A Modelling and Analysis Method for Manufacturing Systems Based on Petri Nets

Armin Zimmermann, Andreas Kühnel, and Günter Hommel

Technische Universität Berlin, Institut für Technische Informatik
Franklinstr. 28/29, Sekr. FR 2-2, 10587 Berlin, Germany
E-mail: azi | andik | hommel@cs.tu-berlin.de

ABSTRACT

A new modelling technique for manufacturing systems is proposed in this paper. Models are constructed with functional blocks known from the field of manufacturing. They are automatically translated into a hidden Petri net. Thus, existing powerful analysis techniques and tools can be applied to the model without knowledge of Petri nets. We emphasise the independent modelling of the manufacturing systems structure and the work plans. Numerical analysis or discrete event simulation of the Petri net model allows to evaluate the performance and dependability of a manufacturing system. A simple application example shows the usefulness of the approach.

1 INTRODUCTION

The design of modern manufacturing systems should be supported by an integrated modelling and performance evaluation method. Such a method based on coloured Petri nets with stochastic timing has been proposed in a previous paper [10]. It introduced a dedicated class of Petri nets especially adapted to manufacturing systems.

Manufacturing systems without continuous states or state transitions can be described as discrete event systems. Petri nets have gained widespread acceptance in the field of modelling and analysis of those systems. Stochastic Petri nets [1] and generalised stochastic Petri nets [3] are two popular extensions of Petri nets which have been widely used in the application field of manufacturing (see, for instance, [2] and [7]). 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. Therefore, coloured Petri nets [5] have been applied to manufacturing systems [6].

This paper deals with the problem that the intended users of the proposed method are design engineers that often have no prior knowledge of Petri nets. Although the class of dedicated Petri nets makes their application easier and better understandable, modelling techniques that have originated from the field of mechanical engineering are sometimes more accepted. In order to make the developed analysis techniques easier applicable, an additional modelling method using functional blocks is proposed. It is inspired by the modelling and simulation tool for manufacturing and assembly planning MOSYS [8].

However, opposed to this tool, there is a clear separation between the description of the manufacturing systems structure and the production route models. This is advantageous for the modeller, who would have to rewrite the whole model in case of a production program change otherwise.

Such a symbolic model is automatically translated into a dedicated Petri net and can be evaluated afterwards. By hiding the underlying Petri net model, the analysis techniques can be used without knowledge of Petri nets. Finally, measures of interest can be obtained from the complete model. The correct behaviour of the modelled system can be checked by qualitative analysis techniques. Quantitative properties are derived by numerical analysis or simulation, showing the performance and dependability of the considered system. The design process becomes more reliable with the opportunity to calculate system properties. An optimisation is possible by evaluating and comparing measures of interest for different model variations.

There are two tools that are related to the method presented here. In MOSYS [8] a manufacturing system is described by a functional specification of its behaviour, tabular work plans and an optional layout description. The functional specification models material flow and production steps. Structural restrictions related to different production steps can thus not be modelled graphically. They occur for example when one machine is used in more than one work plan. Additionally, there is no clear separation between structural and work plan related model elements.

The tool GRAMAN [9] offers separate modelling of production routes and manufacturing system structure. While for the work plans coloured Petri nets are used, the structure of the system is specified with predefined building blocks. An internal model is generated from these two descriptions. The connection of submodels that are created from the building blocks during this translation is done by fusing transitions that represent synchronised activities. Unfortunately, this means that those connections have to be known at the time when the submodel is specified. This somehow contradicts the modularity of the specification. The modeller has to learn two different description languages before using the tool. Additionally, it is not possible to specify machine properties that depend on a processing task.

The paper is organised as follows. Section 2 introduces functional blocks as a description technique. The subsequent section 3 illustrates its application using an example. In section 4 the generation of a Petri net model from the functional block description is described. Further analysis steps lead to a performance and dependability evaluation of the model. This is briefly discussed in section 5. Finally, some concluding remarks are given.

2 FUNCTIONAL BLOCKS

Petri nets have evolved into a powerful modelling mechanism for manufacturing systems in the scientific community. Their industrial application is, however, still very limited. Manufacturing engineers are the intended users of the modelling environment that is presented here. The basic modelling elements therefore resemble the ones used in application domain specific tools like MOSYS [8].

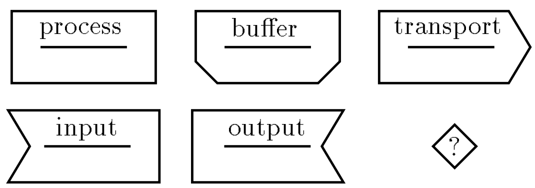

Figure 1: Modelling elements

Figure 1 shows the basic blocks that are available in the proposed modelling technique. The six symbols have the following meaning:

- Rectangles (`process`) are used to model machines and other resources that might change the processing state of a part.
- A box with cut lower corners (`buffer`) is used for buffers and all places where parts can be stored.
- Boxes with an arrow-like shape at the right (`transport`) model transport facilities of all kinds, e.g. conveyor belts and automatic guided vehicles.
- Boxes with the two shapes that are shown on the left of the lower part of figure 1 (`input` and `output`) depict system in- and outputs. At these places parts come into existence (or vanish) from the point of view of the model.
- The last symbol in the figure has a special meaning and can only be used in work plan models (see section 3). Alternatives in a work plan are modelled using it.
- Arcs (not shown in the figure) specify paths between resources in the manufacturing system structure or the sequence of orders in a work plan.

This set of model elements is sufficient to describe the basic manufacturing system elements and their functionality. MOSYS provides additional symbols for *test*, *assembly* and *disassembly* operations. These functions are special cases of the machine symbol here. Using the Petri net generation technique presented in section 4, other symbols with special functions could be added easily if needed. The possibility of specifying alternatives in a work plan together with the corresponding attributes graphically is new.

Taking a closer look, every basic modelling element corresponds to a group of actual basic blocks. There might be actual blocks of the element *transport* modelling a conveyor belt, a transport robot, or an AGV. All actual blocks of a basic element fulfil the same task, but in different ways.

During the process of modelling a manufacturing system, we might want to create a model of a machine. One actual block from the set that corresponds to the `process` symbol is therefore selected. The next step is the instantiation of the selected machine. While the type of the machine is given by the selected actual block, we still have to supply some attribute values before the new machine model is created. Every actual basic block has a corresponding set of attributes. An attribute has – like a variable in a programming language – a name, a type and a value. The set of attribute names and types is defined by the actual basic block. Section 3 explains the later usage of the attributes during the subsequent generation of a Petri net model.

The main difference to other modelling techniques is the separate modelling of the manufacturing systems structure and the work plans. The motivation for a separation of the two model parts has been given in the introduction. Both model parts use the same building blocks as shown in figure 1. This is important for the modeller who only needs to learn one description technique. Opposed to that, GRAMAN uses different modelling formalisms for the specification of the manufacturing systems structure (building blocks) and the production plans (Petri nets).

Model elements in a structural model and the arcs connecting them describe the resources of a manufacturing system and the possible material flows. The horizontal line inside the symbols (see fig. 1) separates structural and functional information of a model element. The attributes of model elements in a structural model describe the work plan independent, i.e. general characteristics of a machine. Examples are the default processing time, the mean time to failure, or the machine setup time. It is obvious that there is no functional information in the structural model. Therefore, the space below the separating line in the symbols is empty in the structural model. The name of a resource is written in the symbol above the line. There must not be more than one resource with the same name.

In addition to the structural model of a manufacturing system we need to specify the production steps with one work plan model for each part. A work plan describes the sequence and attributes of all production steps needed to produce one part. It constitutes a path through the corresponding structural model. Only the machines (and the possible material flows) that are available in the structural model can be used in a work plan.

A model element in a work plan model specifies one production step. Inside the graphical symbol, there is the used resource (e.g. the machine or transport facility) described with its name from the structural model. The name of the processing step can be found below the line. From the graphical description it is then clear *what* processing step is done *where*. Further information about *how* it is done is given by the attributes of the model element. It should be noted that the work plan attributes describe the actual production step. Examples are a processing time that differs from the default value or the time to adjust the machine for the processing step.

Arcs between model elements in a work plan model describe the order in which the processing steps have to be executed. To each arc there is the name of the part being produced and its current processing state attached. Not more than one successor is allowed inside a work plan model. Choices between two or more different possible work plan parts are the only exception. They are modelled with the question mark box shown in figure 1. There might be several arcs going from such a box to subsequent model elements (processing steps).

Assembly and disassembly operations are special cases that can be modelled as well. Their description is omitted here.

3 AN EXAMPLE

A simple example is shown in figure 2. It is modelled using the elements described in the previous section. As described above, the presented modelling technique uses separate graphical models for the structure of a manufacturing system and its work plans. The first figure shows the structure and resembles the actual layout of the modelled system. This makes the model better understandable.

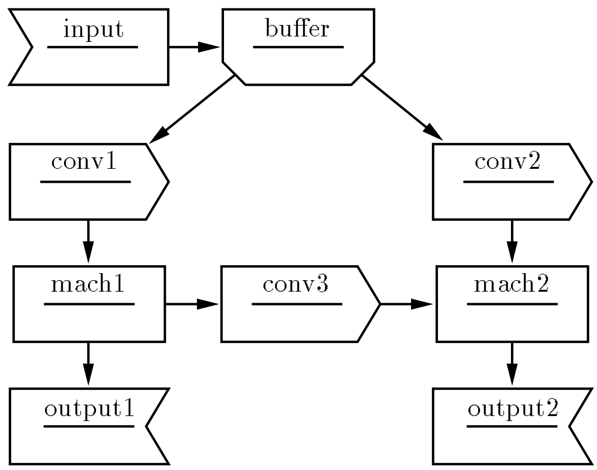

Figure 2: Structural model

In the figure there is one model element for each resource in the manufacturing system. The shapes of the boxes show the type of resource, as described in section 2. Additionally, the system's interaction with the outside world is modelled with the boxes named `input`, `output1` and `output2`. Arriving parts are stored in a `buffer`. Two conveyor belts (`conv1` and `conv2`) take parts from the buffer and transport them to one of the two machines `mach1` and `mach2`. There is one system output connected to each machine. A third conveyor belt `conv3` connects machine 1 and machine 2 for parts that have to be processed by both. The description of attributes and their values for the model elements is omitted here. One example will be covered in the subsequent section.

Please recall that in the structural model there is only the name of the actual resource written inside each symbol. An arc between two model elements depicts a possible material flow between them. It should be noted that such a model lacks some of the modelling power of Petri nets. For instance, it is not clear from figure 2, whether it is possible for a part to bypass machine 1 without being loaded into it. A similar Petri net model would indicate the inner structure of a machine.

In addition to the structural model shown above, we need a description of the work plans of all parts being produced with the considered manufacturing system. This is done with one model for each part. Figure 3 shows the work plan model of a part `A`.

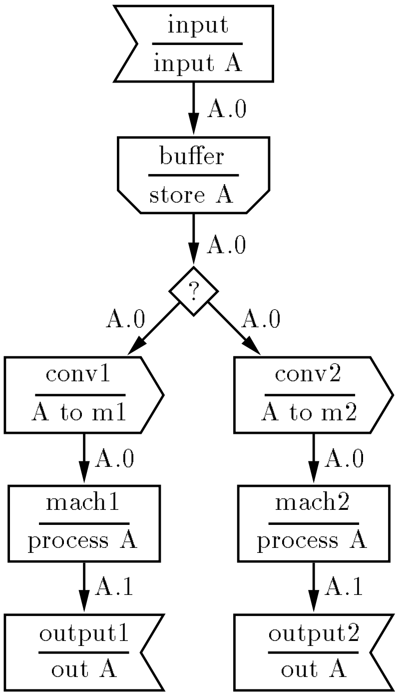

Figure 3: Work plan model of part A

It describes the path of the parts through the structure. Each box contains the name of the resource used and an identification of the processing step, separated by a horizontal line. Only resources and their connections that are defined in the structural model can be used here. The attributes and their values are again omitted.

The part name and its current processing state are written beside each arc, separated by a dot. Raw parts (`A.0`) arrive through the input and are stored in the `buffer`. There are two alternative paths afterwards. The raw parts can be transported to `mach1` by `conv1` or to `mach2` by `conv2`. An alternative symbol is therefore used. Each outgoing arc from an alternative has two optional attributes: a boolean function and a probability. Thus, scheduling decisions and quantitative distributions between alternative paths can be modelled. After the processing of `A.0` by `mach1` or `mach2`, the new processing state is called `A.1`. The finished parts then leave the system through one of the outputs. The work plan of the second part `B` is omitted.

4 PETRI NET GENERATION

In order to use the analysis methods for manufacturing systems presented in [10], the functional blocks explained above have to be translated into a Petri net model. The set of token colours and corresponding place and arc types is predefined in this model class. Different models for the structure and work plans are used in the same way as proposed here for functional blocks. Transitions and places in a work plan model form a path through the structural model, thus describing the processing steps and the material flow of a part. Please refer to [10, 11] for the details.

During the modelling process using the mentioned class of dedicated Petri nets, a library of model templates can be used. Each template describes a set of similar manufacturing system resources. It consists of a structural model and a generic work plan model. This set of library models is now used to generate a Petri net from a functional block model. In general, each functional block is automatically substituted by a submodel taken from the library. To ensure that there is a matching template for each functional block, the set of usable blocks is automatically derived from the library at the startup of the software tool. If new Petri net templates are added to the library, more functional blocks are available.

Figure 4 shows the structural part of a library template model. It is used to describe simple machines being subject to failures and repairs. Expressions in square brackets at transitions are guard functions,

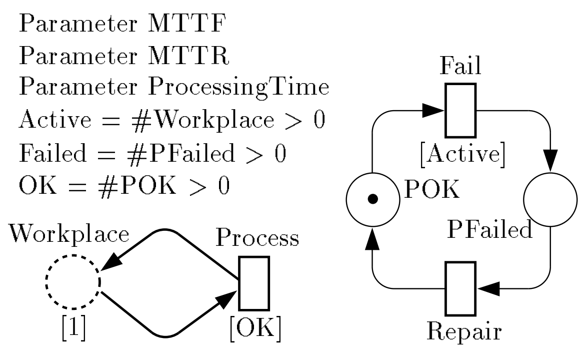

Figure 4: Library template example

that have to be true for the transition to be enabled. At places, they describe a capacity. If the transition `Process` fires, it takes a token from place `Workplace` and puts back a token in a different processing state. `Active`, `Failed` and `OK` are definitions that make the model easier readable (`#Workplace` is the number of tokens in place `Workplace`). The remaining parts of the model should be clear.

Please note the three parameter definitions at the top of the figure. They denote parameters that have to be supplied by the user during the instantiation of the model and are used e.g. for firing times of transitions. Every attribute of a functional block corresponds to a parameter in a Petri net template. The attribute value supplied by a functional block sets the value of the library parameter during the translation.

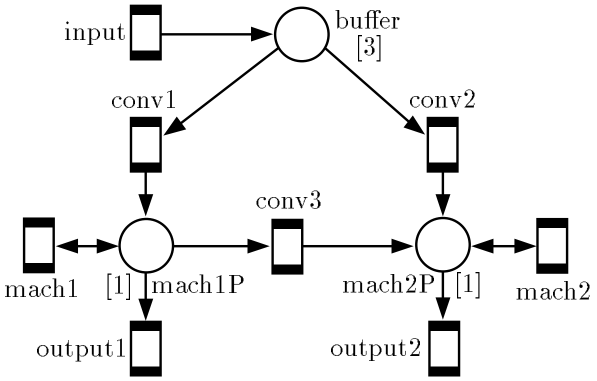

Figure 5: Generated Petri net

Figure 5 shows the structural model after translation of figure 2. It contains only the top level of hierarchy; transitions with black bars are *substitution transitions*. There is a subpage associated to each of them. Every subpage is an instantiated library template. There is a set of *connection rules*, which can be specified in the library. They define how the substitution transitions and places have to be connected during the translation. Opposed to GRAMAN, we propose to fuse places instead of transitions.

5 FURTHER ANALYSIS

A Petri net model is generated for the structural model as well as for each work plan model as explained in the previous section. Those models are hidden from the modeller and are automatically analysed as shown in the following.

One complete model has to be compiled from the different model parts afterwards. This is done automatically by adding a *transition table* to each transition in the structural model. Such a table contains the descriptions of all firing possibilities, including the input/output behaviour, the firing time distribution, and guard functions. Every time a transition appears in a work plan model, a new firing possibility (row) is added to the transition table. Thus, the resource constraints that are imposed by the structure and the synchronisation of the work plans are compiled into one model. The resulting model contains all necessary information from the structural and work plan models. Please refer to [10] for the details of the compilation.

The resulting model can then be used for a performance and dependability evaluation. Reachability graph generation together with a direct numerical analysis or simulation can be used for that task. Both algorithms are available in the software tool that implements the methods presented here.

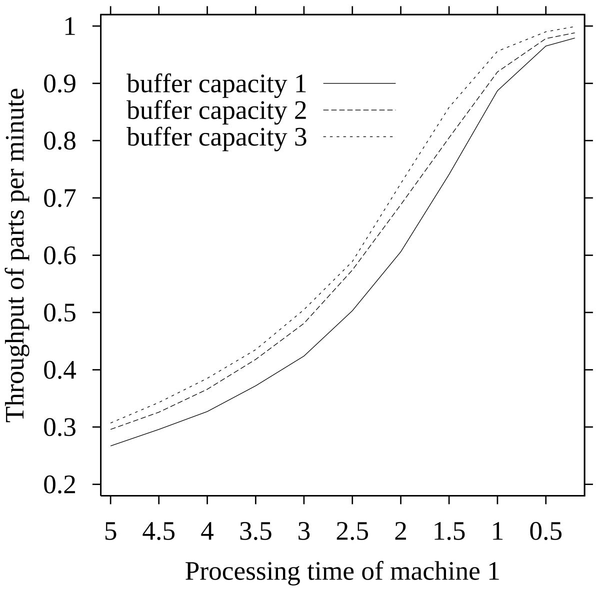

Figure 6: Throughput versus processing time

Two experiments show possible results of a performability analysis. The impact of the processing time of machine 1 on the overall throughput (measured in parts per minute) is shown in figure 6. The capacity of the two buffers at machine 1 and 2 is an additional parameter. We can see that the processing time of machine 1 is crucial in the range from 1 to 3. Smaller processing times do not increase the productivity very much.

The second experiment shows the influence of the mean time to failure (MTTF) for both machines on the throughput for different buffer capacities. It is clear from the figure that the MTTF should be no less than 100, and that the buffer capacity should at least be two.

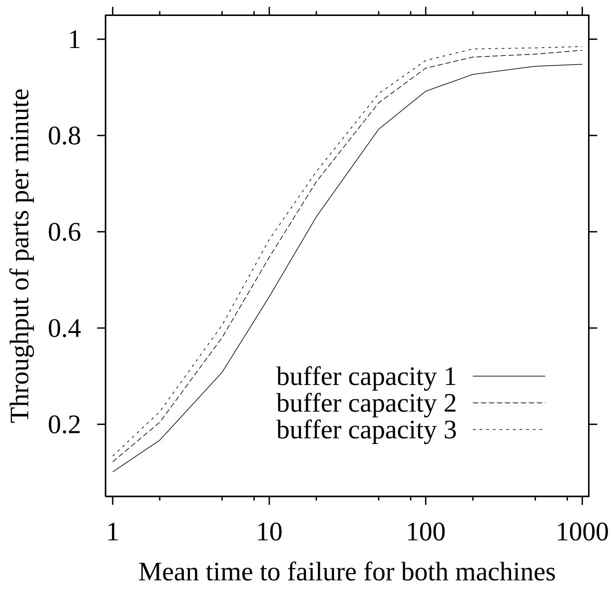

Figure 7: Throughput versus mean time to failure

6 CONCLUSIONS

In this paper a new modelling technique for manufacturing systems is presented. We propose the use of functional blocks for this application area, which are better understandable for manufacturing engineers than more formal description languages like Petri nets. An important feature is the independent modelling of the manufacturing systems structure and the work plans.

The resulting models are automatically translated into a hidden Petri net model. This is necessary to make use of the already existing analysis algorithms for them. After the subsequent numerical analysis or discrete event simulation, measures of interest can be obtained. The performance and dependability of a manufacturing system can thus be evaluated.

The method could only briefly be explained in the paper. It is presented together with a small application example. All techniques described have been implemented as an extension of the software tool TimeNET [4]. This includes a customised graphical user interface for the functional block 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] 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).

[3] 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.

[4] 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.

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

[6] J. Martínez, 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.

[7] M. Silva and R. Valette, Petri Nets and Flexible Manufacturing, in: G. Rozenberg, ed., *Advances in Petri Nets 1989*, *Lecture Notes in Computer Science*, Vol. 424 (Springer Verlag, 1989) 374–417.

[8] G. Spur, K. Mertins, B. Wieneke-Toutaoui, and M. Rabe, MOSYS – a planning system for manufacturing and assembly planning, in: *Simulation in Manufacturing (SIM5)* (Bedford, 1989, IFS) .

[9] J. L. Villaroel, J. Martínez, 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.

[10] 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) 235–250.

[11] A. Zimmermann, K. Dalkowski, and G. Hommel, A Case Study In Modeling And Performance Evaluation Of Manufacturing Systems Using Colored Petri Nets, in: *Proc. of the 8th European Simulation Symposium* (Genoa, Italy, 1996) 282–286.