In Proc. 29th Int. Symposium on Fault-Tolerant Computing Systems (FTCS), Madison, Wisconsin, pp. 228-231, June 1999

The DSPNexpress 2.000 Performance and Dependability Modeling Environment

Christoph Lindemann, Andreas Reuys, and Axel Thümmler
University of Dortmund
Department of Computer Science
44221 Dortmund, Germany
{cl, reuys, thummler}@cs.uni-dortmund.de
http://www4.cs.uni-dortmund.de/~Lindemann/

Abstract

This paper describes the latest version of the software package DSPNexpress, a tool for modeling with deterministic and stochastic Petri nets (DSPNs). Novel innovative features of DSPNexpress 2.000 constitute an efficient numerical method for transient analysis of DSPNs with and without concurrent deterministic transitions. In particular, DSPNexpress 2.000 can perform transient analysis of DSPNs without concurrent deterministic transitions in three orders of magnitude less computational effort than the previously known method. Furthermore, DSPNexpress 2.000 contains an effective numerical method for steady-state analysis of DSPNs with concurrent deterministic transitions.

1. Innovative Features of DSPNexpress

To effectively employ model-based evaluation of computer and communication systems, software environments are needed that simplify model specification, modification, as well as automate quantitative analysis. Due to the complexity of practical modeling applications requiring sophisticated solution methods, the development of effective software tool support for stochastic Petri nets is an active research area. Software packages for stochastic Petri nets include GreatSPN [4], QPN-tool [2], SPNP [6], SURF-2 [3], and UltraSAN [12].

This paper describes the latest version of one such software package, the DSPNexpress 2.000 modeling environment. The previous version of DSPNexpress, DSPNexpress1.5 is known for its highly efficient numerical method for steady state analysis of deterministic and stochastic Petri nets (DSPNs, [1]) without concurrent deterministic transitions [8], [9]. This numerical method analyzes complex DSPNs with four orders of magnitude less computational effort that the previously known method implemented in the version 1.4 of the package GreatSPN. Novel innovative features of the DSPNexpress 2.000 include:

(1) Efficient numerical method for transient analysis of DSPNs without concurrent deterministic transitions based on an iterative numerical solution of one-dimensional Volterra integral equations [10]. As shown in Section 4, this method can perform numerical transient analysis of complex DSPNs in some minutes CPU on a modern workstation.

(2) An implementation of an effective numerical method for transient and steady-state analysis of DSPNs with two deterministic transitions concurrently enabled. These tasks require numerical solution of two-dimensional Volterra equations by an iterative scheme and direct quadrature, respectively. On a modern workstation, transient analysis of quite complex DSPNs (i.e., with 10 thousand tangible markings with moderate stiffness in the parameter settings) requires about 40 minutes of CPU time [11], steady-state analysis less than 10 minutes of CPU time.

(3) Orthogonal software architecture especially tailored to numerical analysis of the stochastic process underlying a discrete-event stochastic system with exponential and deterministic events (i.e., a Markov regenerative process [5] or a generalized semi-Markov process [10]) based on interprocess communication with UNIX sockets rather than writing intermediate results in files.

(4) Plug-and-play interface such that numerical solvers can easily be utilized to quantitative evaluation of arbitrary discrete-event stochastic systems with exponential and deterministic events specified in other modeling formalisms than just DSPNs (e.g., hardware systems represented as finite state machines).

In previous work, transient analysis of DSPNs was always based on the restriction that deterministic transitions are not concurrently enabled. Choi, Kulkarni, and Trivedi observed that the marking process underlying a DSPN with this restriction is a Markov regenerative stochastic process [5]. They introduced a numerical method for transient analysis of such DSPNs based on numerical inversion of Laplace-Stieltjes transforms. More recently, German et al. developed a numerical method for transient analysis of DSPNs based on the approach of supplementary variables [7]. While these methods are certainly of theoretical interest, they are both not suitable for application in practical dependability modeling projects.

The remainder of this paper is organized as follows. Section 2 describes the software architecture of DSPNexpress 2.000. The graphical interface of the package is briefly recalled in Section 3. To illustrate the applicability of DSPNexpress 2.000 for practical dependability modeling projects, we present performance curves of the newly implemented transient solver in Section 4. Finally, concluding remarks are given.

2. The Software Architecture of the Numerical Solvers

The core of the package DSPNexpress constitutes the solution engine for discrete-event stochastic systems with exponential and deterministic events. The software architecture of this solution engine and its software modules are shown in Figure 1. The solution engine is drawn as the big white rectangular box. The six software modules are drawn as rectangles. These software modules are invoked from the solution engine as UNIX processes. Interprocess communication with sockets drawn as broken ellipses is employed for passing intermediate results from one module to the next.

Steady state analysis of DSPNs without concurrent deterministic transitions relies on analysis of an embedded Markov chain (EMC) underlying such DSPNs [1]. To efficiently derive the probability matrix of this EMC, the concept of a subordinated Markov chain (SMC) was introduced. Recall that a SMC associated with a state

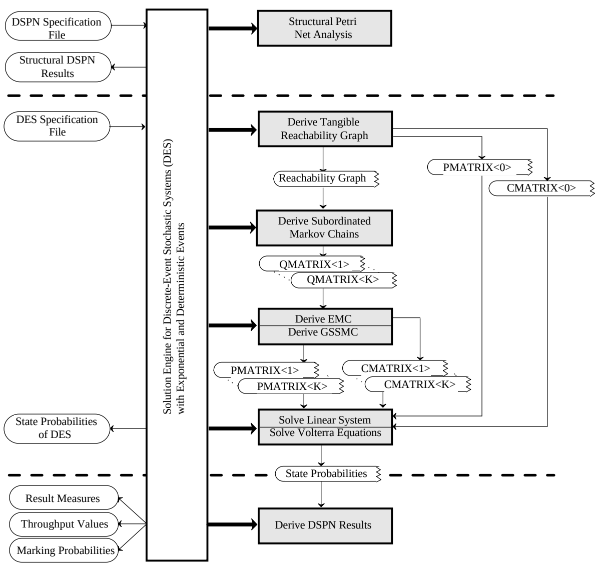

Figure 1. The solver architecture for DSPNs

s_i is a CTMC whose states are given by the transitive closure of all states reachable from s_i via the occurrence of exponential events [9]. After generating the reachability graph comprising of tangible markings (states) of the DSPN, for each state the generator matrix of its SMC is derived. These tasks are performed in the modules *Derive Tangible Reachability Graph* and *Derive Subordinated Markov Chains*, respectively. Entries of this probability matrix are computed by transient analysis of the SMCs. Subsequently, a linear system corresponding to the stationary equations of the EMC is solved. These task are performed in the submodules *Derive EMC* and *Solve Linear System*.

Transient analysis of DSPNs is based on the analysis of a general state space Markov chain (GSSMC) embedded at equidistant time points nD (n = 0,1,2,...) of the continuous-time marking process. The Chapman Kolmogorov equations of the GSSMC constitute a system of Volterra integral equations [11]. Steady state analysis of DSPNs with concurrent deterministic transitions relies on the same approach [10]. The transition kernel of the GSSMC specifies one-step jump probabilities from a given state at instant of time nD to all reachable new states at instant of time (n+1)D. Key drivers for the computational efficiency of the GSSMC approach constitute the separability and piece-wise continuity of the transition kernel [11]. Furthermore, the elements of the transition kernel can effectively be determined by an extension of the concept of subordinated Markov chains. Numerical computation of kernel elements relies also on transient analysis of these CTMCs. This task is performed in submodule *Derive GSSMC*. Subsequently, for transient analysis a number of iterations corresponding to the mission time are performed on the system of Volterra equations [11] whereas for steady state analysis a linear system is solved for each mesh point [9], [10]. This task is performed in the submodule *Solve Volterra Equations*.

We would like to point out that only the front end and the back end of the solution engine is tailored to DSPNs. That is instead of a DSPN specification file provided by the graphical interface of DSPNexpress, a specification file of an arbitrary discrete-event stochastic system with exponential and deterministic events (e.g., finite state machines) could be quantitatively evaluated by the solution engine of DSPNexpress using an appropriate filter.

3. The Graphical User Interface

Of course, the package DSPNexpress also provides a user-friendly graphical interface running under X11. To illustrate the features of this graphical interface, consider the snapshot shown in Figure 2. The first line displays the name of the package *DSPNexpress* and the actual version *2.000*, the affiliation of the authors, *University of Dortmund*, *Computer Systems and Performance*

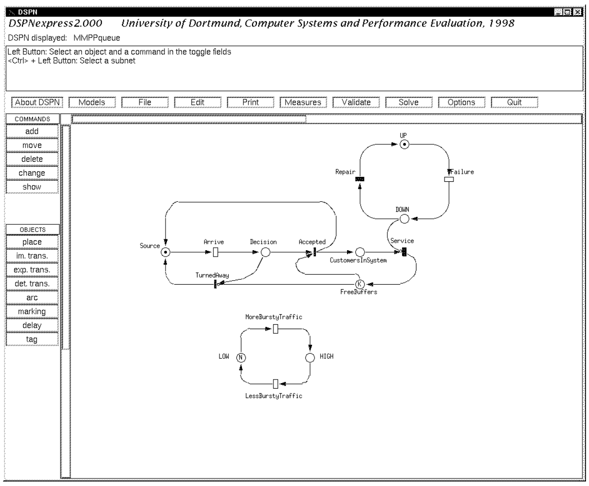

Figure 2. The graphical user interface

Evaluation Group, and the year of release *1998*. A DSPN of a single-server, finite- capacity queue with failure and repair is displayed. The model is named *MMPPqueue* because customers arrive according to a Markov modulated Poisson process. Recall that in DSPNs three types of transitions exist: immediate transitions drawn as thin bars fire without delay, exponential transitions drawn as empty bars fire after an exponentially distributed delay whereas deterministic transitions drawn as black bars fire after a constant delay.

At any time, DSPNexpress provides on-line helpmessages displayed in the third line of the interface. The command line and the object line are located on the left side of the interface. The buttons are located in a vertical line between the on-line help line and the working area. The working area constitutes the remaining big rectangle which contains the graphical representation of the DSPN *MMPPqueue*. This DSPN is displayed with the options *tags on*. Thus, each place and each transition of this DSPN is labeled (e.g., *Source*, *Arrive*, *Decision*, *Service*, etc.). A detailed description of the features of the graphical interface is given in [9].

4. Application Example

To illustrate the applicability of the transient solver of DSPNexpress 2.000 for practical dependability modeling projects, we consider a DSPN for an MMPP/D/1/K queue with breakdown and repair. The DSPN is shown in the working area of the graphical interface in Figure 2. The K tokens residing in place *FreeBuffers* in the initial marking represent the finite number of buffers of the single-server queueing system. The number tokens residing the place *LOW* control the mean firing time of the exponential transition *Arrive*. That is, the Markov modulated Poisson arrival stream is represented by defining the firing delay of the exponential transition *Arrive* dependent on the number of tokens in the place *LOW*. The number of tangible markings of this DSPN is given by

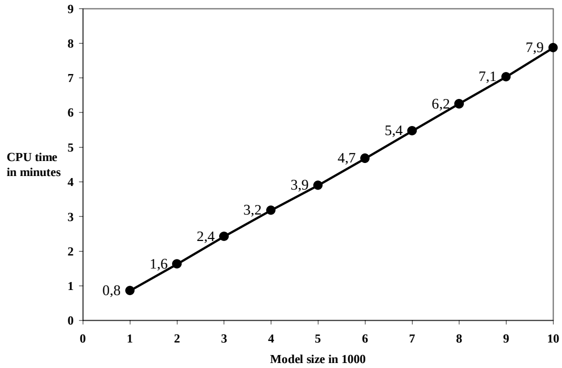

Figure 3. CPU time versus model size

$2 \cdot (K+1) \cdot (N+1)$. The constant service requirement is assumed as D = 1.0. We assume that after a failure the partly completed service is lost and is restarted after repair. In all experiments, model parameters of the Markov modulated arrival process are set such that the effective arrival rate λ_{eff} = 0.9. At time t = 0, zero customers reside in the queue and the system is *UP*. Failures of the system are assumed to be exponentially distributed. Repair times are assumed to be constant The experiments have been performed on a Sun Sparc Enterprise station with 1 GByte main memory running the operating system SunOS5.6. For the performance tests the CPU time has been measured by the UNIX system call *clock*. Figure 3 plots the CPU time required for computing the transient solution at instant of time t = 100 for increasing model size. We observe a linear growth of CPU time. This is due to the exploitation of the separability of the transition kernel of the GSSMC resulting in an almost linear growth of the nonzero kernel elements to be considered in the iterative scheme [11]. Figure 4 plots the memory requirements for storing the nonzero elements of the transition kernel versus model size and, thus, provides further evidence along this line. In these experiments, the number of discretization steps employed in the numerical quadrature is M = 10. A DSPN of an MMPP/D/1/K queue with failure and repair was already considered in [7] and a computational effort of 100 hours of CPU was reported. Figures 3 and 4 show

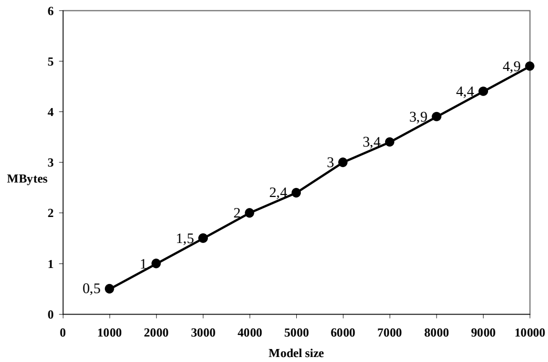

Figure 4. Memory requirements versus size

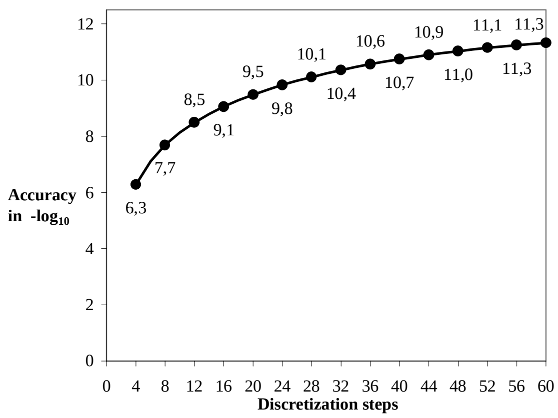

Figure 5. Numerical accuracy

that the software package DSPNexpress performs numerical transient analysis of such DSPNs three orders of magnitude faster than the previously known numerical method based on the approach of supplementary variables [7]. Since the DSPN does not contain concurrently enabled deterministic transitions, the stationary of time-averaged state probabilities of its marking process can be computed by an embedded Markov chain as already implemented in the previous version of the software package DSPNexpress [8]. Note that by setting the mission time sufficiently long, the transient solver can also be employed for computing stationary or time-averaged distributions. We use this fact for estimating the numerical accuracy achieved by the newly implemented transient solver for a given numerical quadrature of the Volterra integral equations. Figure 5 plots the accuracy of the stationary distribution of the DSPN achieved by the transient solver versus the number of discretization points employed in the iterative scheme. We observe that for already 10 discretization points a numerical accuracy of less that 10^{-7} is obtained.

5. Conclusions

This paper provided an overview of DSPNexpress 2.000, the new version of a widely distributed software package for modeling with deterministic and stochastic Petri nets. While the previous version of DSPNexpress was known for its highly efficient numerical solver for stationary analysis of DSPNs without concurrent deterministic transitions [8], DSPNexpress 2.000 also provides a method for transient analysis of DSPNs [11]. Furthermore, both the stationary analysis and the transient analysis is no longer restricted to the case that deterministic transitions cannot be concurrently enabled [10].

To illustrate the applicability of the newly implemented transient solver of DSPNexpress for practical dependability projects, we presented curves for an MMPP/D/1/K queue with failure and repair plotting the CPU time and memory requirements versus model size and mission time, respectively. For this DSPN, the transient solver of DSPNexpress implementation based on the GSSMC approach [9], [11] requires a couple of minutes of CPU time on a modern workstation whereas as reported in [7] the previously known method requires more than 100 hours of CPU time.

6. References

[1] M. Ajmone Marsan and G. Chiola, On Petri Nets with Deterministic and Exponentially Distributed Firing Times", in: *G. Rozenberg (Ed.) Advances in Petri Nets 1987, Lecture Notes in Computer Science 266*, 132-145, Springer 1987.

[2] F. Bause, P. Buchholz, and P. Kemper, QPN-tool for the Specification and Analysis of Hierarchically Combined Queueing Petri Nets, in: H. Beilner, F. Bause (Eds.) *Quantitative Evaluation of Computing and Communication Systems*, Lecture Notes in Computer Science, Vol. 977, pp. 224-238, Springer 1995.

[3] C. Beounes et al., SURF-2: A Program for Dependability Evaluation of Complex Hardware and Software Systems, *Proc. 23rd Int. Symp. on Fault-Tolerant Computing, Toulouse France*, 668-673, 1993.

[4] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaudo, GreatSPN 1.7: Graphical Editor and Analyzer for Timed and Stochastic Petri Nets, *Performance Evaluation*, **24**, 47-68, 1995.

[5] H. Choi, V.G. Kulkarni, and K.S. Trivedi, Transient Analysis of Deterministic and Stochastic Petri Nets, in: *M. Ajmone Marsan (Ed.) Application and Theory of Petri Nets 1993*, *Lecture Notes in Computer Science 691*, 166-185, Springer 1993.

[6] G. Ciardo, J. Muppala, and K.S. Trivedi, SPNP: Stochastic Petri Net Package, *Proc. 3rd Int. Workshop on Petri Nets and Performance Models Kyoto Japan*, 142-151, 1989.

[7] A. Heindl and R. German, A Fourth Order Algorithm with Automatic Step Size Control for the Transient Analysis of DSPNs, *Proc. 7th Int. Workshop on Petri Nets and Performance Models, Saint Malo, France*, 60-69, 1996.

[8] Ch. Lindemann, DSPNexpress: A Software Package for the Efficient Solution of Deterministic and Stochastic Petri Nets, *Performance Evaluation*, **22**, 3-21, 1995.

[9] Ch. Lindemann, *Performance Modelling with Deterministic and Stochastic Petri Nets*, John Wiley & Sons 1998.

[10] Ch. Lindemann and G.S. Shedler, Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation*, **27&28**, 565-582, 1996.

[11] Ch. Lindemann and A. Thümmler, Transient Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation*, (to appear).

[12] W.H. Sanders, W.D. Obal, M.A. Qureshi, and F.K. Widjanarko, The UltraSAN Modeling Environment, *Performance Evaluation*, **24**, 89-115, 1995.