The DSPNexpress 2000 Performance and Dependability Modeling Environment

Christoph Lindemann, Marco Lohmann, Axel Thümmler, and Oliver Waldhorst
University of Dortmund
Department of Computer Science
August-Schmidt-Str. 12
44227 Dortmund, Germany
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 2000 constitute efficient numerical methods for transient and steady-state analysis of DSPNs with concurrent deterministic transitions. In particular, DSPNexpress 2000 can perform transient analysis of DSPNs without concurrent deterministic transitions in three orders of magnitude less computational effort than the previously known method. To outreach from stochastic Petri net modeling to system specification languages used in industrial projects, DSPNexpress 2000 contains filters to the commercial design packages StateMate™ and Together™. Due to an open interface, the solvers of DSPNexpress can be utilized for analysis of discrete-event stochastic systems with exponential and deterministic events specified in arbitrary modeling formalisms.

Key words:

Software tools for system performance and dependability evaluation,
evaluation techniques for system performance and dependability,
transient and steady-state analysis of generalized semi-Markov processes,
numerical solvers for Fredholm integral equations.

1 Introduction

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], Möbius [18], QPN-tool [3], SPNP [6], and UltraSAN [17].

This paper describes the latest version of one such software package, the DSPNexpress 2000 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 [11], [13]. This numerical method analyzes 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 DSPNexpress 2000 constitute efficient numerical methods for transient and steady-state analysis of DSPNs with concurrent deterministic transitions. 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 [10]. Using the same approach, Telek and Horvath developed state equations for transient analysis of Markov regenerative stochastic Petri nets in which timed transitions keep their remaining firing times in case their firing process gets preempted for subsequent resumption instead of discarding them and restarting the firing process [19]. While these methods are certainly of theoretical interest, they are both not suitable for application in practical performance and dependability modeling projects. Numerical inversion of Laplace-Stieltjes transforms can only be employed for the analysis of simple models (i.e., DSPNs with a few tangible markings). The practical applicability of the supplementary variables approach is severely limited because it requires, already in the restricted case of no concurrent deterministic transitions, numerical solution of partial differential equations.

In previous work, we introduced an effective numerical method for transient and steady-state analysis of deterministic and stochastic Petri nets (DSPNs) with concurrent deterministic transitions [14], [15], [16]. This approach is based on the analysis of a general state space Markov chain (GSSMC) whose Chapman-Kolmogorov equations constitute a system of

multidimensional Fredholm integral equations. 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. In general, a transition kernel is a functional matrix. Key contributions of the GSSMC approach constituted the observations that most of the elements of the transition kernel of the GSSMC are constants (99% for DSPNs corresponding to queueing systems as shown in Section 4) and that the remaining elements comprise of piece-wise continuous functions. Such functional kernel elements are always separable. That is, elements depending on several clock readings can be expressed as the sum and/or product of constants, and functions depending on just one functional expression. It is known that a system of Fredholm integral equations with separable kernel is of a particularly simple form [8]. Therefore, its numerical solution requires orders of magnitude less computational cost than numerical solution of partial differential equations. As shown in Section 4, transient analysis of quite complex DSPNs (i.e., with 20 thousand tangible markings for mission time T = 100) requires about 26 minutes of CPU time [15], steady-state analysis less than 5 minutes of CPU time. Furthermore, we showed in [15] that DSPNexpress 2000 performs transient analysis of DSPNs without concurrent deterministic transitions in a few minutes of CPU time (i.e., three orders of magnitude less computational effort than the previously known method [10]).

To outreach from stochastic Petri net modeling to system specification languages used in industrial projects, DSPNexpress 2000 contains filters to the widely known commercial design packages StateMate™ and Together™. Thus, the numerical solvers of DSPNexpress can also be utilized for quantitative evaluation of system specifications with Harel state charts [9] as well as state charts and activity diagrams of the Unified Modeling Language (UML [7]). Due to an open interface, the solvers of DSPNexpress can be utilized for analysis of discrete-event stochastic systems with exponential and deterministic events specified in arbitrary modeling formalisms.

The remainder of this paper is organized as follows. Novel innovative features of the DSPNexpress 2000 are described in Section 2. In Section 3, we recall the GSSMC approach discuss the software architecture of the numerical solvers, and give a brief glance at the graphical user interface of the package. To illustrate the practical applicability of DSPNexpress 2000 in complex performance and dependability modeling projects, Section 4 presents curves plotting the CPU time and memory requirements of two DSPNs versus model size. These curves illustrate that DSPNexpress 2000 can effectively be employed for the transient and stationary analysis of DSPNs with large state space and two deterministic transitions concurrently active. Finally, concluding remarks are given.

2 Innovative Features of DSPNexpress 2000

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 [11], [13]. Furthermore, DSPNexpress1.5 contained already a graphical user interface running under X11 allowing easy model specification, modification, graphical animation, as well as automate quantitative analysis. Novel innovative features of the DSPNexpress 2000 include:

(1) An implementation of an efficient numerical method for transient analysis of DSPNs without concurrent deterministic transitions based on an iterative numerical solution of one-dimensional Fredholm integral equations [15].

(2) An implementation of an effective numerical method for transient and steady-state analysis of DSPNs with two deterministic transitions concurrently enabled [14], [15]. These tasks require numerical solution of two-dimensional Fredholm equations by an iterative scheme and direct quadrature, respectively.

(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 [14]) based on interprocess communication with UNIX sockets rather than writing intermediate results in files.

(4) Filters to the commercial design packages StateMate™ and Together™ so that the numerical solvers of DSPNexpress can also be utilized for quantitative evaluation of system specifications with Harel state charts [9] as well as state charts and activity diagrams of the Unified Modeling Language (UML [7]).

(5) Open interface of the numerical solvers so that they can easily be utilized for the 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).

3 DSPNexpress Software Architecture

3.1 Methodological Results

Methodological results published in [14],[15],[16] introduced an approach for the cost-effective numerical analysis of DSPNs with concurrent deterministic transitions. The approach is based on representing the marking process of the DSPN as a finite-state generalized semi-Markov process (GSMP) with exponential and deterministic events.

Subsequently, transient and steady-state analysis of this GSMP is performed by considering a general state space Markov chain (GSSMC) embedded at equidistant time points nD (n=1,2,...) of the GSMP. We showed that both the continuous-time GSMP representing the marking process of the DSPN and the discrete-time GSSMC have the same limiting distributions provided that such limits exist [14]. Otherwise, these two processes have the same time-averaged distribution. The GSSMC is completely specified by a transition kernel and an initial distribution at time t = 0.

The algorithmic generation of the simplest form of the transition kernel of this GSSMC given the building blocks of the GSMP is discussed in a recent paper [16]. 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. In general, entries of the transition kernel of a GSSMC are functions of clock readings associated with the current state (i.e., functions in c_1 and c_2) and functions for clock readings associated with the new state (i.e., functions in a_1 and a_2). In general, the transition kernel of the GSSMC, denoted by $\mathbf{P}(c_1,c_2,a_1,a_2)$, has the form:

$$\mathbf{P}(c_1,c_2,a_1,a_2)=\left(\begin{array}{c|c|c} \mathbf{P}_{11} & \mathbf{P}_{12}(a_1) & \mathbf{P}_{13}(a_1,a_2) \\ \hline \mathbf{P}_{21}(c_1) & \mathbf{P}_{22}(c_1,a_1) & \mathbf{P}_{23}(c_1,a_1,a_2) \\ \hline \mathbf{P}_{31}(c_1,c_2) & \mathbf{P}_{32}(c_1,c_2,a_1) & \mathbf{P}_{33}(c_1,c_2,a_1,a_2) \end{array}\right)\begin{array}{c} 1 \\ \vdots \\ N_1 \\ \hline N_1+1 \\ \vdots \\ N_1+N_2 \\ \hline N_1+N_2+1 \\ \vdots \\ N \end{array}$$

$$1 \qquad N_1 \;\; N_1+1 \qquad N_1+N_2 \;\; N_1+N_2+1 \qquad N$$

Here, N_1 and N_2 denote the number of tangible markings of the DSPN enabling only exponential transitions and the number of tangible markings in which exactly one deterministic transition is enabled. The total number of tangible markings of the DSPN is denoted by N. Thus, the number of tangible markings in which two deterministic transitions are concurrently enabled is given by $N - N_1 - N_2$.

In [16] we present four theorems that provide the foundation for an algorithmic generation of the transition kernel. First, we formally proof that kernel elements $p_{ij}(.)$ can always be computed by appropriate summation of transient state probabilities of continuous-time Markov chains. Second, we derive a set of conditions on the building blocks of the GSMP (and, hence structural properties of the DSPN) under which kernel elements are constant; i.e., $p_{ij}(.) = k_{ij}$ for $0 < c_1,c_2,a_1,a_2 \le D$ where k_{ij} is a positive real number. Third, we proof that functional kernel elements are always separable. That is, functional kernel elements

depending on several clock readings can be expressed as the sum and/or product of constants, functions depending on just a single new clock reading and functions taking into account just a single old clock reading. Fourth, we derive conditions on the building blocks of the GSMP (and, hence structural properties of the DSPN) for which state probabilities $\pi_i(a_1, a_2)$ are symmetric in respect to clock values of deterministic events concurrently active. That is $\pi_i(a_1, a_2) = \pi_i(a_2, a_1)$ for $0 < a_1, a_2 \le D$. The exploitation of the properties of the transition kernel substantially reduces the computing time and memory requirements for the numerical solution of the system of Fredholm integral equations.

The form of the multidimensional Fredholm integral equations that constitute the time-dependent and stationary equations of the GSSMC have been presented in [14] and [15]. To write the system of time-dependent equations for the GSSMC in vector notation, we define three vectors of transient state probabilities at time nD, n = 1,2,..., respectively.

$$\begin{aligned}
&\pi_{exp}^{(n)} = \left(\pi_1^{(n)}, \pi_2^{(n)}, \ldots, \pi_{N_1}^{(n)}\right) \\
&\pi_{det1}^{(n)}(a_1) = \left(\pi_{N_1+1}^{(n)}(a_1), \pi_{N_1+2}^{(n)}(a_1), \ldots, \pi_{N_1+N_2}^{(n)}(a_1)\right) \\
&\pi_{det2}^{(n)}(a_1, a_2) = \left(\pi_{N_1+N_2+1}^{(n)}(a_1, a_2), \pi_{N_1+N_2+2}^{(n)}(a_1, a_2), \ldots, \pi_{N}^{(n)}(a_1, a_2)\right)
\end{aligned} \tag{1}$$

Furthermore, we introduce functions $y^{(n)}(c_1)$ and $z^{(n)}(c_1, c_2)$ as short hand notation for derivatives of transient state probabilities at time nD. That is:

$$y^{(n)}(c_1) = \frac{d\pi_{det1}^{(n)}(c_1)}{dc_1} \quad \text{and} \quad z^{(n)}(c_1, c_2) = \frac{\partial^2 \pi_{det2}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \tag{2}$$

As shown in [13], [14] the GSSMC approach allows the numerical analysis of DSPNs with concurrent deterministic transitions with different delays. However, for ease of exposition we present only the restricted case that all deterministic transitions of the DSPN have the same firing delay D. The extension of the time-dependent equations for DSPNs with deterministic transitions having different delays can be performed exactly as for the system of stationary equations introduced in [13]. Then, using the submatrices $\mathbf{P}_{ij}(.)$ of the transition kernel, time-dependent state probabilities for the GSSMC underlying a DSPN with two deterministic transitions concurrently enabled can be derived by the (discrete-time) forward Chapman-Kolmogorov equations. Thus, with (1) and (2) for n = 0,1,2,... we have:

$$\begin{aligned}
\pi_{exp}^{(n+1)} = \pi_{exp}^{(n)} \cdot \mathbf{P}_{11} &+ \int_0^D y^{(n)}(c_1) \cdot \mathbf{P}_{21}(c_1) dc_1 \\
&+ \int_0^D \int_0^{c_2} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{31}(c_1, c_2) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{31}(c_2, c_1) dc_1 dc_2
\end{aligned} \tag{3}$$

$$\pi_{det1}^{(n+1)}(a_1) = \pi_{exp}^{(n)} \cdot \mathbf{P}_{12}(a_1) + \int_0^{a_1} y^{(n)}(c_1) \cdot \mathbf{P}_{22}(c_1, a_1) dc_1 + \int_{a_1}^{D} y^{(n)}(c_1) \cdot \mathbf{P}_{22}(c_1, a_1) dc_1$$

$$+ \int_0^{a_1} \int_0^{c_2} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{32}(c_1, c_2, a_1) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{32}(c_2, c_1, a_1) dc_1 dc_2 \qquad (4)$$

$$+ \int_{a_1}^{D} \int_0^{a_1} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{32}(c_1, c_2, a_1) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{32}(c_2, c_1, a_1) dc_1 dc_2$$

$$\pi_{det2}^{(n+1)}(a_1, a_2) = \pi_{exp}^{(n)} \cdot \mathbf{P}_{13}(a_1, a_2) + \int_0^{a_1} y^{(n)}(c_1) \cdot \mathbf{P}_{23}(c_1, a_1, a_2) dc_1 + \int_{a_1}^{a_2} y^{(n)}(c_1) \cdot \mathbf{P}_{23}(c_1, a_1, a_2) dc_1$$

$$+ \int_0^{a_1} \int_0^{c_2} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{33}(c_1, c_2, a_1, a_2) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{33}(c_2, c_1, a_1, a_2) dc_1 dc_2 \qquad (5)$$

$$+ \int_{a_1}^{a_2} \int_0^{a_1} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{33}(c_1, c_2, a_1, a_2) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{33}(c_2, c_1, a_1, a_2) dc_1 dc_2$$

for $0 < a_1, a_2 \le D$ with $a_1 \le a_2$

$$\pi_{det2}^{(n+1)}(a_1, a_2) = \pi_{exp}^{(n)} \cdot \mathbf{P}_{13}(a_1, a_2) + \int_0^{a_2} y^{(n)}(c_1) \cdot \mathbf{P}_{23}(c_1, a_1, a_2) dc_1 + \int_{a_2}^{a_1} y^{(n)}(c_1) \cdot \mathbf{P}_{23}(c_1, a_1, a_2) dc_1$$

$$+ \int_0^{a_2} \int_0^{c_2} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{33}(c_1, c_2, a_1, a_2) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{33}(c_2, c_1, a_1, a_2) dc_1 dc_2 \qquad (6)$$

$$+ \int_{a_2}^{a_1} \int_0^{a_2} z^{(n)}(c_1, c_2) \cdot \mathbf{P}_{33}(c_1, c_2, a_1, a_2) + z^{(n)}(c_2, c_1) \cdot \mathbf{P}_{33}(c_2, c_1, a_1, a_2) dc_1 dc_2$$

for $0 < a_1, a_2 \le D$ with $a_2 \le a_1$

The system of equations (3) to (6) constitutes a system of two-dimensional Fredholm integral equations of the second type already written in an iterative scheme for its numerical solution. This iterative scheme is called *Picard iteration*. Due to the decomposition in disjoint subregions, all elements of $\mathbf{P}(c_1,c_2,a_1,a_2)$ are piece-wise continuous. Thus, the iterative scheme (3) to (6) converges to the stationary or time-averaged solution when n goes to infinity. Moreover, by taking the limits $n \to \infty$ in (3) to (6), we derive the system of stationary equations for the GSSMC underlying a DSPN with two deterministic transitions concurrently enabled. For steady-state analysis, as described in [13], [14] the system of Fredholm integro-differential equations (3) to (6) can be converted to system of a pure Fredholm integral for which efficient numerical solution techniques based on direct quadrature and subsequent solution of one large but very sparse linear system are known [8].

Note that if the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ is symmetric with respect to clocks of concurrent deterministic events, we have $\pi_i(a_1,a_2) = \pi_i(a_2,a_1)$ for $0 < a_1,a_2 \le D$. As a

consequence, Equation (6) need not be evaluated and two-dimensional integrals can be simplified in (3) to (5). That is e.g.,

$$\int_0^{a_2}\int_0^{c_2} z^{(n)}(c_1,c_2)\cdot\mathbf{P}_{33}(c_1,c_2,a_1,a_2)+z^{(n)}(c_2,c_1)\cdot\mathbf{P}_{33}(c_2,c_1,a_1,a_2)dc_1dc_2$$
$$=2\int_0^{a_2}\int_0^{c_2} z^{(n)}(c_1,c_2)\cdot\mathbf{P}_{33}(c_1,c_2,a_1,a_2)dc_1dc_2$$

3.2 Organization 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 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 [13]. 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. A multithreaded execution can be employed for the computation of the entries of the probability matrix of the EMC using the sockets PMATRIX<1> to PMATRIX<K>. Similarly, the conversion factors required by the EMC approach are passed through the sockets CMATRIX<1> to CMATRIX<K>. Subsequently, a linear system corresponding to the stationary equations of the EMC is solved and the state probabilities of the continuous-time marking process of the DSPN are derived using the conversion factors. These task are performed in the submodules *Derive EMC* and *Solve Linear System*. These software modules constituted the core of version 1.5 of DSPNexpress.

DSPNexpress 2000 contains two new software modules: *Derive GSSMC* and *Solve Fredholm equations*. As mentioned above, transient analysis of DSPNs is based on the analysis of an embedded GSSMC. The Chapman Kolmogorov equations of the GSSMC

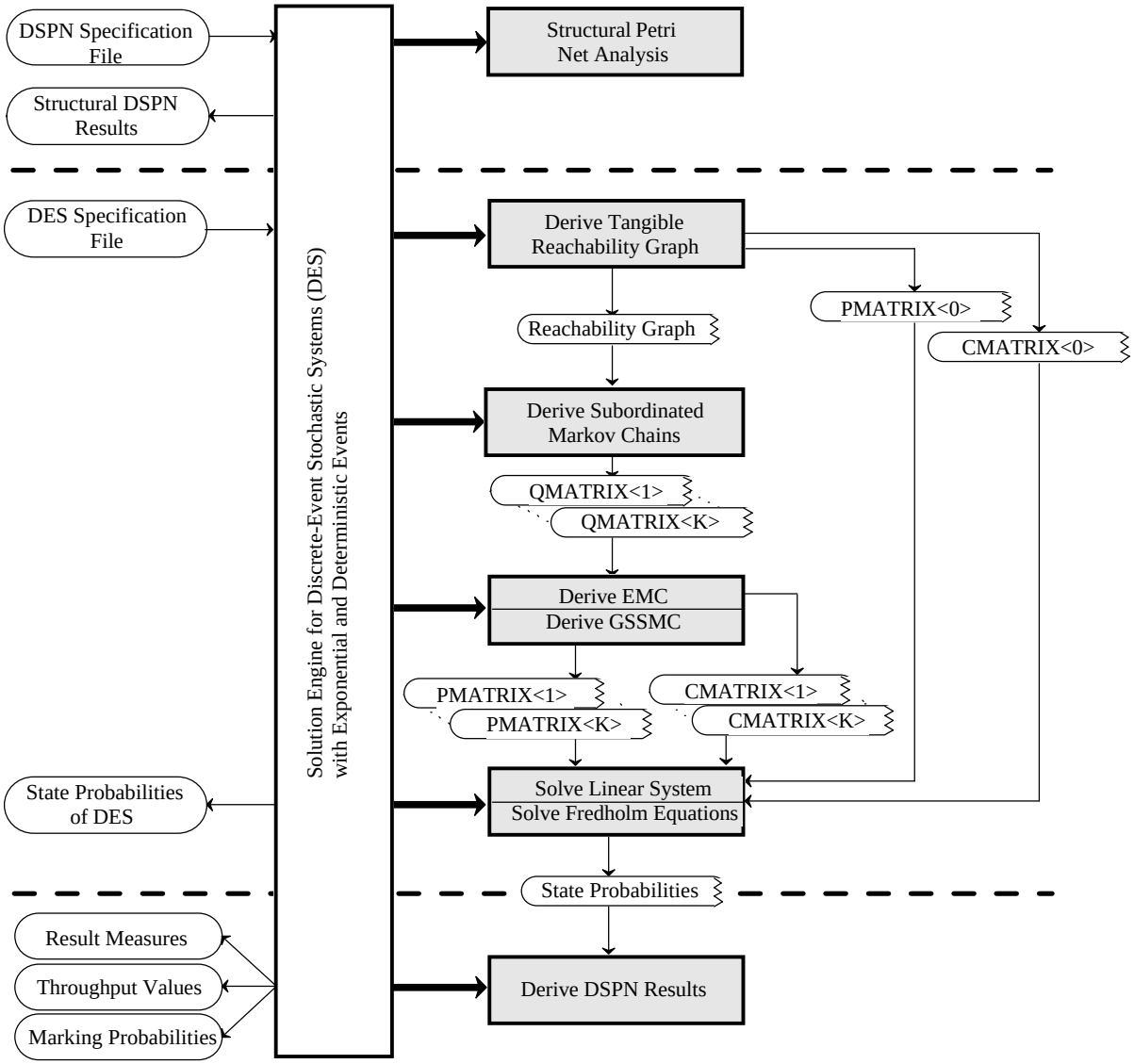

Figure 1. The software architecture of the numerical solvers of DSPNexpress 2000

constitute a system of Fredholm integral equations introduced in (3) to (6). Steady state analysis of DSPNs with concurrent deterministic transitions relies on the same approach [14]. Numerical computation of kernel elements of the GSSMC relies also on transient analysis of subordinated Markov chains and subsequent summation of appropriately selected transient probabilities [13]. This task is performed in submodule *Derive GSSMC*. As in case of the computation of the entries of the propability matrix of the EMC, a multithreaded execution can be employed for determining the kernel elements using the sockets PMATRIX<1> to PMATRIX<K>. Note that conversion factors are not required in the GSSMC approach. Thus, the sockets CMATRIX<1> to CMATRIX<K> are not used.

After *Derive GSSMC* has completed, for transient analysis of DSPNs the number of iterations corresponding to the mission time is performed on the system of Fredholm equations (3) to (6) whereas for steady state analysis one large but very sparse linear system is solved using GMRES. This task is performed in the submodule *Solve Fredholm Equations*.

As indicated in Figure 1 only the front end and back end of the solution engine are tailored to DSPNs.

3.3 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 2000*, the affiliation of the authors, *University of Dortmund*, *Computer Systems and Performance Evaluation Group*, and the year of release *1999*. A DSPN of a two-server, finite-capacity queue 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 help messages 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.

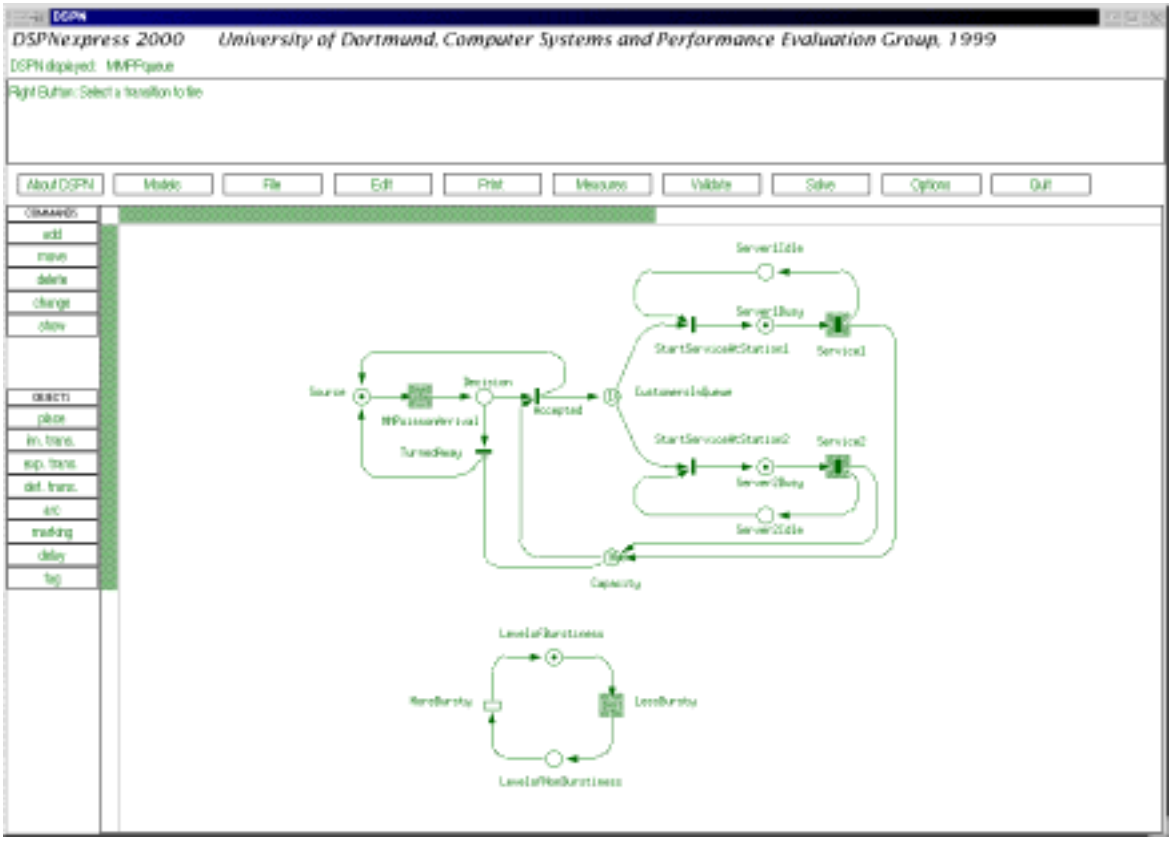

Figure 2. The graphical user interface of DSPNexpress

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*, *MMPoissonArrival*, *Decision*, *Accepted*, etc.). A detailed description of the features of the graphical interface is given in Chapter 10 of [13].

4 Application Examples

To illustrate the practical applicability of the DSPNexpress software for transient and steady-state analysis of DSPNs, we consider DSPNs of two queueing systems of high interest for communication network performance analysis. For these two DSPNs we present curves for CPU solution time and memory requirements versus model size. 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 *times*.

Figure 3 shows a DSPN of an MMPP/D/2/K queue already displayed in the working area of the screenshot of DSPNexpress in Figure 2. The K tokens residing in place *Capacity* in the initial marking represent the finite number of buffers of the queueing system. The token residing the subnet comprising of the places *Bursty mode* and *Normal mode* controls the mean firing time of the exponential transition *Markov modulated Poisson arrival*. That is, the Markov modulated Poisson arrival stream is represented by defining the firing delay of the corresponding exponential transition dependent on the location of the tokens in this subnet. Tokens contained in the places *Customers in queue* represent customers waiting in the queue. Tokens contained in the places *Server 1 busy* and *Server 2 busy* represent customers currently being served. The number of tangible markings of this DSPN is given by $2(K+2)$. In 2 markings are only exponential transitions enabled whereas in 4 markings exactly one deterministic transition is enabled. The number of tangible markings in which two deterministic transitions are concurrently enabled is given by $2(K-1)$. The constant service requirements are modeled by the deterministic transitions *Service 1* and *Service 2* which have firing delay D = 1.0. We assume that the immediate transitions *Start service at station 1* and *Start service at station 2* have both associated firing weights 1/2, such that arriving customers to an empty system join each server with equal probability. Since a service completion at either server leads to the same next tangible marking, the transition kernel of the GSSMC underlying the DSPN of Figure 3 is symmetric. Thus, Equation (6) of the system of Fredholm equations (3) to (6) need not be evaluated.

In all experiments, model parameters of the arrival process are set such that the effective arrival rate $\lambda_{eff} = 0.9$. For the transient analysis we set the initial distribution such that with

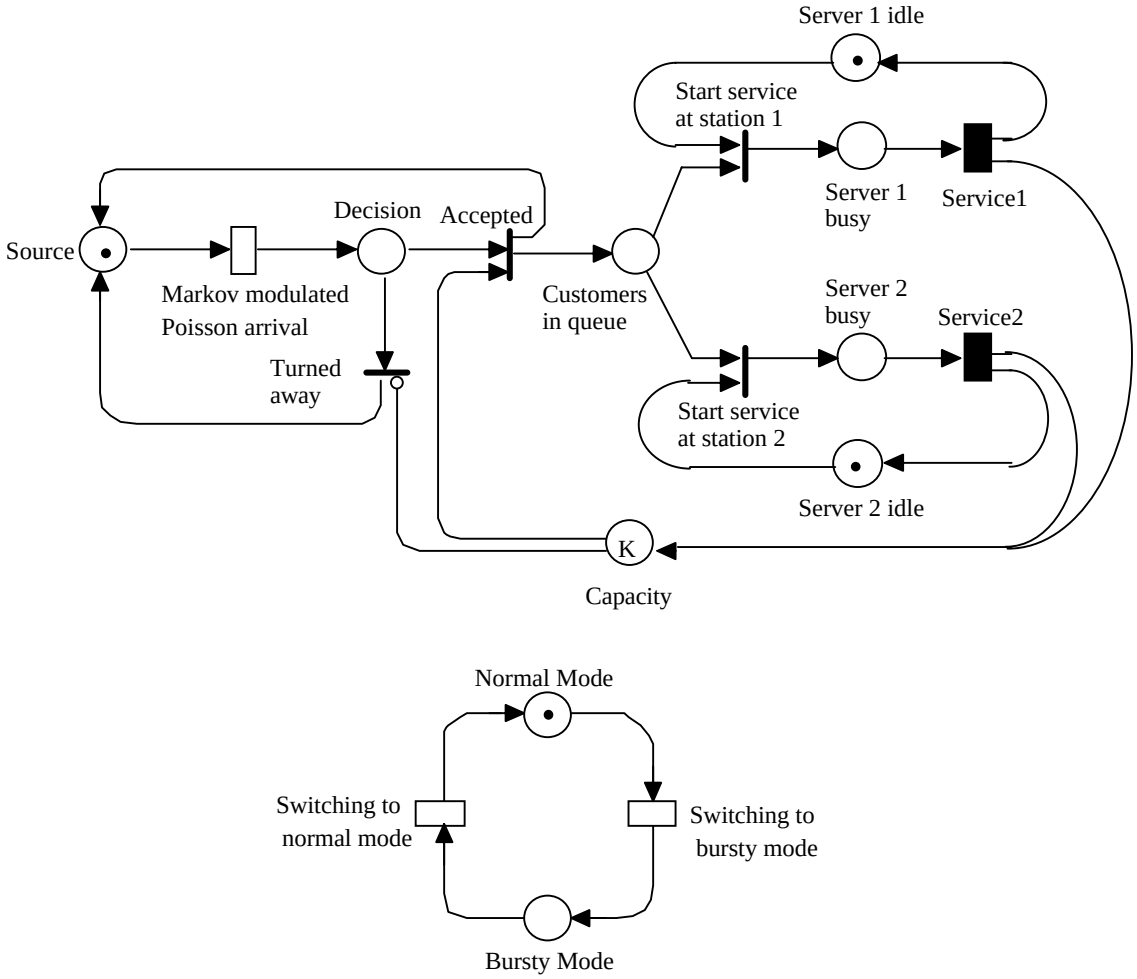

Figure 3. DSPN of an MMPP/D/2/K queue

probability 1.0 no customers reside in the system at time t = 0 and that the arrival process is in normal mode. The number of discretization steps employed in each dimension in the composite Simpson quadrature rule for integral expressions in the iterative scheme is set to M = 10. As indicated by Figure 7, this leads to a numerical accuracy of more than 10^{-9}. The employment of higher-order quadrature rules like Gauss-Laguerre rules yield a numerical accuracy of at least 10^{-14} which is close to optimal on a Sun Sparc under the Solaris operating system.

Figure 4 plots the CPU time required for computing the transient solution at instant of time T = 100 and for the steady-state solution, respectively, versus the model size. For both transient and steady-state analysis, we observe a linear growth of CPU time. This is due to the exploitation that almost all kernel elements are constants rather than functionals as evidently illustrated in Table 1. The solver also exploits the separability of the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ in the iterative and direct solution of the Fredholm equations (3) to (5). Furthermore, the solver employs a dynamic sparsing method by setting both constant and functional kernel elements smaller than a given threshold $\varepsilon = 10^{-16}$ to zero. This results in an

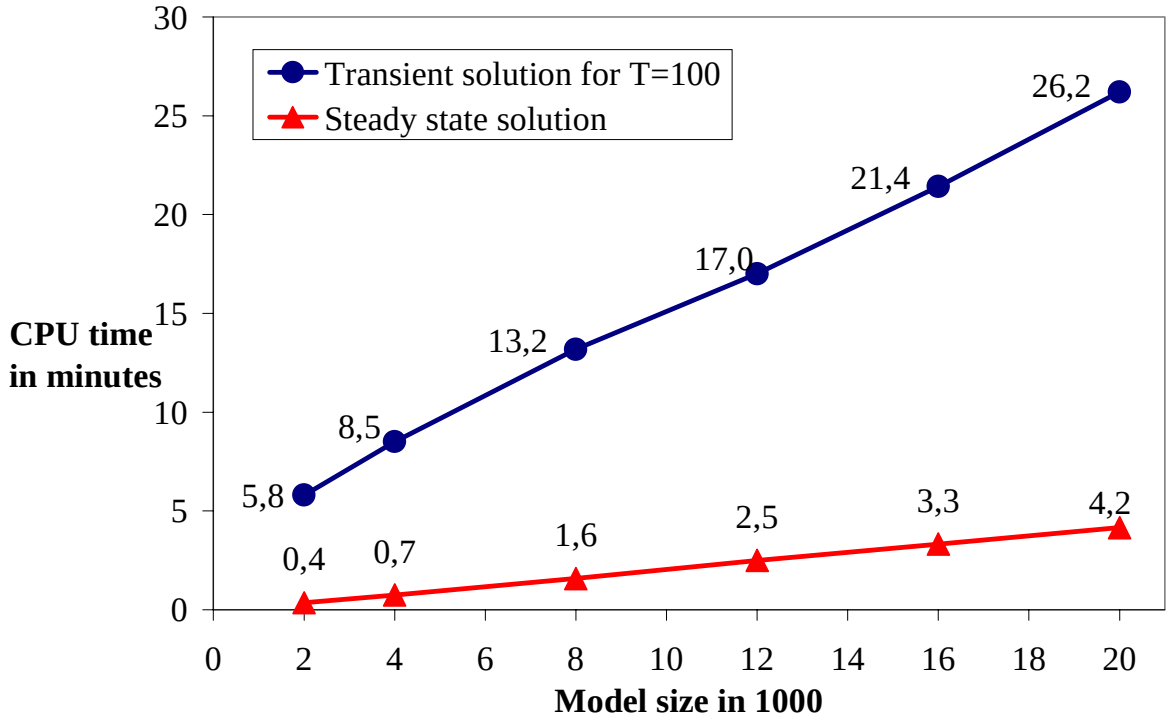

Figure 4. Transient and steady-state analysis of MMPP/D/2/K queue: CPU time versus model size

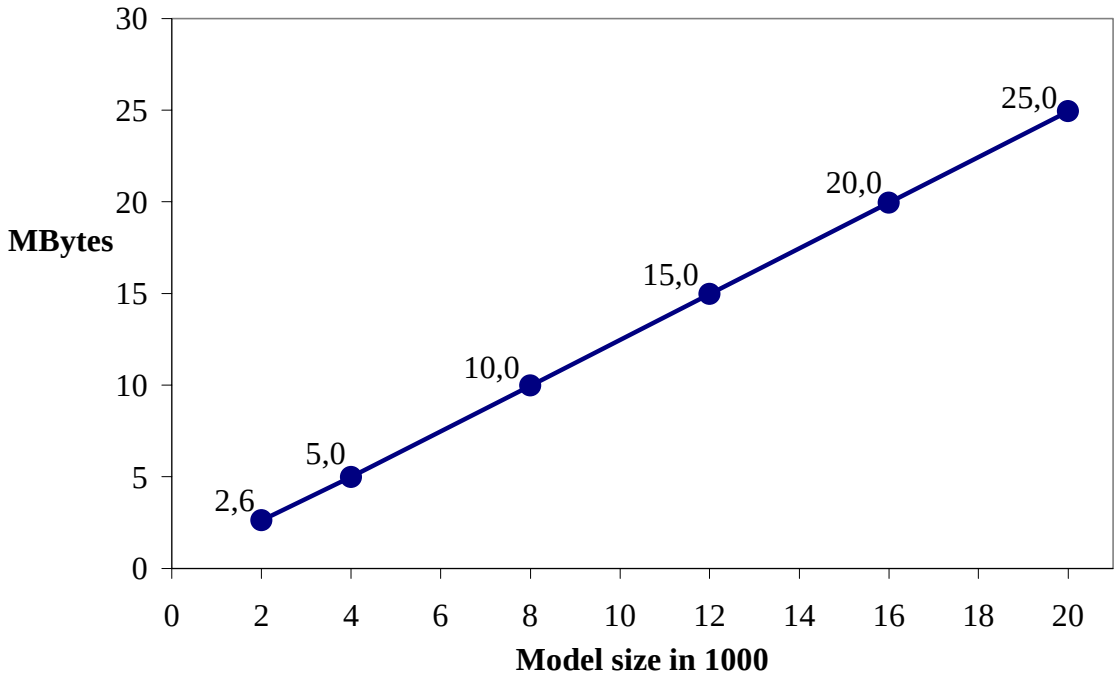

Figure 5. MMPP/D/2/K queue: memory requirements versus model size

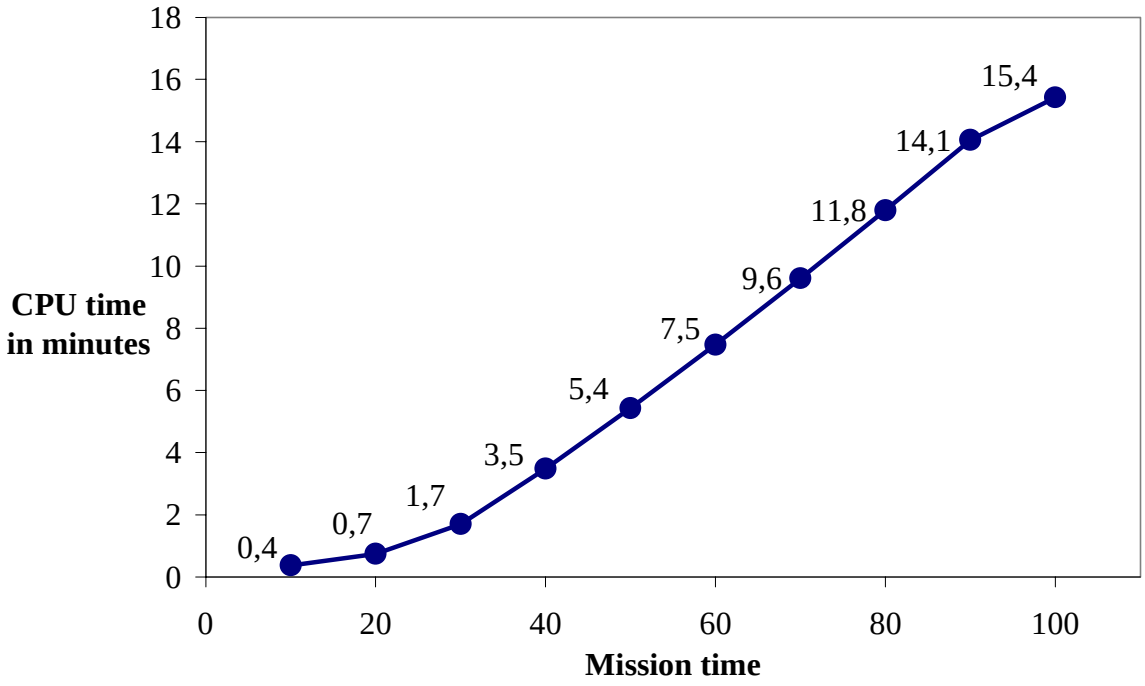

Figure 6. Transient analysis of MMPP/D/2/K queue: CPU time versus mission time

almost linear growth of the nonzero kernel elements for this class of DSPN models. Figure 5 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 a second experiment, the model size is kept fixed to 10020 and the mission time (i.e., the number of iterations that have to be performed by the iterative scheme) is varied from 10 to 100. As expected, Figure 6 shows a linear growth of CPU time for increasing mission time, since in each step of the iterative scheme a constant number of vector matrix multiplications is performed.

States of GSSMC	Nonzero entries	Constant entries	Functionals in 1 variable	Functionals in 2 variables	Functionals in 3 variables	Functionals in 4 variables
2004	2004997	99,30 %	0,20 %	0,49 %	$2{,}5\cdot10^{-4}$ %	$1{,}0\cdot10^{-4}$ %
4008	8009997	99,65 %	0,10 %	0,25 %	$6{,}3\cdot10^{-5}$ %	$2{,}5\cdot10^{-5}$ %
6012	18014997	99,77 %	0,07 %	0,17 %	$2{,}8\cdot10^{-5}$ %	$1{,}1\cdot10^{-5}$ %
8016	32019997	99,83 %	0,05 %	0,12 %	$1{,}6\cdot10^{-5}$ %	$6{,}3\cdot10^{-6}$ %
10020	50024997	99,86 %	0,04 %	0,10 %	$1{,}0\cdot10^{-5}$ %	$4{,}0\cdot10^{-6}$ %
12024	72029997	99,88 %	0,03 %	0,08 %	$6{,}9\cdot10^{-6}$ %	$2{,}8\cdot10^{-6}$ %
14028	98034997	99,90 %	0,03 %	0,07 %	$5{,}1\cdot10^{-6}$ %	$2{,}0\cdot10^{-6}$ %
16032	128039997	99,91 %	0,02 %	0,06 %	$3{,}9\cdot10^{-6}$ %	$1{,}7\cdot10^{-6}$ %
18036	162044997	99,92 %	0,02 %	0,06 %	$3{,}1\cdot10^{-6}$ %	$1{,}2\cdot10^{-6}$ %
20040	200049997	99,93 %	0,02 %	0,05 %	$2{,}5\cdot10^{-6}$ %	$1{,}0\cdot10^{-6}$ %

Table 1. Classification of elements of the transition kernel of MMPP/D/2/K

Table 1 shows the number of nonzero entries of the transition kernel of the GSSMC for increasing model size; i.e., K = 1000 to 10000 and provides percentages for each of the five different types of kernel elements. Note that this table shows the number of kernel entries whose analytic expressions are nonzero. The employment of dynamic sparsing of kernel entries in the practical computational scheme leads to a substantial reduction of nonzero entries and, thus, of memory requirements. From Table 1 we observed that for this class of DSPN models, i.e., DSPNs corresponding to queueing systems, kernel entries which are functionals in 3 and 4 clock values occur very rarely and are independent of the model size. Furthermore, functional entries in 1 and 2 clock values grow linearly with increasing model size whereas constant entries grow quadratically. From Table 1, we observe that for this example more than 99% of nonzero kernel entries are constant. For DSPNs corresponding to queueing systems like the one shown in Figure 3, the exploitation of constant entries in the transition kernel is key for their highly efficient transient and steady-state analysis.

To justify the GSSMC approach implemented in DSPNexpress 2000 software, we compare numerical accuracy and CPU solution time with the well-known Greedy approach of approximating deterministic delays by an Erlang distribution. For this experiment, we consider an MMPP/D/1/K with arrival rate λ_{eff} = 0.9, service time D = 1.0, and K = 1000. As measure of accuracy, the time-averaged mean queueing length is considered. Since this DSPN does not contain concurrent deterministic transitions, this quantity can also be computed with the EMC approach of DSPNexpress1.5 to determine the numerical accuracy.

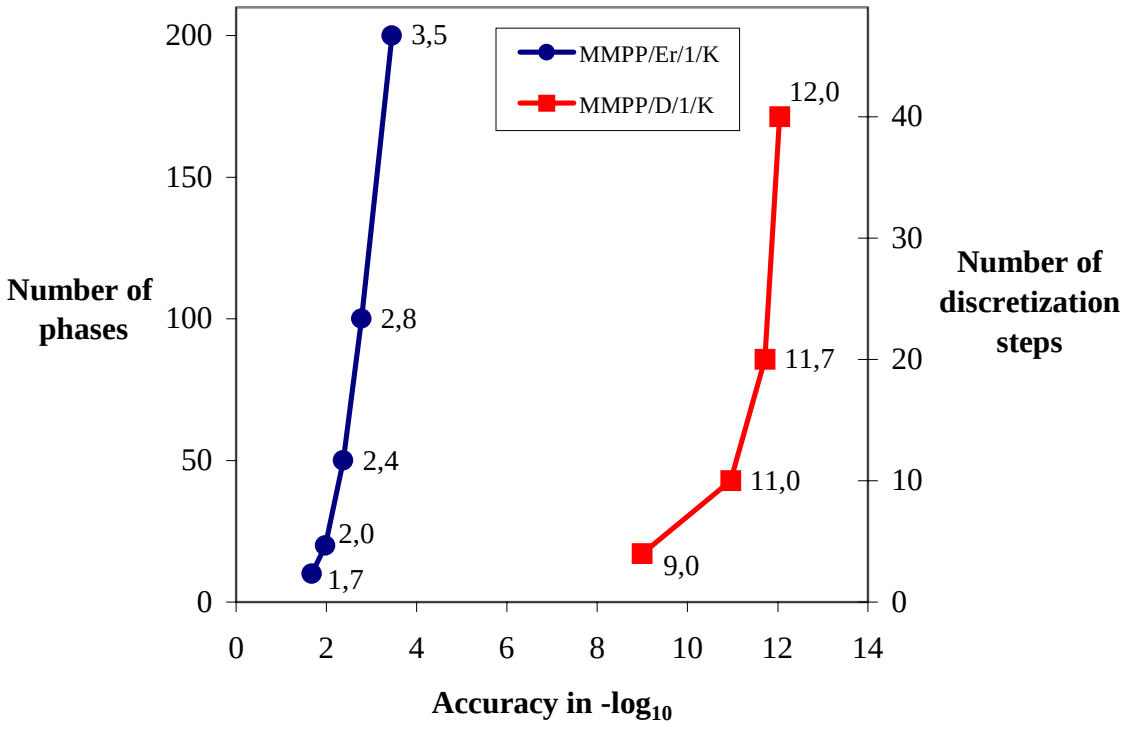

Figure 7. Accuracy for analysis of MMPP/D/1/K queue versus MMPP/E_r/1/K queue

For the GSSMC approach, the number of discretization steps for integral expressions in (3) to (5) was set to M = 5, 10, 20, and 40. Erlang distributions with r = 10, 20,.., 200 phases and mean value D = 1.0 were considered. Figure 7 in which the y-axis is drawn to scale with respect to the required CPU solution time evidently shows the benefits of the GSSMC approach for analysis of DSPNs; i.e., with the same amount of CPU solution time, the GSSMC approach yields an accuracy of 8 orders higher than the Erlang approximation.

Figure 8 shows a DSPN of an $M_i/D/M_i/K_i/2/L$ multiserver multiqueue system (MSMQ) with exponential walking times as introduced in [2]. The queues have a Markovian arrival stream with possibly different arrival rates λ_1 and λ_2, two deterministic servers, and capacities K_1 and K_2, respectively. As shown in Figure 8, the walking time between individual queueing systems is assumed to be exponentially distributed with rates μ_1 and μ_2, respectively. The MSMQ is comprised of two queues that receive arrivals from the external world and of two servers that cyclically attend the queues. Tokens residing in places *Capacity 1* and *Capacity 2* represent free buffers of each system. The two tokens residing in Figure 8 in place *Server available 1* represent the current position of the two servers. The exponential transitions *Walking 1 and Walking 2* have infinite server firing policy in order to take into account the concurrent walking of servers. After at most two customers received service at one queue, customers of the next queue will be served; i.e., limited service discipline. To explicitly model concurrent enabling of deterministic transitions, the DSPN submodels representing individual queues contain each two deterministic transitions; that is *Service 1,1* and *Service 2,1* as well as *Service 1,2* and *Service 2,2*. The conflicting immediate transitions have both firing weights 1/2 so that each server is chosen with equal probability by queued customers.

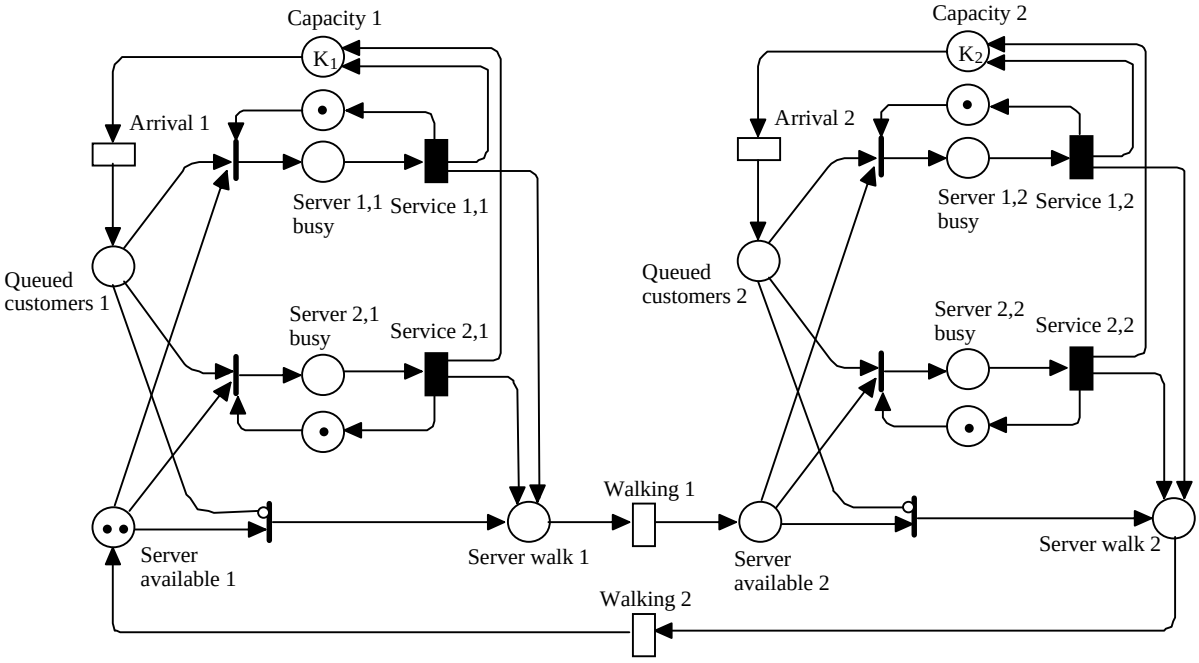

Figure 8. DSPN of an $M_i/D/M_i/K_i/2/L$ multiserver multiqueue system

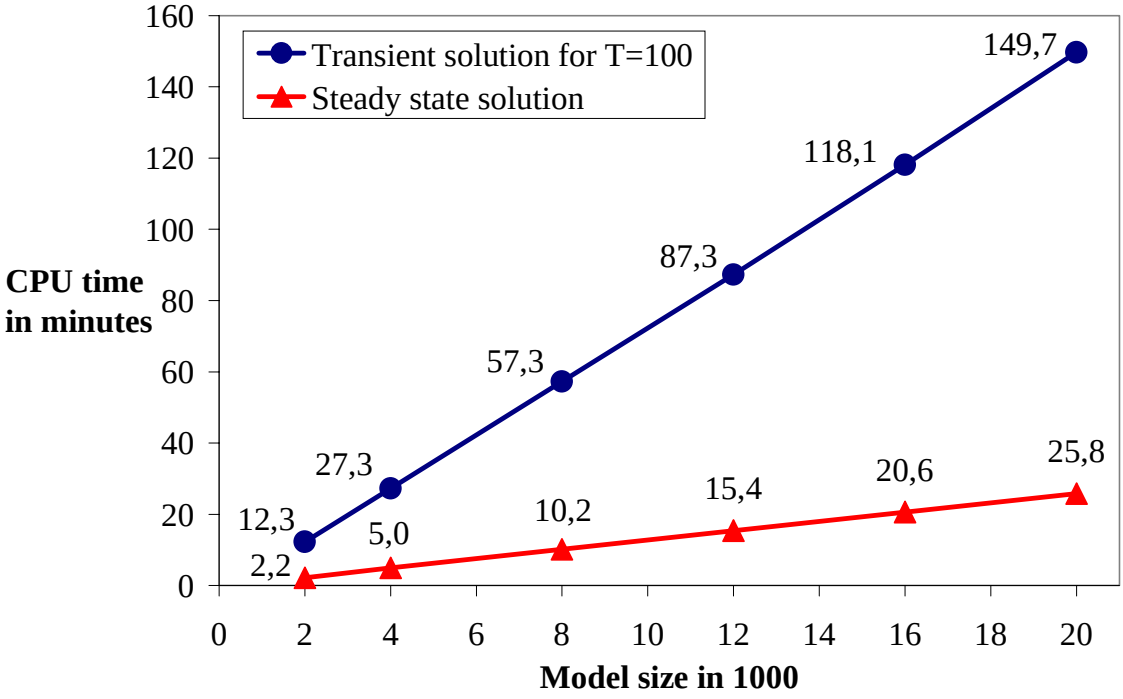

Figure 9. Transient and steady-state analysis of MSMQ system: CPU time versus model size

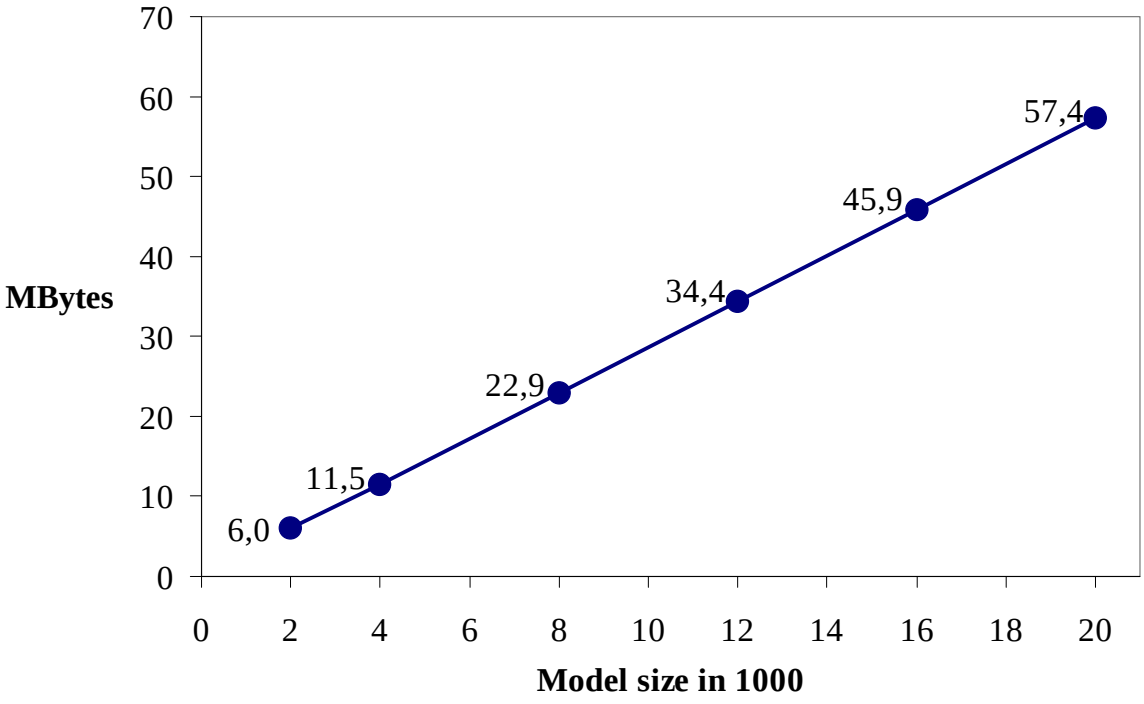

Figure 10. MSMQ system: memory requirements versus model size

The DSPN shown in Figure 8 has $17K_1K_2 + 7K_1 + 7K_2 + 1$ tangible markings. In $3K_1K_2 + 3K_1 + 3K_2 + 3$ markings are only exponential transitions enabled whereas in $8K_1K_2 + 4K_1 + 4K_2$ are exactly one deterministic transition enabled. The number of tangible markings in which two deterministic transitions are concurrently enabled is given by $6K_1K_2 - 2$. In all experiments, we consider arrival rates $\lambda_1 = \lambda_2 = 0.5$, walking times of the servers to the other queue with rates $\mu_1 = \mu_2 = 1.0$, and deterministic service times D = 1.0. Furthermore without loosing the representiveness of the experiments, we assumed $K_1 = K_2$.

Figures 10 and 11 plot CPU solution time and memory requirements versus increasing model size. As in case of the MMPP/D/2/K queue, both transient analysis for mission time T = 100 and steady-state analysis is considered. We observe that the curves of Figure 9 and 11 have the same shape as corresponding curves of Figure 4 and 6. Again, we observe a linear growth in CPU solution time and memory requirements for increasing model size. However, for a particular model size, say 20000, the analysis of MSMQ requires about 6 times as much CPU time as the MMPP/D/2/K. As shown in Figure 10, the memory requirements for the analysis of MSMQ are twice as large as for the MMPP/D/2/K queue. On a first glance, this observations looks surprising because for a given model size this DSPN contains considerably less markings in which two deterministic transitions are concurrently enabled than the DSPN of the MMPP/D/2/K queue. However, as illustrated by Table 2, the classification of the entries of the transition kernel provide the explanation for the high CPU solution time.

Table 2 provides percentages for each of the five different types of kernel elements for the MSMQ with exponential walking times. As before, this table shows the number of kernel entries whose analytic expressions are nonzero. Table 2 indicates that for this class of DSPN models, i.e., because of the exponential walking times, there is an almost even split between functionals kernel entries and constant kernel entries. This is opposed to Table 1 and the

States of GSSMC	Nonzero entries	Constant entries	Functionals in 1 variable	Functionals in 2 variables	Functionals in 3 variables	Functionals in 4 variables
1841	305035	27,83 %	23,33 %	18,64 %	16,35 %	13,85 %
4036	977358	27,27 %	22,86 %	18,75 %	16,82 %	14,30 %
5761	1493510	26,94 %	22,63 %	18,82 %	17,05 %	14,56 %
7792	2276822	26,72 %	22,58 %	18,86 %	17,12 %	14,72 %
10129	3590882	26,61 %	22,53 %	18,89 %	17,18 %	14,79 %
11857	4808125	26,54 %	22,48 %	18,91 %	17,22 %	14,85 %
13721	6323849	26,48 %	22,44 %	18,93 %	17,25 %	14,90 %
15721	8180660	26,42 %	22,41 %	18,94 %	17,28 %	14,94 %
17857	9629948	26,39 %	22,39 %	18,95 %	17,30 %	14,97 %
20129	11993229	26,37 %	22,37 %	18,95 %	17,32 %	14,99 %

Table 2. Classification of elements of the transition kernel of MSMQ system

single reason for the considerably higher CPU solution time shown in Figure 9. If the walking time distribution is assumed to be deterministic, the classification of kernel entries will be similar to Table 1. As a consequence, the CPU solution time will also be reduced substantially.

Conclusions

This paper introduced DSPNexpress 2000, the new version of a widely distributed software package for modeling with deterministic and stochastic Petri nets (DSPN). While the previous version of DSPNexpress was known for its highly efficient numerical solver for steady-state analysis of DSPNs without concurrent deterministic transitions [11], DSPNexpress 2000 also provides a method for transient analysis of DSPNs [15]. Furthermore, both the stationary analysis and the transient analysis is no longer restricted to the case that deterministic transitions cannot be concurrently enabled [14]. To illustrate the applicability of the newly implemented solvers of DSPNexpress 2000, we presented performance experiments for an MMPP/D/2/K queue and a multi-server multi-queueing system. We presented curves plotting the CPU time and memory requirements for transient and steady-state analysis versus model size and mission time, respectively. These curves evidently show that DSPNexpress 2000 can analyze quite complex DSPNs with two deterministic transitions concurrently active with reasonable computing time and memory requirements.

In current work, we are integrating the exploitation of special structures and isomorphisms presented for the analysis of DSPN without concurrent deterministic transitions in [12] in the GSSMC approach. Following the lines of [12], we expect a further reduction of CPU solution time by one order of magnitude.

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] M. Ajmone Marsan, S. Donatelli and F. Neri, GSPN Models of Multiserver Multiqueueing Systems, *Performance Evaluation,* **11**, 227-240, 1990.

[3] 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*, 224-238, Springer 1995.

[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] M. Fowler, *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley 1997.

[8] R.P. Kanwal, *Linear Integral Equations: Theory and Technique,* Birkhäuser 1997.

[9] D. Harel and M. Politi, *Modeling Reactive Systems with StateCharts: The StateMate Approach*, McGraw Hill 1998.

[10] A. Heindl and R. German, A Fourth Order Algorithm with Automatic Step Size Control for the Transient Analysis of DSPNs, *IEEE Trans. Softw. Engin.,* **25**, 194-206, 1999.

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

[12] Ch. Lindemann, Exploiting Isomorphisms and Special Structures in the Analysis of Markov Regenerative Stochastic Petri Nets, in: W.J. Stewart (Ed.) *Numerical Computations with Markov Chains*, 383-402, Kluwer 1995.

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

[14] Ch. Lindemann and G.S. Shedler, Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation, Special Issue Proc. of PERFORMANCE '96*, **27&28**, 565-582, 1996.

[15] Ch. Lindemann and A. Thümmler, Transient Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation, Special Issue Proc. of PERFORMANCE '99*, **39&40**, 34-51, 1999.

[16] Ch. Lindemann and A. Thümmler, Numerical Analysis of Generalized Semi-Markov Processes, 1999 (submitted for publication).

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

[18] W.H. Sanders, Integrated Frameworks for Multi-Level and Multi-Formalism Modeling, *Proc. 8th Int. Workshop on Petri Nets and Performance Models, Zaragoza Spain,* 2-11, 1999.

[19] M. Telek and A. Horvath, Transient Analysis of Age MRSPNs by the Method of Supplementary Variables, *Proc. 4th Int. Symp. on Parallel and Distributed Systems, Durham North Carolina*, 1998.