Specifications and Solution Techniques for Non-Markovian Stochastic Petri Nets*

Andrea Bobbio
Dipartimento di Informatica
Università di Torino, 10149 Torino, Italy

Miklós Telek
Department of Telecommunications
Technical University of Budapest, 1521 Budapest, Hungary

Abstract

Allmost all the available tools for the analysis of *Stochastic Petri Nets (SPN)* assume that the stochastic nature of the problem is restricted to be a Continuous Time Markov Chain *(CTMC)*, but in reality, there are instances in which the *CTMC* assumption is too weak. The evolution of the stochastic systems with non-exponential timing becomes a stochastic process, for which in general, no analytical solution is available.

In order to properly define Non-Markovian Stochastic Petri Nets special specifications should be added at the *PN* level. These specifications are usually referred to as the *firing policy*. The semantics of different firing policies appeared in the literature is discussed, together with their implication on the behavior of the associated marking process.

Different approaches and numerical techniques have been explored in the literature for dealing with non-Markovian *SPN*s: - techniques based on the theory of Markov regenerative processes; - techniques based on the use of supplementary variables; - techniques based on the approximation of the original non-Markovian process by means of a Markov chain defined over an extended state space. The paper explore the background of the above solution techniques with respect to different classes of models.

1 Introduction

The semantics of *PN* with generally distributed firing times has been considered for a long time. In [1], in order to completely define the behaviour of the marking process, each timed transition was assigned an individual memory policy specifying how the firing of the transition was dependent on its past history. The memory policy proposed in [1] was an attribute attached to each individual transition so that the memory of the

*M. Telek thanks the support of OTKA grant No. F-23971.

overall marking process resulted from the superposition of the memories of the individual transitions.

Based on the concepts defined in [1], Ajmone and Chiola developed the *Deterministic and Stochastic PN (DSPN)* model [2], where in each marking, a single transition is allowed to have associated a determinsitic firing time with enabling memory policy. Choi et al. have derived the transient solution of the same model in terms of a Markov regenerative process, and have subsequently extended the *DSPN* model by accommodating at most a single transition with generally distributed firing time [6] in each marking. They have called this class of models *Markov Regenerative Stochastic PN (MRSPN).* Further elaborations of *SPN* models with non exponential distributions but restricted to enabling memory policies only have been presented in [7, 10, 15].

Bobbio and Telek have enlarged the class of *MRSPN* by introducing the concept of marking processes with non-overlapping memories. In this new framework, they have accomodated into the model age memory policies [5, 17] and preemptive repeat identical policies [3, 4].

This paper summarizes the model specification and solution techniques of *MRSPN*s. The rest of the paper is organized as follows. Section 2 discusses the memory model of Petri net transitions and the considered *firing policies.* Section 3 introduceses the analysis techniques proposed for the numerical analysis of Markov Regenerative Stochastic Petri Nets. An application example is evaluated in Section 4, and the paper is concluded in Section 5.

2 The individual memory model

A marked Petri Net is a tuple $PN = (P, T, I, O, H, M_0)$, where: P is the set of places, T the set of transitions, I, O and H are the input, the output and the inhibitor functions, respectively, and M_0 is the initial marking. The reachability set $\mathcal{R}(M_0)$ is the set of all the markings that can be generated from the initial marking M_0. The marking process $\mathcal{M}(t)$ denotes the marking occupied by the *PN* at time t.

We define a non-Markovian *SPN* as a stochastically timed *PN* in which the time evolution of the marking process cannot be mapped into a *Continuous Time Markov Chain (CTMC).* In the spirit of many modeling formalisms [13], in which the complexity of the solution must be hidden to the modeler, a complete set of specifications must be given at the *PN* level, in order to univocally define the underlying marking process. Therefore, the way in which the future evolution of the marking process depends on its past history needs to be specified at the *PN* level.

The most consistent way to introduce memory into a *SPN* is provided in [1]. Each timed transition t_g is assigned a random firing time γ_g with a general distribution $G_g(t)$ with support on $[0, \infty)$. A clock, associated to each individual transition, counts the time in which the transition has been enabled. An *age variable* a_g associated to the timed transition t_g keeps track of the clock count. A timed transition fires as soon as the age variable a_g reaches the value of the firing time γ_g. A very similar formulation, in the simulation setting, has been discussed by Haas and Shedler [12].

In the original view [1], two main firing policies were introduced:

- *enabling memory* if the age variable a_g is reset each time the corresponding tran-

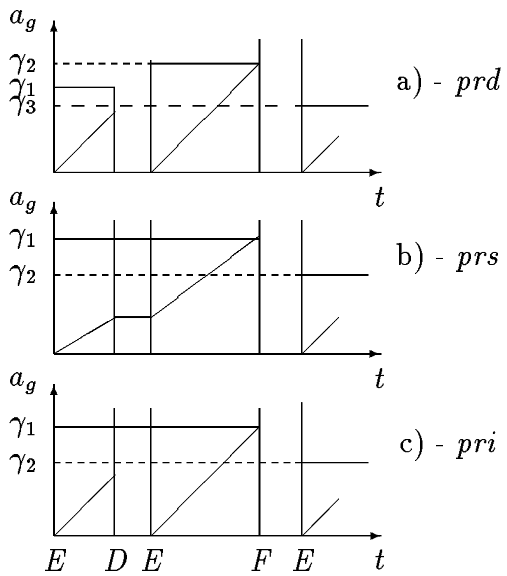

Figure 1: Pictorial representation of different firing policies.

sition t_g is disabled or fires;

- *age memory* if the age variable a_g is reset only when the corresponding transition t_g fires.

We define the *activity period* of a transition t_g as the interval of time during which the corresponding age variable a_g is different from 0. In [1], the firing time was implicitly assumed to be resampled at the beginning of any activity period of the transition.

However, in a more general view, the random firing time γ_g of a transition t_g can be sampled in a time instant antecedent to the beginning of an activity period. To keep track of the resampling condition of the random firing time associated to a timed transition, we assign to each timed transition t_g a binary indicator variable ι_g that is equal to 1 when the firing time is sampled and equal to 0 when the firing time is not sampled. We refer to ι_g as the *resampling indicator variable.* When a transition enters an activity period, if the resampling indicator variable ι_g is zero, the firing time is resampled and ι_g is switched to 1; whereas, if ι_g is already equal to 1, the firing time is not resampled. ι_g is reset to 0 at each firing of t_g. We define the *resampling period* of a transition as the time interval during which the indicator variable ι_g is equal to 1, i.e. the firing time of the transition maintains its constant value without any intermediate resampling.

The *active period* and the *resampling period* are elements of memory of the marking process. Hence, in general, the memory of a transition t_g is captured by the tuple (a_g, ι_g). At any time epoch t, transition t_g has memory (its firing process depends on the past) if either a_g or ι_g are different from zero.

Adopting the previous formalization of the memory concept, the following individual memory policies have been introduced in the past. A timed transition t_g can be (Figure 1):

- *Preemptive resume (*prs*)*:
  If the associated clock counts the time according to an age memory policy and the

firing time is resampled when the transition becomes active. More formally, both the age variable a_g and the resampling indicator ι_g are reset only when t_g fires.

- *Preemptive repeat different (*prd*):*
  If the associated clock counts the time according to an enabling memory policy and the firing time is resampled when the transition becomes active. More formally, both the age variable a_g and the resampling indicator ι_g are reset each time t_g is disabled.

- *Preemptive repeat identical (*pri*):*
  If the associated clock counts the time according to an enabling memory policy but the firing time can be resampled only after the transition fired. More formally, the age variable a_g is reset each time t_g is disabled but the resampling indicator ι_g is reset only when t_g fires.

In the described individual memory models, a *prs* transition cannot be disabled and restarted before firing, and a *pri* transition cannot be resampled before firing.

3 Markov Regenerative Stochastic Petri Nets

The first definition of the class of *MRSPN*s comes from Choi et al. [6]:

Definition 1 *A SPN is called a Markov Regenerative Stochastic Petri Net if its marking process is a Markov regenerative process (*MRGP*).*

*MRGP*s [14] are discrete state continuous time stochastic processes with embedded Regenerative Time Points (*RTP*), at which the process enjoys the Markov Property. Based on the concept of memory of the general transitions *RTP*s can be defined as follows:

Definition 2 *A regeneration time point in the marking process is an instance of time when all the active and sampled time interval of the general transitions are concluded.*

The importance of these definitions comes from the fact that *MRSPN*s can be studied by the results available for *MRGP*s [14]. The analysis methods of *MRSPN*s published in the literature so far are based on one of the following approaches: Markov Renewal Theory [6, 5]; Method of Suplementary Variable [15, 10]; Approximate analysis by Phase type expansion [9].

3.1 Analysis by Markov Renewal Theory

By the memoryless property of the *MRGP*s in the *RTP*s the analysis of a *MRSPN* can be divided into independent subproblems which are the analysis of the stochastic (subordinated) processes between the consecutive *RTP*s, called regeneration periods. The measures required for the transient analysis of *MRSPN*s based on the Markov regenerative theory are commonly refered to as global and local kernels. The global kernel describes the occurence of the consecutive *RTP*:

$$K_{ij}(t) = Pr\{M_{(1)} = j \, , \, \tau_1^* \leq t | \, \mathcal{M}(0) = i\}$$

where $\mathcal{M}(t)$ denotes the marking process τ_1^* is the next *RTP* and $M_{(1)}$ is the right continuous state of the marking process at the next *RTP*. The local kernel describes the state transitions probabilities up to the consecutive *RTP*:

$$E_{ij}(t) = Pr\{\mathcal{M}(t) = j \, , \, \tau_1^* > t | \, \mathcal{M}(0) = i\}$$

The analysis of this measures is a function of the memory policy of the general transition dominates the regenerative period, i.e. the transition whose sampling period coincidences with the tagged regenerative period. For a *prd* type general transition the analysis is given in [6], for a *prs* type in [5] and for a *pri* type in [3].

Based on the global and the local kernels the transient analysis can be carried out in time

$$V_{ij}(t) \; = \; E_{ij}(t) + \sum_k \int_0^t \, d\,K_{ik}(y) \; V_{kj}(t-y)$$

or in transform domain

$$\mathbf{V}^{\sim}(s) \; = [\mathbf{I} \; - \; \mathbf{K}^{\sim}(s)]^{-1} \; \mathbf{E}^{\sim}(s)$$

where $V_{ij}(t)$ denotes the state transition probability over $(0,t)$, i.e.: $V_{ij}(t) = Pr\{\mathcal{M}(t) = j \,|\, \mathcal{M}(0) = i\}$ and $\mathbf{V}^{\sim}(s)$ is the Laplace-Stieltjes transform of the transition probability matrix.

For the purpose of the steady state analysis an *MRSPN* the following measures of the subordinated processes should be known:

$$\alpha_{ij} = E[\int_0^\infty I_{\mathcal{M}^{(i)}(t)=j} dt] \; ; \qquad \pi_{ij} = Pr\{M_{(1)} = j \mid \mathcal{M}(0) = i\}$$

α_{ij} is the expected time a subordinated process starting from state i spends in state j, and π_{ij} is the probability that the subordinated process starting from state i is followed by a subordineted process starting from state j. Indeed the matrix $\mathbf{\Pi} = \{\pi_{ij}\}$ is the transition probability matrix of the discrete time Markov chain embedded into the *RTP*s.

The analysis of these measures are also conditional to the type of the dominant transition of the subordinated processes. For a *prd* type general transition the analysis is given in [2], for a *prs* type in [17] and for a *pri* type in [4]. These measures can also be obtained from the global and local kernels eighter in time and transform domain:

$$\alpha_{ij} = \int_{t=0}^\infty E_{ij}(t) dt = \lim_{s \to 0} E_{ij}^{\sim}(s)/s \; ; \qquad \pi_{ij} = \lim_{t \to \infty} K_{ij}(t) = \lim_{s \to 0} K_{ij}^{\sim}(s)$$

The steady state analysis of an *MRSPN* based on these measures is a 2-step method:

Step 1: Evaluate $P = \{p_i\}$ the unique solution of:

$$P = P\mathbf{\Pi} \quad ; \qquad \sum_i p_i = 1$$

Step 2: The steady-state probabilities of the *MRGP* become:

$$v_j = \lim_{t \to \infty} Pr\{\mathcal{M}(t) = j\} = \frac{\sum_k p_k \; \alpha_{kj}}{\sum_k p_k \; \alpha_k}$$

3.2 Analysis by the Method of Suplementary Variable

The marking process ($\mathcal{M}(t)$) together with the age variable (a) of the dominant transition of an *MRSPN*[1] with at most one active *prd* type general transition is a Markov process over the state space $S \times R$, where S is the set of reachable tangible markings and R is the (sub)set positive real numbers.

The joint process can be analyzed by the method of supplementary variable [8] as shown in [10]. Following the concept and the notations of [11] we breafly summarize the approach.

Let T^G the set of general transitions. The state space is devided into $\#T^G + 1$ parts. S^E is the set of states in which no general transition is active ($a \doteq 0$), and $S^g, g \in T^G$ is the set of states in which the general transition t_g is active. The superscript ^{E} refers to the states in S^E and the superscript ^{g} (or ^{h}) refers to the states in S^g (or S^h). The probability of being in state n at time t is $\pi_n(t) = Pr[\mathcal{M}(t) = n]$. The, so called, *age rate* describes the state together with the age of the process at time t:

$$p_n(t,x) = \frac{Pr[\mathcal{M}(t) = n, x < a \leq x + dx]}{dx} \cdot \frac{1}{1 - F^g(x)}$$

The firing time distribution of transition t_g is $F^g(x)$. And the matrix, refered to as branching probability matrix, describes the state transition due to the firing of a general transition is denoted by $\Delta^{g,h}_{i,j}$.

With the use of proper vectors and matrices a system with *prd* transitions is characterized as follow.

Partial differential equation describes the process evolution in S^g:

$$\frac{\partial}{\partial t}\mathbf{p^g}(t,x) + \frac{\partial}{\partial x}\mathbf{p^g}(t,x) = \mathbf{p^g}(t,x)\mathbf{Q^g}$$

The age increases as fast as the time when general transition t_g is enabled.

Ordinary differential equation describes the process evolution in S^E:

$$\begin{aligned}\frac{d}{dt}\pi^{\mathbf{E}}(t) &= \pi^{\mathbf{E}}(t)\mathbf{Q^{E,E}} + \sum_g \pi^{\mathbf{g}}(t)\mathbf{Q^{g,E}} + \\ &\sum_g \int_0^\infty \mathbf{p^g}(t,x)dF^g(x)\mathbf{\Delta^{g,E}}\end{aligned}$$

State probabilities in S^E can change by the firing of an exponential transition (1st term), by disabling a general transition after which only exponential transitions are enebled (2nd term), or by firing of a general transition after which only exponential transitions are enebled (3rd term).

Boundary condition is given by:

$$\begin{aligned}\mathbf{p^g}(t,0) &= \pi^E(t)\mathbf{Q^{E,g}} + \sum_h \pi^h(t)\mathbf{Q^{h,g}} + \\ &\sum_h \int_0^\infty \mathbf{p^h}(t,x)dF^h(x)\mathbf{\Delta^{h,g}}\end{aligned}$$

[1]when t_g is the dominant transition $a = a_g$

General transition t_g can be activated by the firing of an exponential transtion in S^E (1st term), by disablig the active general transtion (2nd term), or by the firing of a general transition (3rd term).

The probability of states in which t_g is active are given by:

$$\pi^g(t) = \int_0^\infty \mathbf{p^g}(t,x)(1 - F^g(x))dx$$

Finally the initial conditions are $\pi^E(0)$ and $\mathbf{p^g}(0,x) = \pi^g(0)\delta(x)$.

The analysis of the transient behaviour by the Suplemantary Variable Approach is based on a numerical evaluation of the above system of equations. An iterative algoritm based on the fix size (h) discretization of the continuous variables proposed by German et al. [11] consists of the following steps:

1. compute age rates in the next time instant

   $$\mathbf{p^g}(ih, jh) = \mathbf{p^g}((i-1)h, (j-1)h)e^{\mathbf{Q^g}h}$$

   and set $\mathbf{p^g}(ih, 0) = 0$

2. compute the state probabilities $\pi^g(ih)$ by $\mathbf{p^g}(ih, jh)$, $j = 0, 1, \ldots$

3. compute the state probabilities $\pi^E(ih)$ by the ordinary differential equation

4. compute the activation rate of general transitions $\mathbf{p^g}(ih, 0)$ by the boundary conditions

5. check the convergence and go back to step 2 or start with the next time instant $(i+1)h$

The transient behaviour of an *MRSPN* by the Suplemantary Variable Approach can be easily obtained by vanishing the derivatives according to the time in the above set of equations. Lindemann proposed an effective numerical method to evaluate the steady state probabilities based on this approach [15].

3.3 Approximate analysis by Phase type expansion

The set on non-Markovian Stochastic Petri Nets with *prs* or *prd* type transitions can be approximatelly analyzed by the method of Phase type expansion. When the firing times are all Phase type distributed [16], this approach gives the exact solution.

The analysis method is composed by the following steps:

Step 1: Approximate the firing time distribution of all the timed transition by a Phase type distribution.

Step 2: Based on the net description, the Phase type model and the memory policy of the transitions compose the expanded state model of the stochastic process which is a CTMC over the state space $S \times T_1 \times \ldots \times T_n$, where S is the set of reachable tangible markings and T_g is the set of the phases of the Phase type model of transition t_g.

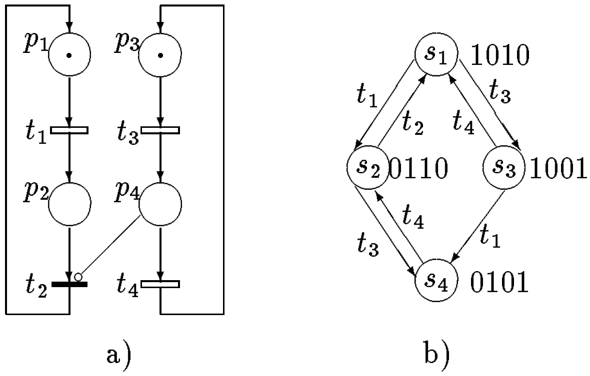

Figure 2: Petri net model of the processor system

Step 3: Analyze the expanded CTMC.

Step 4: Evaluate the marking probabilities and the other required Petri net measures.

Cumani has realized a package, called ESP, which automatically performes Step 2 to 4 [9].

4 Job execution in a processor system

A real life example is described and analyzed by a *MRSPN* in this section. Consider a processor system with two terminals, indicated as A and B, submitting jobs for execution. Jobs from terminal A require a generally distributed processing time, while jobs from terminal B experience an exponentially distributed processing time. Further, terminal B generates higher priority jobs, which preempt jobs coming from terminal A. Place p_1 in Figure 2a represents the terminal A in the thinking phase and transition t_1 models the exponentially distributed submission time. Place p_2 indicates that job A is being executed, and transition t_2 represents the random execution time. In a similar fashion, transition t_3 models the generation of higher priority jobs from terminal B (place p_3). A token in place p_4 represents a type_B job being processed. Transition t_4 is the processing time of jobs submitted by terminal B. To capture the fact that type_B jobs have higher priority than type_A jobs, we introduce an inhibitor arc from place p_4 to transition t_2 which interrupts the execution of any type_A job until the execution of the type_B job is completed. The firing time of transitions t_1, t_3 and t_4 are assumed to be exponentially distributed, while the service time modeled by transition t_2 is assumed to be generally distributed. The service policy of type_A jobs is preemptive resume which means that an interrupted job is resumed from the point it was interrupted when the processor completes the higher priority job. To capture this behaviour t_2 has an associated age (*prs*) memory policy. Note that the preemption policy assumed for transition t_4 is completely ininfluent for the overall behaviour of the system as t_4 will complete its activity once enabled.

This model was solved by the method based on the Markov Regenerative Theory [5] by assuming the following values. The firing time of transition t_1 and t_3 are exponentially

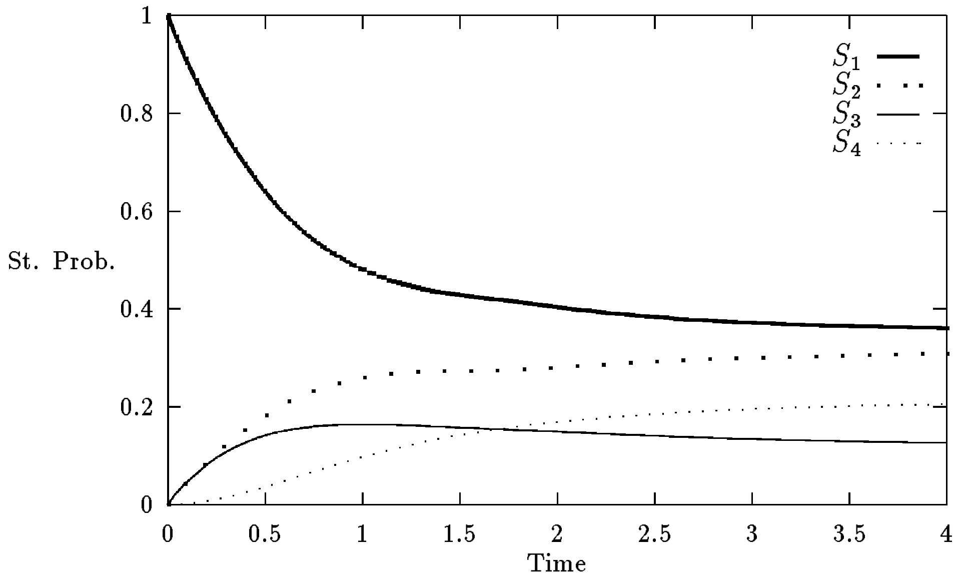

Figure 3: Transient behavior of state probabilities of the processor system

distributed with parameter $\lambda_1 = \lambda_3 = 0.5$. The firing time of transition t_2 is assumed to be uniformly distributed between 0.5 and 1.5 with a *prs* service discipline. The firing time of transition t_4 is exponentially distributed with parameter $\lambda_4 = 1$. The associated reachability graph is shown in Figure 2b.

5 Conclusion

The paper has surveyed the specification and analysis of stochastic models with non-exponential timing through Stochastic Petri Nets. Different analysis approaches proposed in the literature for the numerical evaluation of *MRSPN* models are summarized. An application example, the performance analysis of a complex processor system, is discussed.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, SE-15:832–846, 1989.

[2] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *Lecture Notes in Computer Science*, volume 266, pages 132–145. Springer Verlag, 1987.

[3] A. Bobbio, V.G. Kulkarni, A. Puliafito, M. Telek, and K. Trivedi. Preemptive repeat identical transitions in Markov Regenerative Stochastic Petri Nets. In *6-th International Conference on Petri Nets and Performance Models - PNPM95*, pages 113–122. IEEE Computer Society Press, 1995.

[4] A. Bobbio and M. Telek. Combined preemption policies in MRSPN. In Ravi Mittal et al, editor, *Fault-tolerant systems and software*, pages 92–98. Narosa Publishing House, New Delhi, India, 1995.

[5] A. Bobbio and M. Telek. Markov regenerative SPN with non-overlapping activity cycles. In *International Computer Performance and Dependability Symposium - IPDS95*, pages 124–133. IEEE CS Press, 1995.

[6] Hoon Choi, V.G. Kulkarni, and K. Trivedi. Markov regenerative stochastic Petri nets. *Performance Evaluation*, 20:337–357, 1994.

[7] G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Transactions on Software Engineering*, 20:506–515, 1994.

[8] D.R. Cox. The analysis of non-markovian stochastic processes by the inclusion of supplementary variables. *Proceedings of the Cambridge Phylosophical Society*, 51:433–440, 1955.

[9] A. Cumani. Esp - A package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In *Proceedings International Workshop Timed Petri Nets*, pages 144–151, Torino (Italy), 1985. IEEE Computer Society Press no. 674.

[10] R. German. New results for the analysis of deterministic and stochastic Petri nets. In *International Computer Performance and Dependability Symposium - IPDS95*, pages 114–123. IEEE CS Press, 1995.

[11] R. German, D. Logothetis, and K. Trivedi. Transient analysis of Markov Regenerative Stochastic Petri Nets: a comparison of approaches. In *6-th International Conference on Petri Nets and Performance Models - PNPM95*, pages 103–112. IEEE Computer Society, 1995.

[12] P.J. Haas and G.S. Shedler. Regenerative stochastic Petri nets. *Performance Evaluation*, 6:189–204, 1986.

[13] B.R. Haverkort and K. Trivedi. Specification techniques for Markov Reward Models. *Discrete Event Dynamic Systems: Theory and Applications*, 3:219–247, 1993.

[14] V.G. Kulkarni. *Modeling and Analysis of Stochastic Systems.* Chapman Hall, 1995.

[15] C. Lindemann. DSPNexpress: a software package for the efficient solution of deterministic and stochastic Petri nets. *Performance Evaluation*, 22:3–21, 1995.

[16] M.F. Neuts. *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, 1981.

[17] M. Telek, A. Bobbio, L. Jereb, A. Puliafito, and K. Trivedi. Steady state analysis of Markov regenerative SPN with age memory policy. In H. Beilner and F. Bause, editors, *8-th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation, Lecture Notes in Computer Science*, volume 977, pages 165–179. Springer Verlag, 1995.