution transition in a production sequence model has to be refined with a submodel. This submodel is then associated to the submodel of the corresponding substitution transition in the structural model. This relationship between both model parts holds for all submodels in the hierarchy. The term *associated Petri nets* is used for this concept of specifying different views of a system in related model parts.

When a monorail vehicle transporting one wafer lot arrives at the exchange place `S98pu`, the lot is taken from the vehicle and stored in the stocker `S98`. An empty vehicle (modeled by `C.0`) remains at the exchange place. The lots are later taken from the stocker and transported to the machines `MP`. Processing them at one of those machines changes the processing state of the lot from `L.1` to `L.m`. The name of a part and its processing state are separated by a dot. Each lot is taken back to the stocker and put on a monorail vehicle later (right hand side of the model).

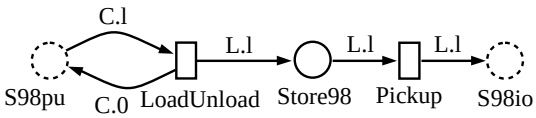

Figure 8: Refined workplan of stocker 98

Figure 8 shows the refined submodel of the left transition `S98` in figure 7. It models the inner behavior of stocker 98 and is associated to the structural model of the stocker shown in figure 3. Advanced features of workplan models like alternatives and guard expressions that control the material flow were not needed for the presented model parts.

The structural and workplan models are automatically merged to create a complete model first [17]. During this process, the information contained in the production route models are added to the structural model.

5 PERFORMANCE EVALUATION

After specifying a manufacturing systems, its performance and dependability can be evaluated. Different variations of the system and their resulting performance and dependability measures are computed and compared.. The aim of this investigation is to obtain a better understanding of the correlations between details of the manufacturing system (e.g. the buffer capacities) and the main performance measures (e.g. the throughput). Proposals can be derived in order to increase the manufacturing system's productivity.

In this work the focus is on steady-state analysis and simulation, which computes the performance of a system in equilibrium (provided that it exists). Either direct numerical analysis or discrete-event simulation can be used to obtain the desired measures from the model. If control strategies have to be evaluated as well, transient analysis evaluates the system behavior after a certain amount of time has elapsed from the initial marking. Other algorithms are needed for this type of evaluation, which are not considered here.

Although the used class of colored Petri nets offers advanced modeling facilities for manufacturing systems, the underlying stochastic process is the same as for a behaviorally equal uncolored model. The techniques developed for these net types can therefore be adapted to the colored case. Analysis and simulation methods for *extended deterministic and stochastic Petri nets* (eDSPNs, [5, 6]) are applied to the colored net type [17].

5.1 Numerical Analysis

For a direct numerical analysis of the model a full exploration of the set of reachable system states Z is necessary. The current state of the model is given by the vector of token multisets in all places and is referred to as the *marking*. The *reachability graph* is defined by the set of vertices corresponding to the markings reachable from the initial marking and the edges corresponding to transition firings. If an immediate transition is enabled in a marking, no time is spent in it during the marking evolution. The reachable markings can be partitioned in vanishing Z^{imm} and tangible markings Z^{tim} accordingly [3].

The behavior of the model is given by the initial marking and the subsequent transition firings, describing a stochastic process [5]. The type of process depends on the types of allowed firing delays and whether certain transitions are enabled together in one marking or not. The firing delay of transitions considered in the techniques used here [6] can either be zero (immediate), exponentially distributed, deterministic, or belong to a class of general distributions called *expolynomial*. Such a distribution function can be piecewise defined by exponential polynomials and has finite support. It can even contain jumps, making it possible to mix discrete and continuous components. Many known distributions (uniform, triangular, truncated exponential, finite discrete) belong to this class.

In a first step for the numerical analysis of a stochastic Petri net model the reachability graph is computed. The following information is needed from the tangible markings Z^{tim}, i.e. the reduced reachability graph:

- $Z^{exp} \subseteq Z^{tim}$, the set of states with only expo-

nentially timed transitions enabled

- $\forall z_i \in Z^{exp}, z_k \in Z^{tim} : \lambda_{z_i,z_k}$, the firing rate of an enabled transition with exponential firing time from state z_i to state z_k,
- U^{gen}, the set of transitions with non-exponentially distributed firing time
- $\forall u \in U^{gen} : Z^u, Q^u, \Delta^u, u^{delay}$. Z^u are the states in which transition u is enabled. Q^u denotes the matrix of exponential firing rates of transitions, which are enabled in parallel to the enabled transition u in states from Z^u. $\Delta^u = [\delta_{z_i,z_k}]$ denotes a matrix, which stores for each state $z_i \in Z^u$ the probability for reaching state z_k by firing immediate transitions after the non-exponential transition has fired in state z_i. u^{delay} denotes the non-exponential firing time distribution function.

Due to the restriction of at most on enabled transition with non-exponentially distributed firing time in each marking, the sets Z^{exp} and Z^u for all $u \in U^{gen}$ dont have common elements, and together they contain all states of Z^{tim}, the set of all tangible states of the reduced reachability graph.

The reduced reachability graph of a model with only immediate and exponential transitions is defined to be isomorphic to a continuous-time Markov chain (CTMC), because of memoryless state changes. In case of a CTMC only the corresponding linear system of equations has to be solved. In case of deterministic or more general non-exponential transitions an additional step is required. The underlying stochastic process is only memoryless at some instants of time, called *regeneration points*. If a transition with non-exponentially distributed firing delay is enabled in a marking, the next regeneration point is chosen after firing or disabling this transition. The time of firing the next exponential transition is taken otherwise.

Therefore in a next step the following matrix integral equations have to be solved for all $u \in U^{gen}$:

$$\begin{aligned} \Omega^u &= \int_0^\infty e^{Q^u t}\, d\, u^{delay}(t) \\ \Psi^u &= \int_0^\infty e^{Q^u t}\, \left(1 - u^{delay}(t)\right) dt \end{aligned}$$

Ω^u denotes the the matrix of state-transition probabilities of the subordinated stochastic process at the end of the enabling period of u and Ψ^u the matrix of expected sojourn times of the states of this process during the enabling period of u.

By taking only the regeneration points into account, a discrete-time Markov chain is embedded. For the later analysis of this Markov chain, the stochastic matrix $\mathbf{P}$ of one-step transition probabilities and a matrix $\mathbf{C}$ of conversion factors have to be computed [6]. $\mathbf{P}$ describes the probabilities of state changes of the embedded Markov chain between two regeneration points. $\mathbf{C}$ describes the conditional sojourn times in the states between two regeneration points. There are some states of the original process which are not states of the embedded system. The time spent in those states from the enabling of a non-exponential transition until its firing or disabling is kept in entries of the $\mathbf{C}$ matrix. In addition to that, the diagonal entries of the $\mathbf{C}$ matrix contain the mean sojourn times in tangible states, which are needed for the conversion at the end of the algorithm. For states with solely exponential transitions enabled, only the diagonal entry of the corresponding $\mathbf{C}$ matrix row is different from zero and can be computed directly from the reduced reachability graph.

$$\begin{aligned} &\forall z_i \in Z^{exp} : \\ &\mathbf{P}_{z_i,z_k} = \begin{cases} 0 & \text{for } i = k \\ \frac{\lambda_{z_i,z_k}}{\lambda_{z_i}} & \text{otherwise} \end{cases} \\ &\mathbf{C}_{z_i,z_k} = \begin{cases} \frac{1}{\lambda_{z_i}} & \text{for } i = k \\ 0 & \text{otherwise} \end{cases} \\ &\forall u \in U^{gen}, \forall z_i \in Z^u : \\ &\mathbf{P}_{z_i} = \Omega^u_{z_i} \Delta^u \\ &\mathbf{C}_{z_i,z_k} = \begin{cases} \Psi^u_{z_i,z_k} & \text{for } z_k \in Z^u \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

λ_{z_i} denotes the sum of all rates of enabled exponential transitions in state z_i.

In order to compute the entries of the $\mathbf{P}$ and $\mathbf{C}$ matrix for transitions with non-exponentially distributed firing times, the evolution of the stochastic process during the enabling of a transition with non-exponentially distributed firing delay is analysed. At most one transition of this type can be enabled per marking for this type of analysis. Therefore only exponential transitions may fire during the enabling period, resulting in a continuous-time subordinated Markov chain (SMC) of the non-exponential transition. The transient and cumulative transient solution of this Markov chain computes the $\mathbf{P}$ and $\mathbf{C}$ matrix entries. Jensens method (also known as randomization or uniformization) can be applied for both.

A linear system of equations based on the $\mathbf{P}$ matrix has to be solved for all steady-state analysis techniques. Standard algorithms like successive over relaxation (SOR) and sparse Gaussian elimination are applicable for this task. The vector of state probabilities in steady-state γ of the embedded markov chain is computed by solving the following set of linear equations:

$$\gamma(\mathbf{P} - \mathbf{I}) = 0, \quad \sum_i \gamma_i = 1$$

where the unity matrix of applicable dimension is denoted by $\mathbf{I}$.

The state probabilities of the actual stochastic process (the vector $\pi \in \mathbb{R}^{|Z^{tim}|}$) can then be obtained as the mean sojourn time in each state between two regeneration points. Formally, this corresponds to multiplying the EMC solution vector by $\mathbf{C}$ and normalizing it.

$$\gamma' = \gamma\,\mathbf{C}, \quad \pi = \frac{1}{\sum_i \gamma'_i}\,\gamma'$$

π_i then denotes the probability of being in state i in steady-state. Finally, the token probability distribution in the places of the net as well as the user-defined performance measures are calculated from the state probability vector. The described algorithms are implemented in TimeNET and have been used for the necessary performance evaluations during the submodel aggregation as described in Section 4.2.

5.2 Discrete Event Simulation

For many models the restriction of not more than one enabled non-exponential transition per marking is violated. Another problem of all analysis methods is the size of the reachability graph. Not only the computational complexity grows, it makes the analysis impossible for some models of realistic size due to memory space restrictions. Discrete-event simulation is still applicable for the performance evaluation in this cases. However, other problems arise with the statistical evaluation of the samples and the accuracy of the results.

The used software tool TimeNET [7] comprises an efficient simulation component [10], which evaluates models without the restriction of enabled non-exponential transitions. The simulation is a stochastic experiment. All samples drawn during the simulation run are random variables. The user-specified performance measures are obtained by estimating the mean value of the sampled data. The precision of this estimate has to be calculated as well, based on the confidence interval derived from the sample variance. The initial transient phase of the simulation run of a steady-state evaluation is detected and ignored. Variance estimation of the samples is performed by spectral variance analysis, allowing a robust estimation even for correlated samples as they are common if only one replication of the simulation process is running.

The length of a simulation run is decreased with a parallelization of simulation processes. Each one of them simulates the whole net and sends sample packets to a central process. As long as the model can be handled on one workstation, this approach is simpler to implement and more efficient than parallel simulation with a distributed model. The central process monitors the accuracy and stops the simulation after reaching the specified threshold. To reduce the simulation length further, variance reduction with control variates is applied [9]. The correlation between an estimator of interest and another stochastic parameter of the model is exploited to reduce the variance of the estimator for the same number of samples.

The simulation component is used for the performance evaluation. For the application example, the goal is to evaluate the throughput (number of wafers produced per week) and the work in process (mean number of lots in the system). Two corresponding performance measures are defined in the model. The number of monorail vehicles and the buffer size of the stockers are important parameters which are variated during the evaluation. All evaluations have been carried out on a cluster of ten UltraSparc workstations with a confidence interval of 95% and a maximum relative error of 10%. Each simulation run typically took 50 seconds real time (including distribution overhead) and 300 seconds overall CPU time to complete.

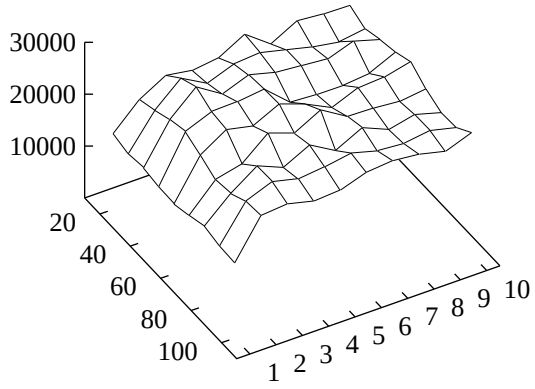

Figure 9: Throughput versus number of vehicles and buffer capacities

Figure 9 shows the production of wafers per week depending on the number of monorail vehicles (1..10) and stocker capacities (20..100). For a stocker capacity of n, the buffer of each one of the stockers can contain up to n lots. The routing strategies of the vehicles have to be adjusted accordingly, because they must not pick up lots that are headed for a full stocker to avoid deadlocks. It is not surprising in the plot that for higher numbers of monorail vehicles the throughput increases. However, this is only the case for numbers up to three. More vehicles do not increase the throughput. In the evaluated range of $30 \ldots 100$ for stocker capacities, no significant influence on the throughput is visible.

The relation of work in process, number of monorail vehicles and stocker capacities has been computed and is plotted in figure 10. Increasing the number of vehicles as well as higher stocker capacities lead to substantially more work in process. The influence of the stocker capacities is more significant.

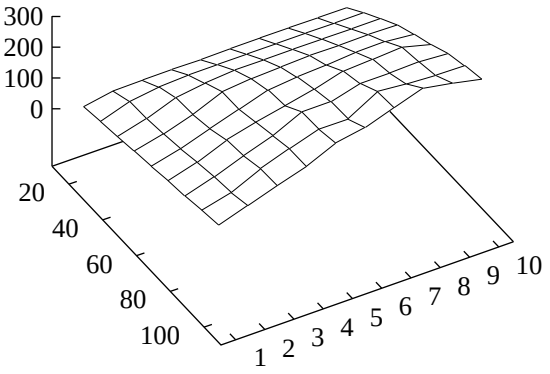

Figure 10: Work in process versus number of vehicles and buffer capacities

Judging from the performance evaluation experiments, three or more monorail vehicles should be used and the stocker capacities should be restricted to 30. However, it is noted that due to not considered vehicle failures and recharging some more vehicles could be necessary.

6 TOOL SUPPORT

A necessary condition for modeling and performance evaluation of the application example is the existence of powerful software tools. Model specification and performance evaluation for the application example have been carried out using TimeNET [7]. Figure 11 shows a sample screen shot of the interface during a modeling session with colored Petri nets.

The upper row of the window contains some menus with basic commands for file handling, editing, and

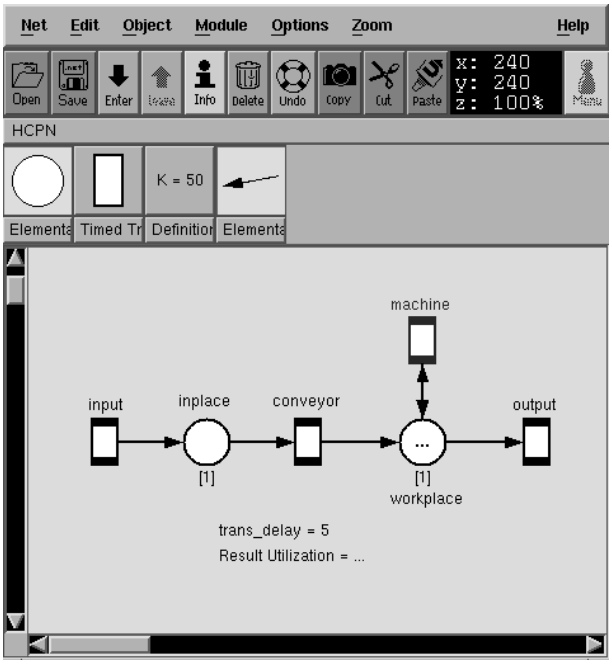

Figure 11: Screenshot of the graphical user interface

adjusting display options. Under the menu item "Module" the analysis algorithms applicable for colored Petri nets can be accessed. Below the icon list the current net class is displayed. The net objects are displayed on buttons at the right. The main drawing area contains a part of the current model. If the object is hierarchically refined (like a substitution transition), double-clicking it displays the refining model. Further information can be found in the references and at `http://pdv.cs.tu-berlin.de/~timenet`.

7 CONCLUSION

This paper presented the modeling and performance evaluation of a subset of a semiconductor fabrication facility. Design decisions should be based on a model-based evaluation to make the design process faster and more exact. A detailed model is developed using stochastic colored Petri nets. The hierarchical refinement is exploited for a modular description. The original specification of identical machines with failures and repairs is aggregated to a simple model without introducing a significant error.

For the performance evaluation, numerical analysis and simulation have specific advantages. They are used for an evaluation of the application example in the paper. Namely the influence of the number of monorail tranport vehicles and the size of in-process buffers (stockers) on the overall throughput and work in process is evaluated. The software tool TimeNET has been used for modeling and evaluating the application example.

REFERENCES

[1] AMD Saxony Manufacturing GmbH: Fab 30 Documentation. http://www.amd.com/locations/fab30.html

[2] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, Series in parallel computing, John Wiley and Sons, 1995.

[3] M. Ajmone Marsan, Stochastic Petri Nets: An Elementary Introduction, in: G. Rozenberg, ed., *Advances in Petri Nets 1989*, *Lecture Notes in Computer Science*, Vol. 424, Springer Verlag, 1990, pp. 1–29.

[4] R. Y. Al-Jaar and A. A. Desrochers, Performance Evaluation of Automated Manufacturing Systems Using Generalized Stochastic Petri Nets, *IEEE Transactions on Robotics and Automation* **6**, 1990, pp. 621–639.

[5] G. Ciardo, R. German, and C. Lindemann, A Characterization of the Stochastic Process Underlying a Stochastic Petri Net, *IEEE Transactions on Software Engineering* **20**, 1994, pp. 506–515.

[6] R. German, *Analysis of Stochastic Petri Nets with Non-Exponentially Distributed Firing Times*, Dissertation, Technische Universität Berlin, 1994.

[7] R. German, C. Kelling, A. Zimmermann, and G. Hommel, TimeNET – A Toolkit for Evaluating Non-Markovian Stochastic Petri Nets, *Performance Evaluation* **24**, 1995, pp. 69–87.

[8] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1992.

[9] C. Kelling, Control Variates Selection Strategies for Timed Petri Nets, in: *Proc. of the European Simulation Symposium*, Istanbul, 1994, pp. 73–77.

[10] C. Kelling, *Simulationsverfahren für zeiterweiterte Petri-Netze*, Dissertation, Technische Universität Berlin, 1995, Advances in Simulation, SCS International.

[11] M. Silva and E. Teruel, Petri Nets for the Design and Operation of Manufacturing Systems, *European Journal of Control* **3**, 1997, pp. 182–199.

[12] H. Westphal and S. Gramlich, On Predictive Supervisory Control of Automation Structures for Semiconductor Fab Using Factory Communications and SPNs, in: *Proc. Int. Conf. on Systems, Man, and Cybernetics (SMC '98)*, 1998, pp. 669–673.

[13] H. Westphal and S. Gramlich, Desing, simulation, and optimal control for the Fab 30 wafer fabrication facility, in: *14th IFAC World Congress* 1999, accepted for publication.

[14] H. Westphal and S. Gramlich, Automation Structure of a Semiconductor Fab using Factory communications, in: *24th Annual Conf. of the IEEE Industrial Electronics Society (IECON'98)*, Aachen, Germany, 1998.

[15] A. Zimmermann, S. Bode, and G. Hommel, Performance and Dependability Evaluation of Manufacturing Systems Using Petri Nets, in: *1st Workshop on Manufacturing Systems and Petri Nets, 17th Int. Conf. on Application and Theory of Petri Nets*, Osaka, Japan, 1996, pp. 235–250.

[16] A. Zimmermann and J. Freiheit, TimeNET$_{MS}$ — An Integrated Modeling and Performance Evaluation Tool for Manufacturing Systems, in: *IEEE Int. Conf. on Systems, Man, and Cybernetics*, San Diego, USA, 1998, pp. 535–540.

[17] A. Zimmermann, *Modellierung und Bewertung von Fertigungssystemen mit Petri-Netzen*, Dissertation, Technische Universität Berlin, September 1997, (in german).

[18] A. Zimmermann and G. Hommel, Modelling and Evaluation of Manufacturing Systems Using Dedicated Petri Nets, *Int. Journal of Advanced Manufacturing Technology* **15**, 1999, pp. 132–137.