

Using the Probabilistic Evaluation Tool for the Analytical Solution of Large Markov Models

Boudewijn R. Haverkort¹
and
Aad P.A. van Moorsel²

Introduction. Stochastic Petri net-based Markov modeling is a potentially very powerful and generic approach for evaluating the performance and dependability of many different systems, such as computer systems, communication networks, manufacturing systems, etc. As a consequence of their general applicability, SPN-based Markov models form the basic solution approach for several software packages that have been developed for the analytic solution of performance and dependability models, e.g., [1, 8]. In these tools, stochastic Petri nets are used to conveniently specify complicated models, after which an automatic mapping can be carried out to an underlying Markov reward model. Subsequently, this Markov reward model is solved by specialized solution algorithms, appropriately selected for the measure of interest.

State-space explosion. One of the major aspects that hampers the use of SPN-based Markov models for the analytic solution of performance and dependability results is the size of the state space. Although typically models of up to a few hundred thousand states can conveniently be solved on modern-day work-stations, often even larger models are required to represent all the desired detail of the system. Our tool PET (*probabilistic evaluation tool*) circumvents problems of large state spaces when the desired performance and dependability measure are transient measures. It does so by an approach named *probabilistic evaluation*.

Probabilistic evaluation. The approach followed in PET has been motivated by what is known in the area of communication protocol design as “probabilistic verification” [5] or “probabilistic validation” [2]. In these approaches, one verifies or validates the correctness of a protocol specification by only addressing part of its underlying state machine. This, of course, then leads to probabilistic statements on the correctness of these protocols, preferably in the form of a lower

bound, i.e., one obtains statements such as “with 99% certainty this protocol specification does not allow for non-admissible states”.

In PET, the ideas behind probabilistic verification and validation are used to compute transient measures in Markov reward models. Important to note is the fact that PET uses solution algorithms which do not rely on *a priori* generation of the overall state space. To achieve this goal, newly developed solution techniques for transient measures have been implemented in PET. These algorithms are based on uniformization, a method considered by different authors to be the method of choice for transient analysis of Markovian models [3, 6].

For acyclic models, orthogonal uniformization (OU) has been developed [6, 7]. In OU the memory occupancy is no longer determined by the size of the state space. This makes it possible to solve very large, and even infinitely large, acyclic Markov models. To deal with non-acyclic models a different method called partial uniformization (PU) can be used [7]. This method is particularly promising for handling large state spaces for general models. It does not generate the whole state space *a priori*, but step-wise traverses the state space, using the fact that for transient measures bounds on the induced error can be straightforwardly obtained along the way.

State-space generation ordering. The power of both OU and PU as a probabilistic evaluation approach is influenced considerably by the order in which the state space is traversed. Preferably, the state space is generated in such order that as few states as possible are used in the evaluation. Different state space exploration approaches are implemented in PET: a standard breath-first generation method, a “mean-time-to-unknown”-method [9] and a simple heuristic [7]. In a comparison of these, it has turned out that the heuristic approach explores the state space in a close to optimal order without much overhead. Other heuristics are presented in [4].

PET: probabilistic evaluation tool. The practical value of the combination of uniformization and state space exploration as a probabilistic evaluation technique, depends on the actual implementation and application. It is therefore essential to experiment with prototype implementations of the algorithms, for realistic systems. Therefore, we have developed PET, and used it to apply the proposed methods to practical models.

Key idea within PET is to generate only that part of the underlying Markov chain that is really needed to derive bounds on the performance/dependability mea-

¹University of Twente, Department of Computer Science, P.O. Box 217, 7500 AE Enschede, the Netherlands (b.r.h.m.haverkort@cs.utwente.nl)

²University of Illinois, Center for Reliability and High-Performance Computing, 1308 W. Main St., Urbana, IL 61801, USA (moorsel@crhc.uiuc.edu)

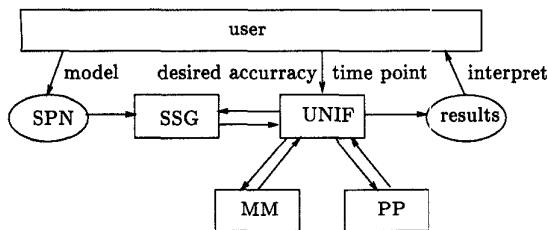


Figure 1: Block-diagram overview of PET

sure of interest. What takes place in PET is a form of “uniformization-on-the-fly,” i.e., uniformization is applied *while* generating the state space. The state space generation is stopped as soon as enough probability mass has been collected to probabilistically evaluate the model.

A block diagram of the tool is given in Figure 1. From this figure, it becomes clear that the tool user has to input the following: (i) a (dependability) model as an SPN; (ii) the desired time point of evaluation t ; and (iii) the desired accuracy.

Depending on the particular uniformization technique to be used, the uniformization tool (UNIF) starts requesting the state space generator (SSG) for states, i.e., for lines of the (partial) generator matrix describing the underlying Markov chain. Memory is reserved dynamically by the memory management unit (MM). Weighting factors in the uniformization summation are provided by the Poisson probabilities calculator (PP); they are calculated only once. If the user requests so, or if needed to fulfill the accuracy requirements, the uniformization tool asks for more states to be generated.

PET currently supports the following uniformization techniques: standard uniformization, dynamic uniformization, orthogonal uniformization, and partial uniformization [6, 7]. PET also includes the three state space generation techniques presented above.

Implementation issues. PET has been implemented in C for Sun Sparc 1 workstations and covers around 4000 lines of commented code. The SPN models that serve as input are described using the C-based SPN language of SPNP, i.e., CSPL [1].

The state space generator is currently only *virtually* present. What actually has been implemented is that the overall state space is generated using the SPNP package [1], however, access to the state space and the generator matrix is done in such a way that, from the viewpoint of the uniformization tool, it is as if the state space is generated stepwise and on-demand.

This feature of PET makes that new state space generation techniques can easily be incorporated. Once fully tested, they can be integrated in an SPN tool that really transforms the SPN to the Markov reward model.

The current version of PET has been used in [4, 7] to study uniformization and state space exploration methods. We refer to these papers for a more detailed discussion and evaluation of the techniques involved in probabilistic evaluation.

References

- [1] G. Ciardo, J. Muppala, K.S. Trivedi, “SPNP: Stochastic Petri Net Package”, *Proceedings of the Third International Workshop on Petri Nets and Performance Models*, IEEE Computer Society Press, pp.142–151, 1989.
- [2] G. Florin, C. Fraize, S. Natkin, “A New Approach of Formal Proof: Probabilistic Validation”, *Dependable Computing and Fault-Tolerant Systems*, 6, J.F. Meyer, R.D. Schlichting (Eds.), pp.357–374, Springer-Verlag, Wien, 1992.
- [3] W.K. Grassmann, “Finding Transient Solutions in Markovian Event Systems through Randomization”, *Numerical Solution of Markov Chains*, W.J. Stewart (Ed.), Marcel Dekker, New York, 1991.
- [4] B.R. Haverkort, “In Search for Probability Mass: Probabilistic Evaluation of High-Level Specified Markovian Models”, *The Computer Journal* (British Computer Society), forthcoming, 1996.
- [5] N.F. Maxemchuk, K. Sabnani, “Probabilistic Verification of Communication Protocols”, *Protocol Specification, Testing and Verification*, VII, H. Rudin, C.H. West (Eds.), North-Holland, pp.307–320, 1989.
- [6] A.P.A. van Moorsel, *Performability Evaluation Concepts and Techniques*, Ph.D. thesis, University of Twente, 1993.
- [7] A.P.A. van Moorsel, B.R. Haverkort, “Probabilistic Evaluation for the Analytical Solution of Large Markov Models: Algorithms and Tool Support”, accepted for publication in *Microelectronics and Reliability*, 1995.
- [8] W.H. Sanders, W.D. Obal II, M.A. Qureshi, F.K. Widjanarko, “The *UltraSAN* Modeling Environment”, forthcoming in *Performance Evaluation, special issue on Performance Modeling Tools*, 1995.
- [9] E. de Souza e Silva, P.M. Ochoa, “State Space Exploration in Markov Models”, *ACM Performance Evaluation Review*, 20, pp.152–166, 1992.