DISCRETE EVENT SYSTEMS – PETRI NET-BASED MODELING AND SIMULATION IN THEORY AND PRACTICE

Matthias Becker[1]

[1]FG Simulation und Modellierung,
Universität Hannover,
Welfengarten 1, 30167 Hannover, Germany

xmb@sim.uni-hannover.de (M. Becker)

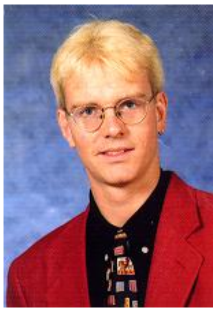
Abstract

The theory of modeling formalisms for Discrete Event Systems has a long history and is well developed, many algorithms for modeling and efficient analysis of the modeled systems exist. However in many practical applications or commercial software, the theory is not used. The reasons are manifold. The question is, whether the theoretical concepts are not suited for practical applications, or whether the problem lies in the proper transfer to practice. Other problems lie in the sometimes missing flexibility of theoretical models, to some extend in missing good software that would enable the practical use of such models. In this work we review Petri net based methodologies with regard to their applicability in practice, and try to understand why many aspects of the theory of modeling and simulation do not find their way into practice. We identify crucial factors such as support of complex models and hierarchic modeling capabilities. These factors not only concern the modeling methodology, but also need to be implemented in a software tool. The availability of a software supporting a modeling concept is another important factor. The software should also have an adequate and appealing graphical representation because at the end, the practitioners have to be convinced to 'buy' the theoretical concept, and that will only be the case, if decision makers can recognize 'their' system easily. Furthermore we survey a number of papers about application of Petri nets to find out to which extend these applications are practical ones, i.e. whether the applications are of academic nature, proof of concept, toy size or inside a productive environment.

Keywords: Modeling, simulation, theory, practice

Presenting Author's Biography

Matthias Becker is involved in both practical applications and fundamental research in the area of modeling, simulation and optimization of discrete event systems.



1 Introduction

Modeling and simulation (M&S) can be approached from two opposed sides. On the one hand modeling and simulation can be approached from a purely theoretical side. The roots of many concepts of M&S stem from theoretical computer science. Especially modeling formalisms such as Petri nets have their origin in automata theory and neighboring fields.

The other side is the wide spread use of simulation in practice. Many people claim to do simulations and they use something informal (such as programming code) to formulate the simulation model which makes validation difficult. Quite often, the theoretical foundations of M&S are neglected to a great extend, even basic things such as proper statistical analysis of stochastical simulations is not done (see remarkable, extensive studies of Pawlikowski et al e.g. in [1]). If people need a model of some system's behavior and a simulation of it, often the rich theoretical results of different formal modeling methodologies are ignored and no formal model is created. Instead people just program a piece of code delivering some numbers and call it a simulation.

E.g. a nice formal methodology for behavior modeling is Petri nets. Petri nets are very easy to learn, and provide a rich theoretical background for analysis of the behavior, plus possibilities for performance evaluation. Even pure simulation is possible, which relieves the modeler from nearly all restrictions usually imposed on the PN model (e.g. most analytical performance evaluation methods allow only exponential distribution of processing times).

Often it is the lack of software, that offers all needed features and looks 'professional' enough for use in practice. Academia does not have the resources to develop tools that are able to compete with commercial tools. E.g. the import of diverse commercial file formats is usually not possible with academic software.

Apart from the functionality, there are other, more psychological reasons that let academic tools not be accepted in practice. One simple but nevertheless important point is the way academic tools look, or how a presentation of a formalism looks.

With a Petri net model (only boxes, circles with dots and arcs, often enough in black and white), no one should meet a decision maker in industry and try to convince him to use Petri nets or hire the 'theoretician'.

Even very good Petri net software (that still looks like Petri nets) failed when people tried to commercialize it, e.g. DesignCPN [2]. On the other hand commercial software often claim to be based on queuing theory because of the fact that they have implemented something in their system and named it a 'queue'.

This gap of M&S between theory and practice is a subject appearing frequently in literature or panel discussions e.g. [3].

In this work we give examples of realistic applications tackled with formal modeling paradigms, and why a certain study succeeded or not, what were the problems etc. We discuss reasons for how the situation is now, and what can be done for academic people to have more success with application of well founded M&S concepts in practice.

We want to find out if it is some kind of dilemma (whether there are some reasons which exclude a formalism to be at the same time universal enough for meeting the requirements of practice AND for being theoretically well founded enough to have success in both M&S theory AND practice) or if there is a solution to the situation.

2 Related Work

In [4, 5] a comprehensive study can be found, which software for Petri nets is suited for a certain kind of real application (related to modeling with UML). This study comprises both academic as well as commercial tools. A first selection is based on criteria such as having a hierarchy concept, support of colored nets, having a GUI with model editor, simulator and analysis methods. Although the criteria seem quite reasonable and essential with regard to real world applications, during this first round a vast majority of the considered 91 tools have been excluded. Only ten tools have been examined more closely.

Among them was DesignCPN, a tool for colored Petri nets [2]. The tool is very good for a scientific application, but seemed not to fulfill all the demands of real applications. Although Jensen [2] claims that his tool can handle large realistic scenarios, this seems only possible with knowledge of internals of the tool (e.g for their VLSI example they made a special extension to the tool). In one of our studies, a realistic semiconductor model (data from [6]) even took minutes to load into the tool, not to speak from changing or analyzing/evaluating the model. [4] indicates that the complexity of models is a general problem. Most tools allow only thousands of places (DesignCPN: 2000), which is not enough, considering that e.g. a semiconductor production plan has several hundreds of steps, each step constituting a sub-model consisting of a number of places.

The same semiconductor model we tried to evaluate with our own combined Petri net/queuing net tool [7], but this also lead to long simulation times and there were difficulties to validate the model. Also, no advantages of the Petri net theory (like qualitative analysis methods e.g. for deadlock avoidance) have been used.

The introduction of an open model description format such as PNML [8] is a good step, so that one tool can be used to apply to one model algorithms that the other tool does not provide.

Another deficiency of tools maybe that the theoreticians concentrate on the e.g Petri net algorithms, but neglect helpful methodologies from other communities. E.g. every Petri net tool should have a stochastic simulation engine, but often that is not the case. As Pawlikowski [1] found out, a vast majority of research papers using simulations did not obey basic but fundamental statistical principles of good statistical practices.

3 Capabilities of Formal Modeling Concepts

Some academic tools offer outstanding and useful features. With Renew [9], nets in nets are possible offering a reusability of models in the purest object oriented sense. Even with uncolored Generalized Stochastic Petri nets (GSPN) complex circumstances such as online simulation (a system simulates itself for scheduling decisions) can be done, see e.g. [10].

And all those complicated mechanisms are embedded in a well-founded framework, which makes validation of the models much easier than in the alternative case, when the system is just implemented in some programming language.

4 The Complexity Issue

Models in academia often have little size, so that test and verification of new algorithms for analysis applied to the model is easy.

In practice the pure size of the real system under study may be the problem. Especially algorithms working on the complete state-space of systems e.g. reachability analysis, Markovian analysis) will not work on models consisting of hundreds of components/places and several hundreds of parts/tokens circulating in it.

A good test-case for complexity is provided by the Modeling and Analysis for Semiconductor Manufacturing Laboratory (MASMLAB) at Arizona State University. They provide anonymized real production data from semiconductor manufacturing. To show the complexity of the model, we give a brief description of the basic features of the first dataset:

This data set contains the description of the production of non-volatile memory chips. There are two different products, the first needs 210, the second needs 245 production steps. For both product routes 28 machines are necessary, with 87 different configurations in the first route, and 103 configurations for the second route. The model includes loading, unloading, processing and travel times of wafers, setup/down times for machines, as well as scrap and rework probabilities.

In order to test the applicability of Petri net based tools for complex systems we undertook two studies.

First we undertook a study with a combined Petri net queuing net simulator [11]. The model is described in XML, and the tool supports hierarchic models. Since the production steps can be modeled similarly, only with different parameters of the machine/need of different resources, the hierarchy could be exploited efficiently and made the modeling part easy.

Figure 1 shows the structure of one production step modeled as Petri net.

Analysis/simulation of the model was more difficult. Markovian analysis is not feasible due to the size of the state-space.

Simulation worked satisfactory. In order to get results nearly one day of simulation was needed. That is OK for factory (re-) planning, but is not fast enough for e.g. online scheduling. Other disadvantages of this approach are that it was a prototype piece of software with no nice graphical representation.

In another study we used DesignCPN, but that tool seemed very impractical. We generated the model automatically from the data. The resulting CPN model was too large to be handled reasonably be DesignCPN. Already the loading of the model from file to the tool took minutes, it was not possible to work with the GUI, that is to change or analyze/evaluate the model.

5 The Abstraction Issue

When building a model, it is important to decide about the level of abstraction. Abstraction means that only the relevant features of the system under study are considered. One might think that this might easily lead to wrong models/wrong results (since some parts of the real system are omitted). Actually the opposite case is true: Abstract systems are more robust, too detailed models often show chaotic behavior. That is the case because one can never measure the modeled system one hundred percent correctly, so a detailed model will multiply the error, if a small incorrectness of a parameter is modeled too detailed (e.g. considering only one value instead of a distribution, or modeling packets in communication systems packet per packet, instead of modeling a packet rate). By using a more abstract model, it will be more likely that the abstraction also captures a larger set of models with parameter intervals, instead of only one specific parameter.

To illustrate the difference, we conducted a real world study by using two approaches in parallel [12]. The real world study was to assess different configurations of a shiplift to be rebuild. We used two different formal modeling methodologies and two tools to provide insights, which impact of the formal modeling methodologies, the chosen tool and the level of abstraction have on the success of the real world model.

Two alternatives for reconstruction of a shiplift are evaluated. At the moment, the shiplift consists out of two long chambers. One chamber is to be rebuild. Instead of rebuilding it in its original length, a shorter and cheaper chamber could also be built.

The Petri net model has been chosen because Petri nets are a universal modeling language that allows a quick construction, validation and evaluation of models of arbitrary systems.

5.1 Detailed Model

Often modelers tend to build a very detailed model. A common belief is, that more details result in more accurate results. Furthermore building a detailed model requires less thinking and less understanding of the real

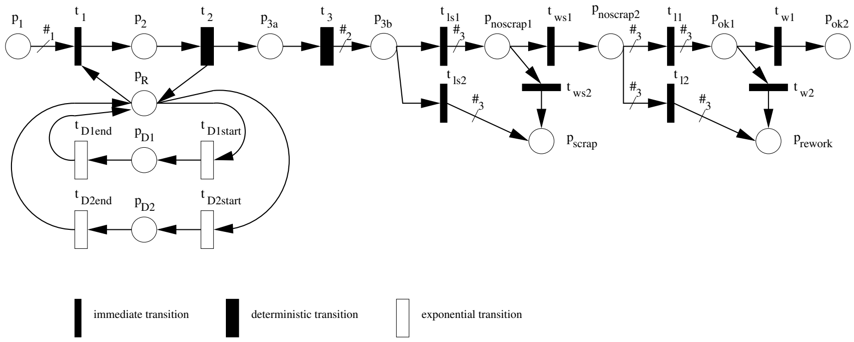


Fig. 1 Model of a machine including maintenance, failure, rework, scrap

system, since in a detailed model, there almost is a one to one correspondence of objects and relations compared to the real system.

An abstract model requires a deeper understanding of the components and the functions of the real system. Only with enough understanding, the modeler can decide to abstract from some features of the real system and to model several features by maybe only one parameter in the model.

Building the detailed colored Petri net model in figure 2 was time consuming and the result is a model that is not so easily to understand (e.g. by shiplift engineers or politicians). One can see that a lot of formulas and arc inscriptions are used, and are essential for the understanding of the model. This is in contrast to the idea that one advantage of Petri nets is a graphical notation that is easily to understand and which adheres to the principle of locality. The timing concept in Renew is not so developed, so that there was quite some work to be done manually to be able to use the correct stochastic distributions and also to have a state of the art stochastical analysis of the results. Also the verification and the calibration of the model was a time consuming activity, caused by the detailed model.

5.2 Abstract Model

Parallel to the colored Petri net model a second model has been developed using (uncolored) Generalized Stochastic Petri Nets (GSPN). The software used was TimeNet (now available in version 4.0). Development of this model required more initial considerations how to represent the reality adequately with the more restricted means of uncolored Petri nets (e.g. different ship lengths, which can be done straight forward in colored nets by adding an attribute 'length' to the tokens), but after that, building, validating and calibrating the model did not take long.

6 Are There Real Practical Applications?

6.1 Survey

In order to get some facts about practically relevant applications of formal modeling methodologies, especially Petri nets, we had a look at the 'Seventh Workshop and Tutorial on Practical Use of Colored Petri Nets and the CPN Tools October 24-26, 2006'. The results of this non-representative survey are the following:

- Sixteen papers of this workshop can be found on the web. From these 16 papers
  - none was an application in a productive environment.
  - Nine where proofs of concept or an initial study/first step of applicability of CPN in a certain field or inside a larger project.
  - Another two were academic case studies of small size.
  - The rest were methodological papers or introductions of new tools.
- Often Petri nets were used somewhere below the surface, e.g. in the backend of some software.
- Many papers reported difficulties with given software, or had problems with the problem size.
- Further applications (65) including some from the workshop above are listed on [13].

6.2 Applicability of Petri Nets

It seems that despite the long existence of various Petri net dialects there exists no field of application where a Petri net tool can just be plug in and non specialists can start to use it.

Most applications today seem to be research applications, that means special researcher have to analyze the problem, whether it can be tackled with Petri nets or not, then make a tools survey, find a software that fits nearly all needs.

Often it is a feasibility study that answers the question: Can this problem be solved with Petri nets or not?

For some time, PN were fashionable in Software Engineering (SE) in the past, but not any more today. State of the art in SE is to use UML for modeling, although a Petri net based formalism would offer lots of additional analysis methods. That is most important for design of high-performance and secure software, and could be

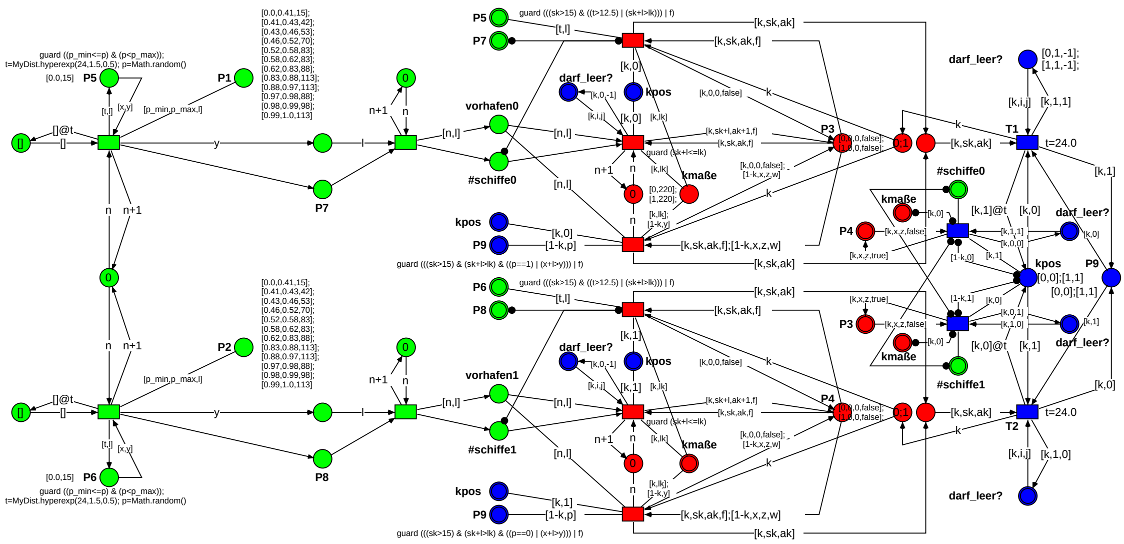


Fig. 2 Model of the shiplift with colored PN of Renew type

done in an integrated approach using a more formal approach, e.g. such as one based on Petri nets [14].

A similar situation we face in protocol specification, which uses mostly SDL based methods. PN would again offer the same functionality, plus analysis methods such as deadlock detection, which is quite important in protocol design.

One field where Petri nets are relatively successful is business process modeling [15].

7 Keys to (more) success?

7.1 Visualization

Despite Petri nets are a methodology with inherent graphical representation, the representation is rather spartan (circles, dots and rectangles) and surely not appealing enough for non experts, decision makers etc.

When trying to advertise Petri nets for deployment in a productive setting visualization is important in order to have an easy! presentation possibility for the intended customers to-be. Only if the decision maker recognizes the model as his facility/his problem, then he maybe can be convinced to sign a contract. A presentable model should be a colored and animated 3D scenarios of the system modeled with a formal methodology under the surface.

There exist universal visualization tools, e.g. PEP [16], which associate to Petri net places and transitions an arbitrary 3D graphical object. Also the moves of transitions in the model can be associated with moves of the 3D objects.

We also tried to use visualization engines from commercial games. (e.g. Petri nets with Half Life 2 [17]).

We also tried to use it for visualization of ad hoc wireless networks, not only for visualization but also simulation, because in principle the radio propagation model like light, features like multiple way propagation, shade, obstacles etc. should be suited better than those primitive models used in special purpose simulators such as ns2 [18].

7.2 Performance

As the survey [5] showed efficiency of the analysis algorithms is a problem when applied to problems of realistic size. Often state space based methods have been developed for toy size examples, when applied to real systems these methods fail.

7.3 Hierarchical Modeling

For understanding, design and analysis of every complex system a hierarchical approach is necessary. This valid not only for manual design of the model, but also for generation of the model out of other data (such as XML based production plan or similar).

Hierarchy is a serious problem when regarding the successful application of Petri nets. There are several incompatible hierarchy concepts and not all of them are theoretically well founded. Often there exits no analysis methods that exploit the hierarchic structure so that unfolding and using standard methods is the only possibility, of course only viable for less complex systems.

8 Conclusion

In this work we discussed the applicability of mainly Petri net based formal modeling, simulation and analysis of systems in practice.

We pointed out which are the crucial aspects that often decide about the success of the transfer of scientific concepts to practical application. We also showed on several examples, why application of theoretic concepts may fail, or why theoretic concepts might not fit the needs of practical application/convince decision makers to use it.

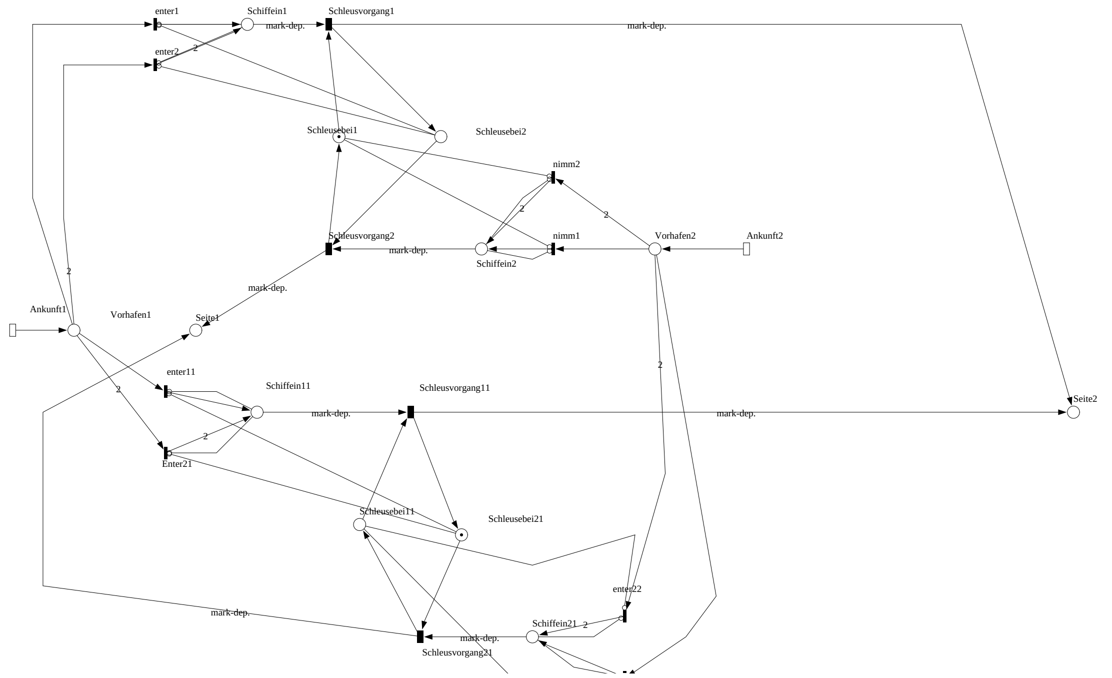


Fig. 3 Model of the shiplift with uncolored GSPN

It turned out that algorithms for complex models and tool support for such models and algorithms is essential. Of course good usability is necessary for practitioners to be able to use the tool.

Often Petri nets based approaches in practice require a high degree of customization by researchers. This is not acceptable for easy use for practitioners for several reasons: Customization of a methodology might violate the preconditions for some analysis methods and furthermore needs time.

There is not easy solution to the problems mentioned here or maybe no solution at all. However in times of lower and lower budgets for research, the topic of making valuable scientific methods more attractive for application in reality should be discussed.

9 References

[1] K. Pawlikowski, H.-D.J. Jeong, and J.-S. Ruth Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, pages 132–139, January 2002.

[2] K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Proactical Use.*, volume Vol. 1 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1992.

[3] George A. Bekey and Boris Kogan. *Modeling and Simulation. Theory and Practice. A Memorial Volume for Professor Walter J. Karplus*. Kluwer, 2003.

[4] Harald Störrle. Erfahrungen aus einer Vergleichsstudie über Petrinetz Werkzeuge. *Proceedings of the AWPN*, 1998.

[5] H. Störrle. An evaluation of high-end tools for Petri nets. *Tech.Rep.Nr.9802, LMU München*, 1998.

[6] `http://www.eas.asu.edu/ masmlab`.

[7] Matthias Becker and Helena Szczerbicka. PNiQ: Integration of queuing networks in generalized stochastic Petri nets. *special issue of IEE proceedings - Software*, 146(1):27–33, February 1999.

[8] `http://www.informatik.hu-berlin.de/top/pnml/`.

[9] `http://www.renew.de`.

[10] M. Becker, Bessey T., and H. Szczerbicka. A study of control via on-line simulation using stochastic petri nets. *In Proceedings of European Simulation Symposium (ESS)*, pages 405–413., 2003.

[11] M. Becker. Modeling and simulation of a complete semiconductor manufacturing facility in petri nets. *In Proceedings of IEEE Conference on Emerging Technologies and Factory Automation, ETFA*.

[12] M. Becker and Bessey T. Planning the reconstruction of a shiplift by simulation of a stochastic petri net model. *In Proceedings of European Simulation Symposium (ESS)*, pages 558–561., 2003.

[13] `http://www.daimi.au.dk/CPnets/intro/example_indu.html`.

[14] Matthias Becker, Lutz Twele, and Helena Szczerbicka. Integrating software performance evaluation in software-engineering. In *First international conference on grand challenges for modeling and simulations at Western MultiConference 2002*, San Antonio, Texas, USA, 2002. SCS.

[15] W.M.P. van der Aalst. Three Good reasons for Using a Petri-net-based Workflow Management System. In S. Navathe and T. Wakayama, editors, *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, pages 179–201, Camebridge, Massachusetts, 1996.

[16] Bernd Grahlmann. The PEP tool. In Orna Grumberg, editor, *Computer Aided Verification*, volume 1254 of *Lecture Notes in Computer Science*, pages 440–443. Springer, 1997.

[17] M. Becker. Generating interactive 3-D models for discrete-event modeling formalisms. *accepted for CyberWorlds 2007, Hannover, Germany.*, 2007.

[18] `http://www.isi.edu/nsnam/ns/`.