Towards Version 4.0 of TimeNET

Armin Zimmermann, Michael Knoke,
Alexander Huck, and Günter Hommel

Technische Universität Berlin
Institut für Technische Informatik und Mikroelektronik
Einsteinufer 17, 10587 Berlin, Germany
{azi, mknoke, ahuck, hommel}@cs.tu-berlin.de

Abstract. This paper presents recent changes to TimeNET, a software tool for the modeling and performability evaluation using stochastic Petri nets. The tool is designed to deal with models including non-exponentially distributed firing delays. Colored stochastic Petri nets are added, and the graphical user interface has been reimplemented to let the tool run under Unix, Linux and Windows.

1 Introduction

Modeling and evaluation of non-trivial technical systems are only possible in practice with the support of appropriate software tools. Numerous commercial, research and prototype software tools have been developed for stochastic discrete event systems in the past; an overview is e.g. given in [1]. This paper describes recent progress of TimeNET, a software tool for the modeling and performability evaluation using stochastic Petri nets. Its main characteristic is the evaluation of models with non-exponentially distributed firing delays. TimeNET has been successfully applied during several modeling and performance evaluation projects and there are several hundred installations in universities and other organizations worldwide.

Previous publications describing the tool TimeNET include [2–4], and numerous applications have been considered. TimeNET is in a transition phase at the time of this writing. Version 3.0 has been stable for some time now, while development of the next major revision takes place. TimeNET 4.0 includes a new graphical user interface written in JAVA to allow platform-independent use of the tool. Another significant improvement is the availability of colored stochastic Petri nets.

TimeNET runs under Solaris 5.9, Debian Linux 3, and the Windows environment. The tool is available free of charge for non-commercial use from its home page at `http://pdv.cs.tu-berlin.de/~timenet`. Version 4.0 is planned to be available to the public in early 2006.

2 Supported Net Classes and Analysis Methods

The classical main model class of TimeNET are *extended deterministic and stochastic Petri nets* (eDSPNs). Firing delays of transitions can either be zero (immediate), exponentially distributed, deterministic, or belong to a class of general distributions called *expolynomial* in an eDSPN. Such a distribution function can be piecewise defined by exponential polynomials and has finite support. It can even contain jumps, making it possible to mix discrete and continuous components. Many known distributions (uniform, triangular, truncated exponential, finite discrete) belong to this class.

Under the restriction that all transitions with non-exponentially distributed firing times are mutually exclusive, stationary numerical analysis is possible. If the non-exponentially timed transitions are restricted to have deterministic firing times, transient numerical analysis is also provided. For the case of concurrently enabled deterministically timed transitions, an approximation component based on a generalized phase type distribution has been implemented. If there are only immediate and exponentially timed transitions, the model is a GSPN and standard algorithms for steady-state and transient numerical evaluation based on an isomorphic Markov chain are applicable.

The tool also comprises a simulation component for eDSPN models, which is not subject to the restriction of only one enabled non-Markovian transition per marking. Steady-state and transient simulation algorithms are available. Results can be obtained faster by parallel replications, using control variates, or with the RESTART method in the presence of rare events. During the simulation run, intermediate results of the performance measures are displayed graphically together with the confidence intervals.

Simple stochastic Petri nets in TimeNET can either be interpreted in continuous time as an eDSPN, or as a discrete deterministic and stochastic Petri net (DDSPN). DDSPNs allow geometric distributions, deterministic times and discrete phase type distributions as delays. Steady-state and transient numerical analysis as well as efficient parallel simulation are available.

Variable-free colored Petri nets (VfCPNs) represent another model class of TimeNET. They were initially developed for the modeling and evaluation of manufacturing systems [5], but are not restricted to this area as shown in alter applications. Firing delays of transitions have the same range as in eDSPNs. Numerical steady-state analysis, an iterative approximation method, and standard simulation are available for VfCPNs. A modeled (production) system can be directly controlled with the tool.

Stochastic colored Petri nets (SCPNs) have been added recently during the ongoing transition to TimeNET version 4.0. Due to the inherent complexity of the considered models, a requirement of only one non-exponential transition per marking is too restrictive. Thus a standard discrete-event simulation has been implemented for the performance evaluation of SCPN models. A distributed simulation method has been developed and implemented for this model class [6] in addition, which allows the efficient simulation of complex models on a cluster of workstations.

Both simulation programs automatically generate a set of C++ code fragments from an SCPN model, which implement the core functions of transitions and other model elements. The results are compiled and linked with a run-time kernel of the simulation. A model-specific and very efficient simulation program is thus obtained, which is then used either for a fast simulation in background or the token game in a single-step mode that cooperates with the GUI. Performance measures can be analyzed during run time with a graphical front end.

The *token game*, i.e. an interactive simulation of the behavior of a model, is available for all mentioned model classes in TimeNET. There are other parts of TimeNET which are not covered here for lack of space.

3 A Model-Class Generic Graphical User Interface

The graphical user interface for TimeNET 4.0 has been completely rewritten in JAVA, and can therefore be run on both Unix- and Windows-based environments. The new GUI retains the advantages of the former one, especially in being generic in the sense that any graph-like modeling formalism can be easily integrated without much programming effort. Nodes can be hierarchically refined by corresponding submodels. The GUI is thus not restricted to Petri nets, and is already being used for other tools than TimeNET. As a stand-alone program it is named PENG, which is short for *platform-independent editor for net graphs.*

Model classes are described in an XML schema file, which defines the elements of the model. Node objects, connectors and miscellaneous others are possible elements. For each node and arc type of the model the corresponding attributes and the graphical appearance is specified. The shape of each node and arc is defined using a set of primitives, and may depend on attribute values of the

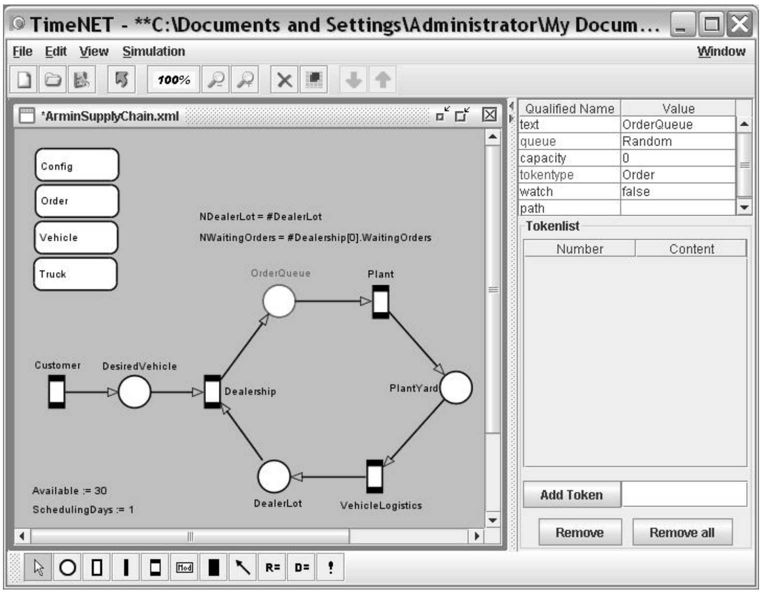

Fig. 1. Sample screen shot of the graphical user interface

object. Actual models are stored in an XML file consistent with the model class definition.

Program modules can be added to the tool to implement model-specific algorithms. A module can select its applicable net classes and extend the menu structure by adding new algorithms. All currently available and future extensions of net classes and their corresponding analysis algorithms are thus integrated with the same "look-and-feel" for the user.

Figure 1 shows a sample screen shot of the GUI. There are standard menus with the necessary editing commands in the top row. Commands should be self-explanatory and follow usual GUI-style. A set of icons below the menu bar allows access to menu commands which are frequently used.

The main window contains the editing area. The current model is edited in it just like in a standard drawing tool with operations for selecting, moving, and others. The lower icon bar shows all objects that are available in the model class. The contents of this bar are automatically derived from the model class description of the currently opened model. Individual attributes of an element are edited by selecting it in the drawing area and changing the values in the right tab. There is one entry in this window for every object attribute as defined in the model class for that object.

Acknowledgments

The authors would like to acknowledge the numerous colleagues and students who have contributed to the development and implementation of TimeNET, as well as the organizations that supported the development through grants and projects.

References

1. Haverkort, B.R., Niemegeers, I.G.: Performability modelling tools and techniques. Performance Evaluation **25** (1996) 17–40
2. Zimmermann, A., Freiheit, J., German, R., Hommel, G.: Petri net modelling and performability evaluation with TimeNET 3.0. 11th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation. Springer LNCS 1786 (2000), 188–202
3. German, R.: Performance Analysis of Communication Systems, Modeling with Non-Markovian Stochastic Petri Nets. John Wiley and Sons (2000)
4. German, R., Kelling, C., Zimmermann, A., Hommel, G.: TimeNET – a toolkit for evaluating non-Markovian stochastic Petri nets. Performance Evaluation **24** (1995) 69–87
5. Zimmermann, A., Freiheit, J., Huck, A.: A Petri net based design engine for manufacturing systems. Int. Journal of Production Research, **39** (2001) 225–253
6. Knoke, M., Kühling, F., Zimmermann, A., Hommel, G.: Towards correct distributed simulation of high-level Petri nets with fine-grained partitioning. 2nd Int. Symp. on Parallel and Distributed Processing and Applications (ISPA'04). Springer LNCS 3358 (2004), 64–74