Performance and Dependability Evaluation of Manufacturing Systems Using Petri Nets*

Armin Zimmermann, Stefan Bode, and Günter Hommel
Technische Universität Berlin, Institut für Technische Informatik
E-mail: {azi, sbo, hommel}@cs.tu-berlin.de

Abstract

The importance of modeling and performance evaluation for the design of manufacturing systems is obvious. A new modeling method based on Colored Petri Nets is introduced in this paper, which is especially tailored to manufacturing systems. We propose the separate modeling of the manufacturing system's structure and the production routes with dedicated colored Petri nets. After an automatic compilation into a complete model, performance and dependability measures can be obtained numerically or by simulation. The firing times associated with timed transitions can be deterministic, exponentially, or generally distributed.

1 Introduction

Modern manufacturing systems are complex configurations of machines, transport systems, and manual workplaces. To be successful in a rapidly changing market, manufacturers have to be able to change their production program very fast. Therefore, the design process of manufacturing systems has to be accelerated. The economic success is decided by the quantitative properties of the system (e.g. the throughput). The above mentioned problems apply not only to the design of new manufacturing systems, but as well to the re-design of existing ones.

Without modeling and quantitative evaluation techniques it is often difficult to predict the behavior of a real manufacturing system with adequate accuracy. This is especially the case if one takes into account failures and repairs of the system and their effect on the performance measures.

To overcome this problem, many techniques for the modeling and quantitative analysis of discrete event systems have been investigated. Among them, Petri nets are now considered as a powerful tool especially suitable for systems that exhibit concurrency, conflicts, and synchronization. To study the performance and the

*This work has been supported by the German Research Council under grant Ho 1257/4-3.

dependability of a system it is necessary to include the notion of time and probability into the model. This is usually done by associating delays or probabilities with transitions. Stochastic Petri nets (SPNs, [1]) and generalized stochastic Petri nets (GSPNs, [4]) are two popular extensions of Petri nets which have been widely used in the application field of manufacturing (see, for instance, [3] and [17]). Nevertheless, if more than one product is processed by one machine in the model, the machine's model has to be replicated due to the lack of distinguishable tokens.

To cope with problems of this type, colored Petri nets (CPNs) have been introduced [8] and applied to manufacturing systems. Martìnez and Silva showed that colored Petri nets are a powerful tool for the modeling of complex concurrent systems [16]. In order to do so, independent subsystems are identified and modeled in isolation. The resulting submodels are then put together by a fusion of corresponding transitions, and a main model is generated. After eventual simplifications, the structure of the main model can be analyzed. Commands as in programming languages can be associated with transitions, resulting in an *interpreted* net model. It is thus possible to include the control system of a FMS in the model. In [19], Viswanadham and Narahari used colored Petri nets for the modeling of automated manufacturing systems. Based on these models, deadlocks can be found by analyzing the invariants. A colored Petri net model of a manufacturing cell controller is described by Kasturia, DiCesare and Desrochers in [9]. The net model is checked to be live after obtaining its invariants. Additionally, it is "implemented" and "executed" in order to control the cell by exchanging messages with it, and to show its current status. Martìnez, Muro and Silva show in [15], how the coordination subsystem of a flexible manufacturing system can be described by a colored Petri net. The obtained model is embedded into the surrounding levels of control (local controllers and scheduling subsystem), while a terminology based upon the Petri net colors is used for the interaction. Analyzing the model detects deadlocks, decision problems, and gives performance measures that depend upon variations in the modeled system. A *scheduler* is needed to decide indeterminacies, which tries to influence the manufacturing system such that an "optimal" behavior is achieved. The authors of [15] propose an expert decision system for this task, using artificial intelligence methods.

In general, colored Petri nets allow a higher level of modeling, but contain complex definitions of colors, types and variables. These textual inscriptions are part of the model behavior's specification, thus spoiling the understandability of the graphical Petri net model. However, it is possible to omit most of the inscriptions using a restricted class of colored Petri nets especially dedicated to manufacturing systems [21].

In manufacturing systems with a certain degree of flexibility in the production program, there is no notion of production line, rather for each product a *production route* is defined [17]. A Petri net model of a manufacturing system includes both the structural information of the modeled system and the specification of the production routes. Such an integrated model is advantageous for visualization, but there is a need to redefine the whole model even if the production route of a single part changes. The independence of the manufacturing system's structure from the parts to be processed should be reflected in the modeling technique.

To overcome this limitation, a technique for the separate modeling of the production routes and the manufacturing system's structure has been proposed in [22]. Both model parts use dedicated colored Petri nets from [21], and are automatically compiled into one unique model. Based on this method, we present an integrated modeling and quantitative evaluation technique in this paper, facilitating numerical analysis or simulation to obtain the desired measures.

Another approach of separate modeling of production routes and the manufacturing system structure has been proposed before. Villaroel, Martìnez and Silva presented GRAMAN [18], a graphical system for describing manufacturing systems. We briefly describe it in the following to contrast it with our approach.

In GRAMAN, a manufacturing system is modeled by a *plant description* and a description of the *work plans*. While for the latter colored Petri nets are used, the structure of the system is modeled by predefined *building blocks*. An *internal model* is generated from these two descriptions: for each building block, a predefined *subCPN* is assigned to the block and parameterized by its structural relations.

There are many cases, however, where more parameters are necessary for an instantiation of a submodel. Properties of a machine may not only be related to the structure (its general capabilities), but to an actual processing task as well (e.g. the processing time or the mean time until a tool breaks while processing part A). In our approach, library modules are used for the refinement of transitions that model machines etc. In addition to the structural description of such a machine, it contains a model of the possible production paths through its structure as well. Both parts are instantiated and parameterized during the modeling process.

In GRAMAN, the translation of connections between the building blocks is done by fusing transitions of the submodels that represent synchronized activities (e.g. the unloading of a machine is done by a robot), resulting in a *coordination model*. This implies that the number and types of connections from a machine to its surroundings are already known at the time when the submodel is specified. This somehow contradicts the aim of a modular modeling technique. Even more, it leads to submodels that do not strictly describe the structure of the modeled subsystem. The description of how a machine's buffer is loaded and unloaded should be done in the submodel describing the robot etc. that does the unloading. We believe that a model of a manufacturing system should reflect its actual structure, in order to enhance the clearness of the model and to facilitate a visualization of its behavior. Opposed to that, GRAMAN allows the "folding"of equivalent model parts, making the model smaller, but less understandable. Moreover, this prevents a natural specification of buffer capacities. Our approach follows the structure of the modeled system and allows places to have capacities, because this is an important attribute of manufacturing system buffers.

GRAMAN's Petri net models of the work plans hierarchically describe the execution of *orders* at different levels of abstraction. These models are not compiled into the internal model, they interact with the structural model by a "rendezvous-type" mechanism. The production orders can be passed to the Petri net model by a superior level of the manufacturing system controller. In contrast to this, our technique automatically compiles the production route specifications into a complete model, as it is aimed at the performance and dependability evaluation and not the control of a manufacturing system.

As our approach makes use of a restricted class of colored Petri nets [21], it is not necessary to hide the Petri nets from the modeler. Namely, the complex arc and guard expressions as well as the definition of types and variables are superfluous. Therefore, it is possible to model both the manufacturing system structure and the production routes with the same type of dedicated Petri nets, without the need for an additional graphical description language.

The remainder of this paper is organized as follows. In section 2 a manufacturing cell and its GSPN model is presented, showing the difficulties encountered when using uncolored nets for the modeling of manufacturing systems. Section 3 recalls the used specialized modeling method which is subsequently applied to the manufacturing cell from section 2. In section 3.4 the derivation of measures from the obtained complete model is shown. Finally, section 4 provides some concluding remarks.

2 Modeling with Uncolored Petri Nets

Throughout this paper, a simple manufacturing cell is used as an example. The left part of figure 1 shows its layout. The raw and finished parts enter and leave the system through the `In` and `Out` stations, respectively. They can be processed in machine `M1` and `M2`. Each of these stations and machines contains at most one workpiece. The robot transfers the workpieces from one place to another.

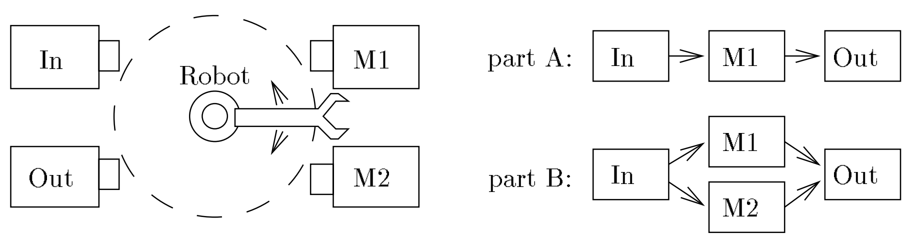

Figure 1: Layout of the manufacturing cell and production routes

The production system behaves as follows: If the robot places a raw part in a machine, an appropriate tool is loaded and the processing starts. While being used, the tool may break and has to be replaced by a worker. There is only one worker available to repair both machines.

In our example, the manufacturing cell processes two different workpieces. Part A is machined by M1. Part B can be machined by M1 or M2 (figure 1, right part). The GSPN in figure 2 models the complete production system behaving as explained above. The process state of part A is represented by a token, passing through the following locations:

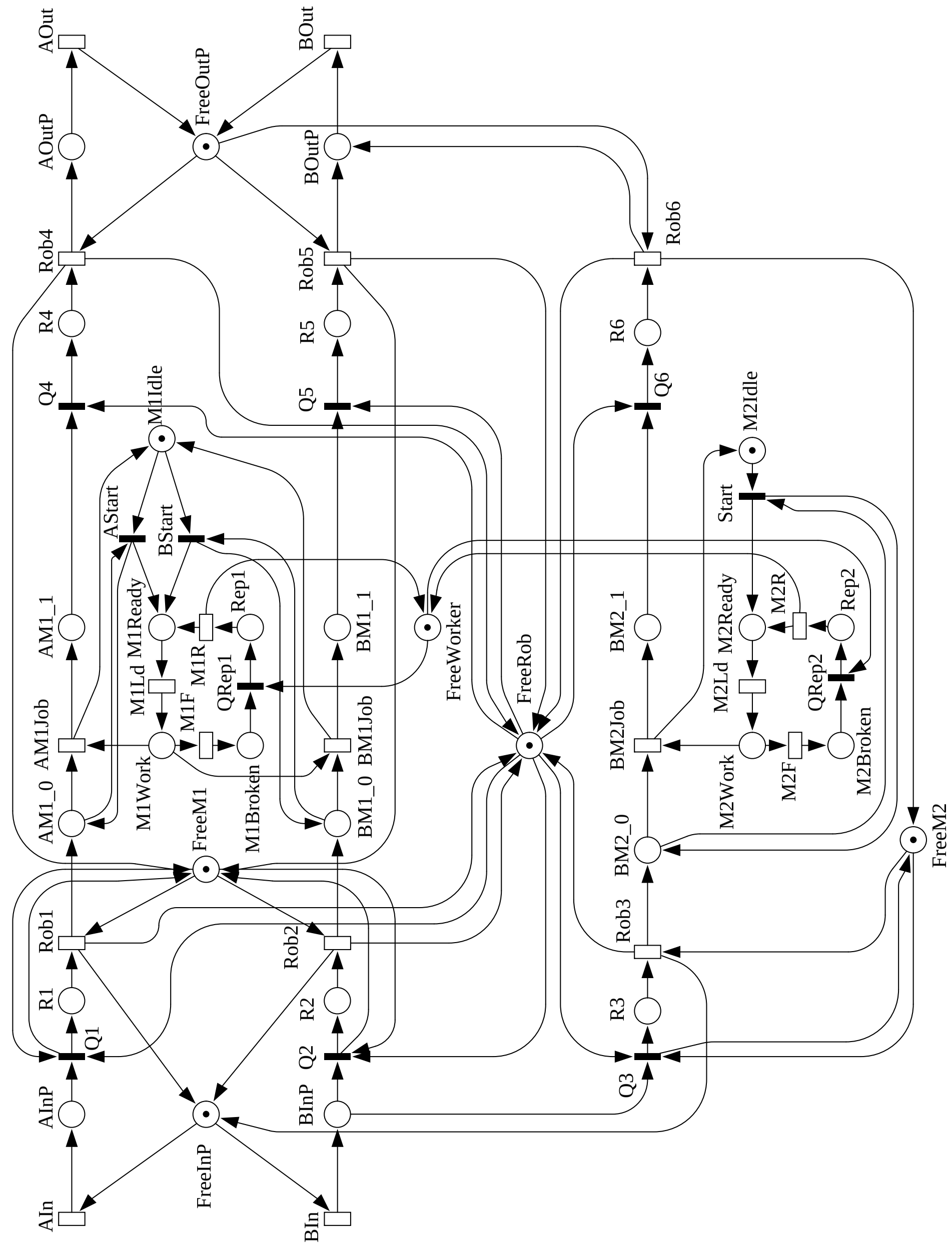

Figure 2: GSPN model of the manufacturing cell

`AIn`	part A enters the cell	`AInP`	input buffer
`Q1`	check for available robot	`R1`	transfer part
`Rob1`	transfer from input buffer to M1	`AM1_0`	raw part in machine
`AM1Job`	machine works on part	`AM1_1`	processing finished
`Q4`	check for available robot	`R4`	transfer part
`Rob4`	transfer from M1 to output buffer	`AOutP`	output buffer
`AOut`	part A leaves the cell		

Part B enters the system by `BIn` and is then transferred to M1 (`Q2`) or to M2 (`Q3`). In the first case the route through the cell is similar to that of parts A:

`BIn - BInP - Q2 - R2 - Rob2 - BM1_0 - BM1Job - BM1_1 - Q5 - R5 -`
`- Rob5 - BOutP - BOut,`

while in the second case machine M2 is used:

`BIn - BInP - Q3 - R3 - Rob3 - BM2_0 - BM2Job - BM2_1 - Q6 - R6 -`
`- Rob6 - BOutP - BOut.`

The limited resources of the cell are modeled by *capacity places*:

`FreeInP`	capacity of input buffer	`FreeOutP`	capacity of output buffer
`FreeM1`	availability of M1	`FreeM2`	availability of M2
`FreeRob`	availability of robot	`FreeWorker`	availability of the worker

The following elements model the behavior of M1 (similar to M2):

`M1Idle`	machine is idle	`AStart`	load tool for part A
		`BStart`	load tool for part B
`M1Ready`	machine is loading tool	`M1Ld`	duration of loading
`M1Work`	machine is working	`M1F`	failure (tool breaks)
`M1Broken`	machine is out of order	`QRep1`	applying for repair
`Rep1`	machine is being repaired	`M1R`	duration of repair

Due to the lacking individual tokens, all buffers that can contain n parts of different type (or state) have to be modeled with n places in order to distinguish between them (see, for instance, `AInP` and `BInP`). Furthermore, a maximum capacity for a buffer cannot be associated with one place in the model, which leads to the need for additional capacity places (e.g. `FreeInP`).

The transitions modeling an active resource have to be unfolded exactly like the places that model buffers. In our example, the robot is described by 12 transitions and 7 places. For every possible action of the robot, a starting immediate transition, one place and a timed transition modeling the transport time is used. This is necessary to guarantee the mutual exclusion of the robot actions. This is a cumbersome and error-prone way, however, to specify which transitions belong to the same resource.

Zurawski and Dillon [23] encountered the same kind of problem and proposed a method to construct uncolored subnets in a systematic way.

In SPNs as well as in GSPNs, the exponential distribution is used for the firing times of transitions due to its analytical simplicity. This is often a good approximation of the real behavior. The processing time of a certain workpiece or a transport delay, e.g., can be modeled more realistically using deterministic times. It has been shown, that the results obtained from models with different distributions may vary significantly [6]. To obtain more realistic results, recent analysis methods [7] have to be utilized for models incorporating non-exponentially distributed firing times.

In general, using uncolored nets leads to models that do not reflect the cell structure, making the model less understandable.

3 Specialized Colored Petri Nets

The consequence of the above mentioned problems is to use Colored Petri Nets (CPNs, [8]), which offer more advanced modeling facilities like distinguishable tokens and hierarchical modeling. 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 new method for the modeling of manufacturing systems is presented.

The main idea lies in the predefinition of two color types, which are adapted to manufacturing systems. *Object tokens* model workpieces inside the manufacturing system, and consist of a name and the current state, e.g. `wheel.raw`. *Elementary tokens* do not have a special color, and are equivalent to tokens from uncolored Petri nets.

Places can contain only tokens of one type. *Object places* are drawn as thick circles. They model the possible locations of workpieces. *Elementary places* are drawn thin. They are used to model states of resources (e.g. a busy machine). Transitions represent possible events, i.e. state changes in the system. Each input and output arc is connected to one place, and only tokens of the appropriate color type can flow through it. Therefore, arcs are drawn thick or thin as well, corresponding to their associated color type.

With this method, the model of the manufacturing cell's structure reflects the layout, which makes it easier to understand. Textual descriptions needed in CPNs for the definition of variables and color types can be omitted, and the specification of the types of places and arcs are implicitly obvious.

To meet the requirements of a modeling technique for manufacturing systems, the structure of the manufacturing system has to be modeled separated from the production routes [18]. We present another method to describe the manufacturing system's structure and production routes separately.

Each of the processing steps of the production routes has to be performed on a machine in the manufacturing system. Therefore, a production route can be thought of as a path through the manufacturing system. This relationship is now reflected on the modeling level. Every transition in a production route model corresponds to a transition in the structural model, indicating that the production route action is executed by the modeled resource. Thus we introduce the term *associated Petri nets* for the production route models.

3.1 Modeling the Structure of the Manufacturing System

Modeling the manufacturing cell described in section 2 with a specialized colored Petri net yields a much more concise and realistic model. Figure 3 shows the top layer of the hierarchical model (the *prime page*).

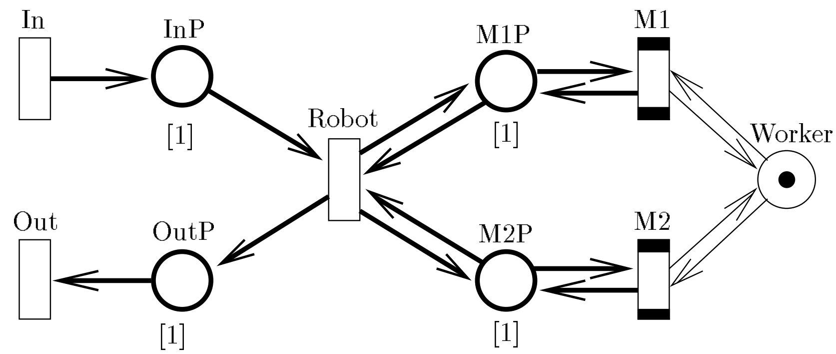

Figure 3: Model of the cell structure

Physically existing locations are represented by object places (with maximum capacities in square brackets):

`InP`	input buffer	`OutP`	output buffer
`M1P`	workplace in machine M1	`M2P`	workplace in machine M2

The elementary place `Worker`, which initially holds one elementary token, represents a globally limited resource: only one machine can be repaired at the same time. All transport actions are performed by the transition `Robot`. There is no need for additional resource places.

The substitution transitions `M1` and `M2` (depicted as ▯) are refined by subpages as shown in figure 4 for `M1`. The port places of the subpage (drawn as dashed circles) are linked to the socket places on the prime page by port assignments. Port places and their assigned socket places are structurally identical (`PP1@M1` means place `PP1` at the subpage of transition `M1`):

```
PP1@M1 -> M1P              PP1@M2 -> M2P
PP2@M1 -> Worker           PP2@M2 -> Worker
```

The state of the machine is described by the location of an elementary token, stepping through elementary places (drawn thin). Initially, the machine is `Idle`. When an appropriate object token enters the machine, the transition `Start` fires (see below). Now the machine begins to load a tool (`Loading, LoadTool`). The firing of `Work` changes the color of the object token in `PP1`, thus modeling the completion of a workpiece. While `Working`, the tool can `Break` and the machine has to be repaired. The transition `StartRep` can only fire, if `PP2` contains an elementary token, i.e., a repairman is available. `PP2` is assigned to the global

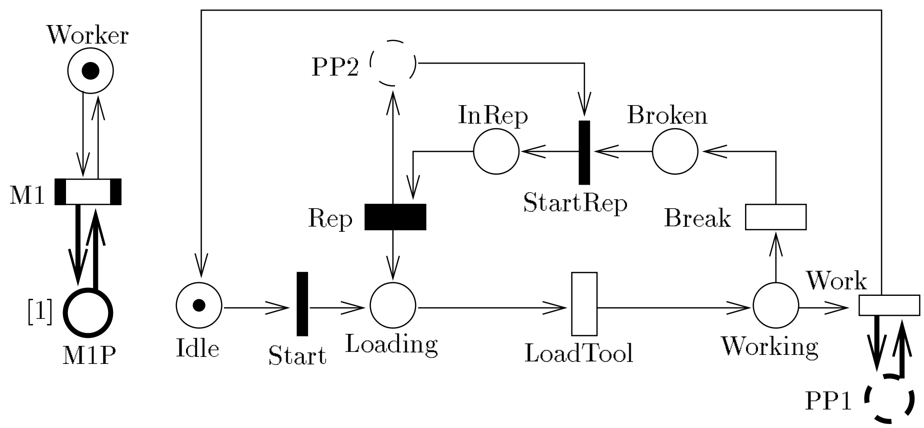

Figure 4: Super- and subpage of machine structure

resource place `Worker` on the prime page. The repairing time is deterministic (transition `Rep`).

Subpages can be taken from a library of templates. A template contains a structural superpage and subpage, with appropriate port assignments. Each time a new subpage is created, the template is copied and instantiated with the name of the substitution transition. A template can be used several times. To avoid name clashes, the elements of subpages are addressed using their name and subpage label, as shown above.

3.2 Modeling of the Production Routes

Given the model of the cell structure, the production routes for different workpieces can be defined (figure 5). They represent paths through the structural model, hence the same places and transitions can be found there possibly several times. The arcs are labeled with the names of object tokens, showing the changes in their processing state.

Alternative routes of workpieces can be modeled using different paths. Square brackets enclose the *version* of a workpiece being in one of the alternative routes (see figure 5). Guards at the starting transitions of each branch decide which path will be chosen, thus implementing a scheduling strategy. If the processing time for a specific workpiece differs from the machine's default, firing time distributions can be specified here.

For each subpage template in the library there is a route template, too (figure 6 for our example). It is used to refine a substitution transition in the production routes. The arc inscriptions and guard expressions in route templates may contain variables. Transition `Start` may only fire if there is a raw part in place `PP1`, which has to be specified in the production subroute (see guard [@PP1 = $x1$]). Each time the appropriate substitution transition is used, the route template is copied

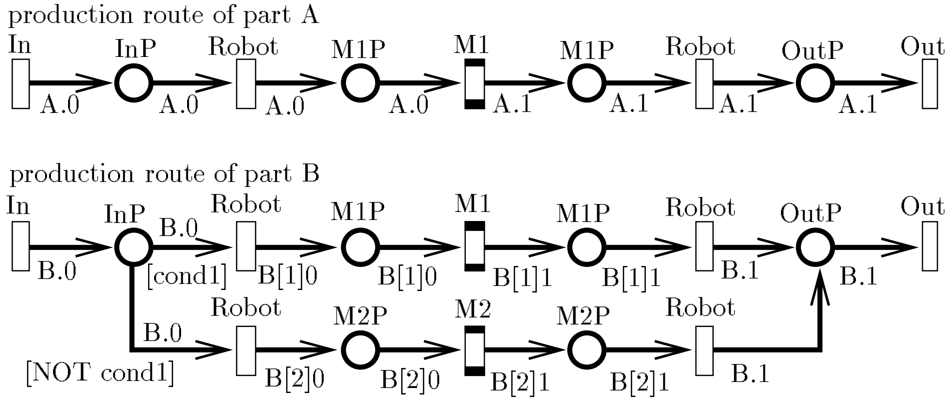

Figure 5: Models of the production routes

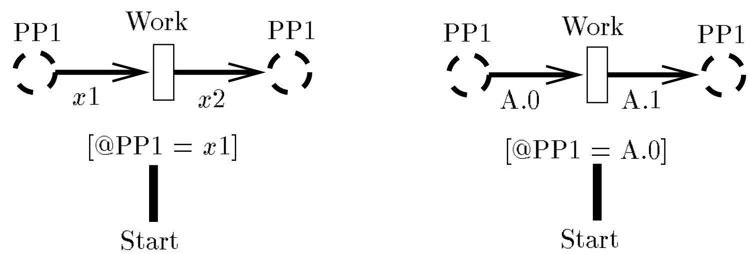

Figure 6: Production subroute template and instance

and all variables are instantiated with names of object tokens. In our example, the substitution transition `M1` is used in the route for A, and the template from figure 6 is instantiated as follows: $x1$ `-> A.0,` $x2$ `-> A.1`.

3.3 Compilation of a Complete Model

Subsequently, the structure and production route nets are automatically merged to create the complete model of the manufacturing cell. During this process, the informations contained in the production route models are added to the structural model. The transitions are enriched with hidden informations, their firing possibilities (see below). This procedure is invisible for the modeler, who only has to construct the model of the cell structure (figure 3) and the production routes for the products (figure 5).

In a colored Petri net, each transition may have several *firing possibilities*. Each of them is characterized by different values of the arc expressions attached to the transition's input and output arcs. In contrast to other colored Petri net modeling techniques, we do not need variables inside these arc expressions. Instead, we obtain all the different firing possibilities automatically from the production route models. A firing possibility in a dedicated Petri net is characterized by an assignment of a token multiset to each input and output arc, a guard (boolean expression) that has to evaluate to true for the firing possibility to be enabled, and a firing time distribution.

For each transition in a production route model, a new possible firing is added to the corresponding transition in the structural model. The guard and the firing time distribution of this firing possibility are copies of the transition's guard and firing time in the production route model, respectively. Empty guards are true by definition. If no firing time distribution is specified in the production route model, the (default) firing time is taken from the structural model. The firing possibility's assignment of tokens to the input and output arcs can be derived from the arc inscriptions in the production route model.

After the compilation, the structural model together with all firing possibilities of the transitions completely describes the behavior of the modeled system and can be evaluated.

3.4 Evaluation

In order to obtain quantitative measures from colored Petri nets, several stochastic extensions have been developed. Zenie proposed colored stochastic Petri nets [20], while Lin and Marinescu introduced stochastic high-level Petri nets [12, 13]. Chiola et al. developed stochastic well-formed colored Petri nets (SWNs, [5]), aiming at the exploitation of symmetries in the model that can be detected at the net level. In order to do so, subsets of the color types have to be specified such that a permutation of a token's color inside its corresponding subset does not alter the behavior of the model. The token colors in our dedicated colored nets correspond to the different types of workpieces (and their various processing states). It is obvious that a manufacturing system behaves differently for each of these workpieces, and thus there are no such symmetries. Moreover, only exponentially timed transitions are allowed in SWNs.

Most of these models as well as the original definition of colored Petri nets [8] focus on homogeneous systems, consisting of identical processing elements performing identical tasks. A (stochastic) colored Petri net is therefore often interpreted as a folded (stochastic) Petri net. A firing semantics is used for colored transitions that allows all its possible firings (equivalent to the "folded" uncolored transitions) to be enabled and fire concurrently. We refer to this as *infinite server* semantics for colored Petri nets, because it is a straightforward extension of the same idea for uncolored nets.

As our approach to the modeling of manufacturing systems intends to follow the modeled systems structure, transitions correspond e.g. to machines or transport systems. Thus, the firing possibilities of these transitions do not model folded machines, but different activities of one machine. Because a machine can only perform one task at a time, at most one of the different firing possibilities of an enabled transition should be fireable. To ensure this, a preselection between the enabled firing possibilities has to be made prior to the further execution. To distinguish this behavior from the one described above, we refer to it as *single server* semantics for colored Petri nets. The behavior of a colored transition with *single server* firing semantics is comparable to the *local preselection policy* defined in [2] for uncolored nets. The *preselection sets* are explicitly given by the colored transitions with single server semantics, which makes the correct specification easier than for uncolored nets.

Although transitions with *single server* semantics are most frequently used when modeling a manufacturing system, there are examples when an *infinite server* semantics is needed. A conveyor belt that transports all parts on it simultaneously is an example, because in that case all firing possibilities of the transition modeling the transport are enabled concurrently. We therefore allow in our models transitions to be either of *single server* or *infinite server* type.

These considerations lead to more compact models that are much better understandable (compare figure 2, 3, and 4). Even though, the reachability graphs of both models are exactly the same (except for the firing time distributions of the transitions modeling the repairing time). After the generation of the reduced reachability graph, numerical analysis techniques (cf. [6]) can be utilized to obtain quantitative measures of the model. It is possible to numerically analyze models that contain transitions with firing time distributions from a wide class of functions that may be immediate, exponential, deterministic and more generally distributed. If no more than one general or deterministic transition is enabled in each marking, a *semi-regenerative stochastic process* underlies the Petri net model. If numerical evaluation is impossible due to the large state space or limitations in the analyzable firing time distributions, simulation has to be utilized. Fast simulation techniques such as parallelization, RESTART [11], and control variates [10] speed up the computation.

In the following, the example manufacturing cell is analyzed and performance measures are derived. Currently, the algorithm to construct the reachability graph directly from the dedicated Petri net models is still under construction. Unfolding the colored Net yields a model similar to the one depicted in figure 2. In contrast to the net shown in figure 2 the firing times of the transitions `M1R` and `M2R` were chosen to be deterministic, because the repairing times are fixed. For the derivation of quantitative measures from the resulting *deterministic and stochastic Petri net*, the software tool TimeNET [7] has been used.

Subsequently, the performance and dependability of the cell is evaluated, comparing different variations of the model and detecting bottlenecks. The following transition rates (1/hour) are used:

Rob$x$	1000	AM1Job	500	BM1Job	500	BM2Job	3333
AOut	500	BOut	500	M1Ld	1000	M2Ld	1000
M1F	1.667	M2F	1.667	M1R	33.33	M2R	33.33

Different transport strategies for the robot were evaluated first. They were implemented by assigning priorities to the immediate transitions Qx. The result for the model with a random choice between the transport tasks is marked 11 in figure 7.

The following strategy turned out to be the most efficient: if there are different transport tasks to be performed, then load M2 first, then load M1, and unload last. The analysis results for this strategy are marked by an asterisk in the same figure. The gain in productivity is not very high, but a change in the strategy does not require any investment.

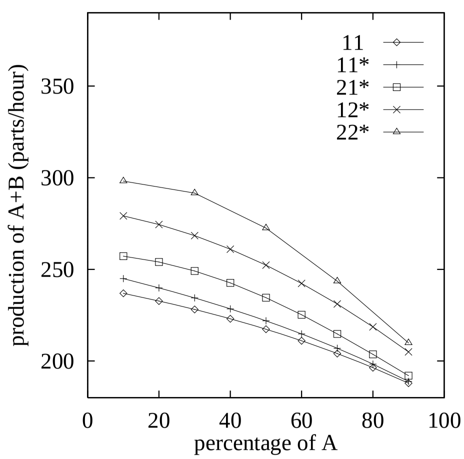

Figure 7: Experiment 1

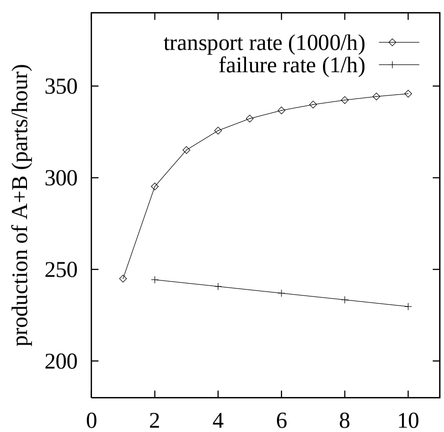

Figure 8: Experiment 2

Moreover, some buffer configurations were compared, evaluating the performance of the system with different proportions of incoming parts A and B and with a constant sum of both. The throughput decreases with increasing percentage of parts A, because they are machined only in M1, while parts of type B may be processed in both machines. Configuration 21 means that the capacity of the input buffer is 2 and the capacity of the output buffer is 1; analogously 11, 12 and 22. The gain in the throughput is much higher when the output buffer capacity is increased compared to increasing the input buffer's capacity.

Secondly, the impact of the transport rate and the failure rate were investigated (figure 8). Speeding up the robot can considerably improve the performance of the system. On the other hand, the gain in productivity must be compared with the high investment costs for a new robot. Taking expensive measures for less machine failures may be useless in comparison to the gain in performance, because the improvements achieved will not be significant.

4 Conclusion

In this paper we investigated modeling techniques for manufacturing systems with Petri nets. Motivated by the problems encountered when using uncolored nets, a specialized modeling method based on colored Petri nets has been introduced. Utilizing this technique, simpler and more concise models are produced that reflect the modeled system's structure. This makes the modeling process easier and less error-prone, and leads to models that are much better understandable. In order to render the modeled active resources correctly, a single server firing semantics for colored Petri nets is used. Therefore, the modeler does not have to use the construct of immediate transitions and additional places to specify the behavior. In addition to immediate and exponentially timed transitions, we allow firing times of transitions to be non-exponentially distributed. Thus, the timing behavior can often be modeled more realistically.

Furthermore, the separate modeling of the production routes and the system's structure is employed. A modification of the routes does not necessitate a complete redesign of the model, thus reflecting that the manufacturing system is independent of the parts being processed. A complete model is derived automatically by a compilation of both model parts. The complete model can subsequently be used to obtain performance and dependability measures using numerical analysis or simulation. The introduced modeling technique has been applied to an example in the paper, showing its usefulness.

Work is currently in progress in order to implement the automatic generation of the complete model as well as the generation of the reachability graph of the dedicated colored Petri nets. The computational cost of the reachability graph generation and the subsequent numerical analysis depend on the complexity of the reachability graph, which may be very high for real application examples. To cope with this problem, we will investigate techniques for the efficient analysis of the dedicated colored Petri net models.

References

[1] 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) 1–29.

[2] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, and A. Cumani, The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets, *IEEE Transactions on Software Engineering* **15** (7) (1989) 832–846.

[3] R. Y. Al-Jaar and A. A. Desrochers, Petri Nets in Automation and Manufacturing, in: G. N. Saridis, ed., *Advances in Automation and Robotics*, Vol. 2 (JAI Press, 1990).

[4] G. Chiola, M. Ajmone Marsan, G. Balbo, and G. Conte, Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications, *IEEE Transactions on Software Engineering* **19** (2) (1993) 89–107.

[5] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad, Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications, *IEEE Transactions on Computers* **42** (11) (1993) 1343–1360.

[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) 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] E. Kasturia, F. DiCesare, and A. A. Desrochers, Real Time Control of Multilevel Manufacturing Systems Using Colored Petri Nets, in: *Proc. Int. Conf. on Robotics and Automation* (1988) 1114–1119.

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

[11] C. Kelling, Rare Event Simulation with RESTART in a Petri Net Modeling Environment, in: *Proc. of the European Simulation Symposium* (Erlangen, 1995) 370–374.

[12] C. Lin and D. C. Marinescu, On Stochastic High-Level Petri Nets, in: *Proc. 2nd Int. Workshop on Petri Nets and Performance Models* (Madison, Wisconsin, 1987) 34–43.

[13] C. Lin and D. C. Marinescu, Stochastic High-level Petri Nets and Applications, *IEEE Transactions on Computers* **37** (1988) 815–825.

[14] C. Lindemann, G. Ciardo, R. German, and G. Hommel, Performability Modeling of an Automated Manufacturing System with Deterministic and Stochastic Petri Nets, in: *Proc. IEEE Int. Conf. on Robotics and Automation* (Atlanta, Georgia, 1993) 576–581.

[15] J. Martinez, P. Muro, and M. Silva, Modeling, Validation and Software Implementation of Production Systems Using High Level Petri Nets, in: *Proc. Int. Conf. on Robotics and Automation* (Raleigh, North Carolina, 1987) 1180–1185.

[16] J. Martinez and M. Silva, A language for the description of concurrent systems modelled by coloured Petri nets: Application to the control of flexible manufacturing systems, in: *Proc. of the 1984 IEEE Workshop on Languages for Automation* (New Orleans, 1984) 72–77.

[17] M. Silva and R. Valette, Petri Nets and Flexible Manufacturing, in: G. Rozenberg, ed., *Advances in Petri Nets 1990*, *Lecture Notes in Computer Science*, Vol. 424 (Springer Verlag, 1991) 374–417.

[18] J. L. Villaroel, J. Martinez, and M. Silva, GRAMAN: A Graphic System for Manufacturing System Design, in: S. Tzafestas, ed., *IMACS Symposium on System Modelling and Simulation* (1989, Elsevier Science Publ.) 311–316.

[19] N. Viswanadham and Y. Narahari, Coloured Petri Net Models for Automated Manufacturing Systems, in: *Proc. Int. Conf. on Robotics and Automation* (Raleigh, North Carolina, 1987) 1985–1990.

[20] A. Zenie, Colored Stochastic Petri Nets, in: *Proc. 1st Int. Workshop on Petri Nets and Performance Models* (1985) 262–271.

[21] A. Zimmermann, A Modeling Method for Flexible Manufacturing Systems based on Colored Petri Nets, in: *Proc. Int. Workshop on New Directions of Control and Manufacturing* (Hong Kong, 1994) 147–154.

[22] A. Zimmermann, Modeling of Manufacturing Systems and Production Routes Using Colored Petri Nets, in: *Int. Conf. on Robotics and Manufacturing* (Cancún, 1995) 380–383.

[23] R. Z. Zurawski and T. S. Dillon, Systematic Construction of Functional Abstractions of Petri Net Models of Typical Components of Flexible Manufacturing Systems, in: *Proc. 5th Int. Workshop on Petri Nets and Performance Models* (Toulouse, France, 1993) 248–257.