PENETO: A PETRI NET SIMULATOR FOR FAST SAFETY AND QUALITY ANALYSIS AND COST PREDICTION

Stefan Heller, Stefan Greiner
[1] DaimlerChrysler Research,
70546 Stuttgart, Germany
Stefan.Heller@DaimlerChrysler.com

Graham Horton
Computer Science Department,
University of Magdeburg,
39106 Magdeburg, Germany

KEYWORDS

Stochastic Petri-Nets, Modelling tools, Safety and Quality Analysis.

ABSTRACT

PeNeTo is Petri-net tool for modeling, visualizing and analysing extended generalized stochastic Petri nets, which is being developed at DaimlerChrysler Research. It contains powerful modeling and parameterization functionalities, analysis by both discrete-event simulation and numerical techniques, and a graphical visualization of the token game. It is used with discrete-event simulation for various system studies at DaimlerChrysler; numerical analysis via the method of supplementary variables and phase-type distributions is currently under development.

The goal of this paper is to present PeNeTo and some of the areas in which it is used. We first describe the environment in which PeNeTo is being developed and applied. From this we derive requirements for the tool's functionality, which are then described. Part of this functionality is demonstrated using an example application. In the final section, the current state of PeNeTo is described, together with plans for its further development.

GOALS AND APPLICATIONS OF PENETO

Modern road vehicles contain increasingly complex and interconnected systems, with ever-increasing levels of connectivity and interaction. This immense growth in complexity makes it more and more difficult to make accurate predictions about such systems with a reasonable amount of effort and time. For this reason, research and development in the automotive industry needs easy-to-use and powerful analysis tools. One example of this is to build a formal model of a system which can be easily analyzed by computer; PeNeTo is one such tool for the modeling and analysis of stochastic Petri nets.

PeNeTo is designed for a broad range of applications, including safety, quality and cost analysis. Safety analysis is concerned with the design of safe automotive systems, including their interactions with humans. Safety analysis makes use of reliability, availability and safety information such as load and throughput of system components and probabilities for critical states. Quality analysis is concerned with building models at the component, system and vehicle levels for obtaining information on product quality. One example of this is comparing the behavior of systems built using different components. Cost analysis allows, for example, the calculation of price models for maintenance and service packages which depend on the level of service offered, as well as the age and mileage of the vehicle. In addition, predictions can be made for costs incurred to the manufacturer during the vehicle's warranty period which are caused by quality problems. In all cases, the goal is that these analyses can be performed as fast and as comfortably as possible, while covering a large number of variants, in order to study and compare different real-life scenarios.

PeNeTo is designed with a broad functionality; it contains a large number of the various extensions to stochastic Petri nets that have been proposed, including immediate transitions, guard functions, user-defined rewards, place capacities, arc multiplicities, marking-dependent firing rates, single-server and infinite-server transitions, and age and enabling memory policies.

TOOL FUNCTIONALITY

PeNeTo consists of a graphical editor, a simulator and a visualization component. It is implemented in Java, in order to ensure platform-independence, and also to make use of its special language features. One example is the dynamic loading of classes, which makes it possible to include functions in the Petri net model which are formulated in Java syntax, which can then be compiled and executed at run-time. This means that such functions execute with the same level of performance as the tool itself, since expensive parsing routines are no longer necessary. Furthermore, the Java compiler and interpreter are available everywhere free of charge.

The PeNeTo graphical editor is designed with multi-document capability, allowing several nets to be edited simultaneously. This in turn allows sub-nets to be copied between models and for these to be compared quickly. Since Petri nets can quickly become very complex and unwieldy, PeNeTo also allows hierarchical modeling. A net may be divided into sub-nets, which may be edited separately and linked together via transition or place interfaces. This supports both logical model development on the one hand, and clear graphical presentation of large models on the other. PeNeTo contains a large number of configurable parameters; model parameterization is supported by dialog windows for each net component, which include plausibility checks for parameter values. All net components, with the exception of arcs, have unique names.

The visualization component of PeNeTo can be started directly from the editor. This allows the user to study the behavior of the net by observing the token game. This is important both for demonstration as well as debugging purposes. The visualization module contains continuous speed settings as well as a step-by-step mode. The enabling state and enabling times of timed transitions are visualized in order to further enhance understanding of the net's dynamic behavior.

The simulator may also be started directly from the editor. Both simulation up to a specified point in time and up to an absorbing state are permissible. Furthermore, a number of replications can be specified, and the simulator will provide appropriate statistical results. These include the values of the user-specified rewards in addition the standard measures for places (probability of being non-empty, average number of tokens) and transitions (throughput and probability of being enabled.)

CLASS OF NETS SUPPORTED

PeNeTo supports a very general class of stochastic Petri nets. These include many additional features which greatly enhance the tool's usefulness to the modeler.

Places can be assigned an initial marking and a maximum capacity, which may either have a constant value or be defined as a function of the current state of the net. An otherwise-enabled transition is disabled, if, by firing, the maximum place capacity would be exceeded.

Transitions have the largest numbers of parameters. Both timed and immediate transitions are supported. Immediate transitions can be assigned a probability or weight, whereas timed transitions can be assigned a firing time distribution chosen from a large set of alternatives. Firing time distributions range from exponential and phase-type distributions to Weibull and Normal distributions. In addition, the parameters for the firing probabilities and distributions may be defined as marking-dependent functions. PeNeTo also allows the use of marking-dependent guard functions and priorities to control the enabling of transitions. Transitions may be of single, multi- or infinite server type; multi-server and infinite-server transitions are treated by the simulator by assigning multiple firing times to the transition in accordance with the current enabling degree. Transitions may also be assigned a memory policy of type age or enabling. Memory policies define how the enabling time of a transition is treated when the transition becomes disabled for any reason other than itself firing. In the enabling case, a transition will "forget" that it has been enabled for a certain period of time; when it once again becomes enabled, a new firing time will be computed. In the case of the age policy, the transition "remembers" its enabling time, which shortens it remaining firing time when it once again becomes enabled. These memory policies are very important for modeling purposes.

PeNeTo also supports multiple arcs, whose multiplicity may be specified as a constant or with a function.

Various types of rewards are also available, including both accumulated and non-accumulated rewards, including impulse rewards associated with the firing of a transition. In addition to the net components themselves, independent parameters and functions may be defined, which can be referenced by the functions used to parameterize the net components. These facilitate the fast and safe modification of model parameters.

EXAMPLES

Figure 1 shows a small Petri net in the PeNeTo editor window, which is part of a model developed at DaimlerChrysler Research. The goal of this project was to study the reliability of a system over the entire life cycle of a vehicle, including the interaction with the driver, in particular the manual deactivation of faulty systems. Rewards are used to compute the probabilities of light and serious damage occurring due to inappropriate behavior by the driver.

Figure 2 shows a screenshot taken during the visualization of the simulation of a small example net. The distribution functions of the transitions can be identified by the transition color. The expired enabling time of each transition is shown by the darker portion of the transition. Tokens are drawn in red. When a transition fires, tokens move symbolically along the input arcs from the input places to the transition and along the output arcs from the transition to the output places, according the various arc multiplicities. Viewing the simulation visualization in this manner significantly enhances the user's understanding of the net's dynamic behavior, and is very useful in presentations to non-specialists and of course for verifying (i.e debugging) the model.

OUTLOOK

PeNeTo is in active use at DaimlerChrysler Research in the form described here. In addition to the current functionality, various extensions are also planned. These include features for automatic report generation and integration with a corporate database containing up-to-the-minute reliability data. This will allow cost and quality analyses to be made using the tool which are based on the latest available information on the reliability of vehicle components.

Many questions of interest can be answered by analyzing the state space of a net. These include qualitative properties such deadlocks and livelocks in addition to the usual quantitative simulation results such as average reward values and marking probabilities. In particular, when all timed transitions in the net are exponentially distributed (or phase-type), then the state space can be converted into a continuous-time Markov chain, for which very efficient numerical transient and steady-state analysis methods are available. For these reasons, state space generation and analysis techniques will also be included in the tool.

A further issue of interest is to increase the performance of the discrete-event simulation itself. This is motivated by the high accuracy requirements for safety analyses, requiring a large number of replications to achieve statistically significant results, and the large degree of stiffness of many models, which require a long time to reach steady-state. In both cases, very expensive computations can result. Our approach to accelerating simulations by aggregation of the state space [1] will be studied in the Petri net context. In addition, the automatic parallelization of replications across a network of workstations will be implemented.

REFERENCES

1. Heller, S., Horton, G. and Luber, M., "Accelerating Discrete-Event Simulation via State Space Reduction", ESM 2001, Prague, June 2001, Society for Computer Simulation and Modeling.

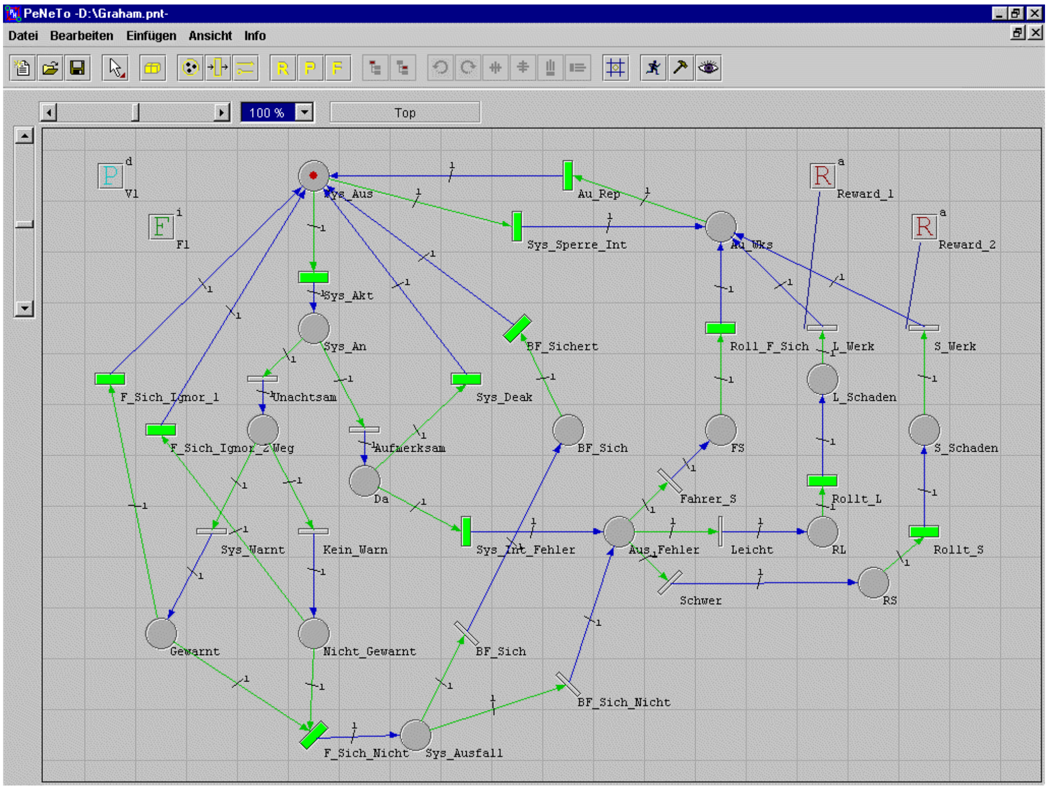

Figure 1: Excerpt from a PeNeTo Model

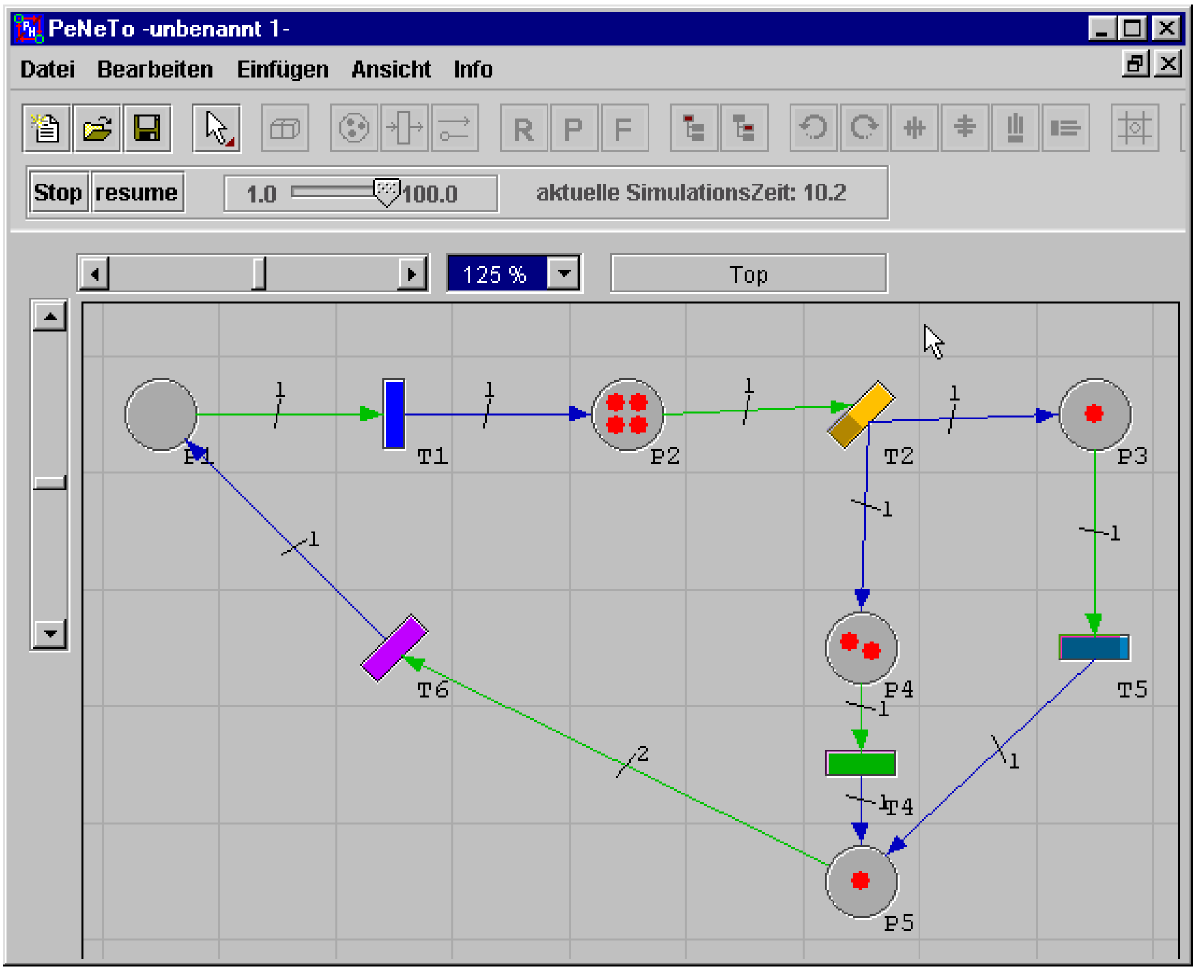

Figure 2: Screenshot of Simulation Visualization