Memory Efficient Analysis for a Class of Large Structured Markov Chains

[Work in Progress]

Paolo Ballarini
Centre for Computational and Systems Biology
Università di Trento
Trento, Italy
ballarini@cosbi.eu

András Horváth
Dipartimento di Informatica
Università di Torino
Torino, Italy
horvath@di.unito.it

ABSTRACT

We consider a class of Markov chains modelling concurrent activities with phase-type (PH) distributed duration. Specifically we focus on systems whose dynamics is characterised by the parallel execution of a number of *active* jobs and such that the termination of an active job causes the remaining active jobs to be restarted from scratch (preemptive restart policy) or deactivated and can lead to the activation of new jobs. In such a framework, the state-space can be partitioned into disjoint sets (called macrostates). Each macrostate represents the parallel execution of the (currently) active jobs and a transition to another macrostate corresponds to termination of one of the active job. Models of this kind are subject to the phenomenon of state space explosion, to an extent which is proportional both to the level of concurrency (i.e., the number of active jobs in a macrostate) and to the dimension of the phase-type timing (i.e., the number of phases used to model jobs' execution). Although techniques exist that allow for handling the infinitesimal generator matrix of such models in a memory efficient manner (e.g., in terms of Kronecker expressions), classical Markovian analysis remains hindered by the memory requirements for storing the vector of the steady-state or transient probabilities.

In this paper we show that a Markov chain satisfying the above assumptions can be analysed by calculations performed on the (small) matrices describing the durations of the jobs without the explicit storage of the (large) vectors of transient of steady state probabilities.

1. INTRODUCTION

Model based, stochastic performance evaluation of distributed systems is usually based on a high level description. Modelling formalisms for the descriptions of the model under study are, for example, stochastic Petri nets [1] or stochastic process algebra [16, 4]. In order to obtain a tractable stochastic model, it is usual to assume that all durations of the system are exponentially distributed. In this case the underlying model is a continuous time Markov chain (CTMC). In many situations, however, the exponential distribution is not a satisfactory approximation of the durations of the real system. In these cases it is possible to make the model more realistic by approximating the actual durations by phase type distributions [17]. For surveys on fitting methods the reader is referred to [13, 12, 3].

It is clear that if many simultaneously active events of the system are modeled by phase type distributions, then the state space of the model explodes. The structure of the CTMC, however, allows for very efficient storage of the state space and the infinitesimal generator through expressions in Kronecker algebra and/or decision diagrams techniques [18, 19, 15]. Solution methods based on these techniques are also available [11, 8, 9, 6]. There remains, however, the problem of storing the vector containing the transient or steady state probabilities of the states. To overcome this problem, in [5] and [10] approximate stationary measures are computed based on aggregation while [7] proposes a compact Kronecker representation for the vector which leads to an approximate solution.

In this paper we consider a class of structured Markov chains. We assume that the state space of the model can be partitioned into macrostates in such a way that

- inside a macrostate the process is described by the parallel execution of independent tasks,
- transition from a macrostate to another happens when a tasks finishes its activity and during the transition from a macrostate to another active tasks get either deactivated or restarted and new jobs can become active.

For what concerns the infinitesimal generator, the description of the process inside a macrostate is given by the Kronecker sum of small matrices while the transition between two macrostates is given by Kronecker products of small matrices. Petri nets with PH distributed firing times [19] and process algebras with PH distributed event durations give rise to a Markov chain of this kind.

In [2] we showed that measures connected to the probability of passing through a given series of macrostates for such models can be calculated by Laplace transform techniques in a memory efficient manner.

In this paper we extend such result to the calculation of

SMCTOOLS 2009, October 19, Pisa, Italy

DOI 10.4108/ICST.VALUETOOLS2009.7424

the steady-state and transient distributions for the class of structured Markov chain models obtained by dropping the *preemptive resume policy* from the one considered in [2]. The remainder of paper is organised as follows. In Section 2 the considered model class is described. In Section 3 we provide the details of the calculations. Some conclusive remarks are given in Section 4 together with few notes about future developments of this on going work.

2. CONSIDERED MODEL

The considered model consists of N macrostates and describes the interaction of A activities (also called jobs or tasks) denoted by $a_i, 1 \le i \le A$.

The duration of the activities are described by PH distributions. PH distributions are given by the distribution of time to absorption in a Markov chain [17]. The number of phases, the row vector of the initial probabilities and the infinitesimal generator of the PH distribution associated to activity a_i are denoted by n_i, $\mathbf{b}_i$ and $\mathbf{T}_i$, respectively. Further, the column vector containing the rates from transient states to the absorbing one is denoted by $\mathbf{f}_i$, i.e., $\mathbf{f}_i = -\mathbf{T}_i\mathbf{e}$ where $\mathbf{e}$ is the column vector of ones. In the following we call $\mathbf{f}_i$ the finishing vector of job a_i. We assume that in case of preemption of an activity, the amount of work already done is lost. Consequently, in a given macrostate a given activity is either active, i.e., it is under execution, or inactive, i.e., it is not under execution. The set of active and inactive activities in macrostate i is denoted by A_i^A and A_i^I, respectively. Change of macrostate occurs when the execution of an activity ends. When in macrostate j the execution of active job $a_i \in A_j^A$ ends then the next macrostate is macrostate k with probability $p_i^{(j,k)}$.

In order to describe the Markov chain underlying the process we follow [19]. The infinitesimal generator of the Markov chain has the block structure shown in Table 1. The blocks in the diagonal are given by a Kronecker sum of matrices while those off-diagonal by a Kronecker product of matrices. A given block of $\mathbf{Q}$ will be denoted as $\mathbf{Q}^{(i,j)}, 1 \le i, j \le N$.

A block in the diagonal describes the activity (i.e., the transitions) of concurrent active jobs inside a macrostate:

$$\mathbf{Q}_k^{(j,j)} = \begin{cases} \mathbf{T}_k & \text{if } a_k \in A_j^A \\ 0 & \text{if } a_k \in A_j^I \end{cases} \tag{1}$$

Cases in (1) are:

- if a job is active, it evolves according to its infinitesimal generator matrix;
- an inactive job does not contribute to the description of the macrostate (i.e., its contribution to the Kronecker sum is a scalar 0).

An off-diagonal block describes transitions from a macrostate to another. It is built as a sum of Kronecker products of matrices, $\mathbf{Q}_{k,l}^{(i,j)}$.

Matrix $\mathbf{Q}_{k,l}^{(i,j)}$ describes what happens to activity a_k when the process arrives to macrostate j from macrostate i as a consequence of termination of activity a_l. These matrices are

$$\mathbf{Q}_{k,l}^{(i,j)} = \begin{cases} \mathbf{f}_k & \text{if } a_k \in A_i^A \text{ and } k = l \text{ and } a_k \in A_j^I \\ \mathbf{f}_k\mathbf{b}_k & \text{if } a_k \in A_i^A \text{ and } k = l \text{ and } a_k \in A_j^A \\ \mathbf{1}_{k,1} & \text{if } a_k \in A_i^A \text{ and } k \ne l \text{ and } a_k \in A_j^I \\ \mathbf{b}_k & \text{if } a_k \in A_i^I \text{ and } a_k \in A_j^A \\ 1 & \text{if } a_k \in A_i^I \text{ and } a_k \in A_j^I \end{cases} \tag{2}$$

with $i \ne j$ and where $\mathbf{1}_{n,m}$ denotes a matrix of 1s of size $n \times n$. Cases in (2) are

- if a job (i.e., job k) is active in macrostate i, it terminates and it is not active in the resulting macrostate j, then its contribution is given by its finishing vector;
- if a job (i.e., job k) is active in macrostate i, it terminates and it is active again in the resulting macrostate j, then its contribution is given by the product of its finishing vector and its initial probability vector;
- if a job (i.e., job k) is active in macrostate i, and it does not terminate (i.e., $k \ne l$) and it is inactive in the resulting macrostate j, then its contribution is given by a vector of 1s, i.e., we can simply "forget" the phase of this job;
- if a job is inactive in macrostate i and it is active in the resulting macrostate j, then its starting phase is determined according to its initial probability vector;
- if a job is inactive in macrostate i and it is inactive in the resulting macrostate j, then it does not contribute to the block.

Clearly, the class of models we consider is very special. Its analysis however is not straightforward and it is interesting in practice to model, for example, machines with multiple failure modes and PH distributed up and down times.

3. ANALYSIS

In this section we consider the calculation of the quantity $(s\mathbf{I} - \mathbf{Q})^{-1}$, the Laplace-transform of $\exp(\mathbf{Q}t)$. This quantity then can be used to calculate transient probabilities by numerical inversion techniques or to calculate steady state probabilities by applying the well-known limit theorem of the Laplace transform, $f(\infty) = \lim_{s\to 0} sf^*(s)$. First we recall results that are necessary in the sequel.

According to the Sherman-Morrison formula, given $\mathbf{G}$ and $\mathbf{G}+\mathbf{E}$ non-singular matrices where $\mathbf{E}$ is of rank one, we have

$$(\mathbf{G} + \mathbf{E})^{-1} = \mathbf{G}^{-1} - \frac{1}{1+g}\mathbf{G}^{-1}\mathbf{E}\mathbf{G}^{-1} \tag{3}$$

where g is the trace of $\mathbf{E}\mathbf{G}^{-1}$. The above relation is given exactly in the form of (3) in [14].

It is well-known that inversion of a matrix can be carried out in blocks

$$\begin{vmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{vmatrix}^{-1} = \tag{4}$$

$$\begin{vmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{vmatrix}$$

In the following, we will apply the above block inverse to cases in which $\mathbf{B}$ is of rank one. Consequently, the matrices $\mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ and $\mathbf{C}\mathbf{A}^{-1}\mathbf{B}$ are of rank one as well and the

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7424
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7424

$$\mathbf{Q} = \begin{vmatrix} \oplus_{i=1}^{A}\mathbf{Q}_i^{(1,1)} & \sum_{j:a_j\in A_1^A} p_j^{(1,2)} \otimes_{i=1}^{A} \mathbf{Q}_{i,j}^{(1,2)} & \dots & \sum_{j:a_j\in A_1^A} p_j^{(1,N)} \otimes_{i=1}^{A} \mathbf{Q}_{i,j}^{(1,N)} \\ \sum_{j:a_j\in A_2^A} p_j^{(2,1)} \otimes_{i=1}^{A} \mathbf{Q}_{i,j}^{(2,1)} & \oplus_{i=1}^{A}\mathbf{Q}_i^{(2,2)} & \dots & \sum_{j:a_j\in A_2^A} p_j^{(2,N)} \otimes_{i=1}^{A} \mathbf{Q}_{i,j}^{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j:a_j\in A_N^A} p_j^{(N,1)} \otimes_{i=1}^{A} \mathbf{Q}_{i,j}^{(N,1)} & \sum_{j:a_j\in A_N^A} p_j^{(N,2)} \otimes_{i=1}^{A} \mathbf{Q}_{i,j}^{(N,2)} & \dots & \oplus_{i=1}^{A}\mathbf{Q}_i^{(N,N)} \end{vmatrix}$$

Table 1: Block structure of the infinitesimal generator of the considered class of Markov chains

inversion formula given in (3) can be applied to write up $(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}$ and $(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$.

The matrix $(s\mathbf{I}-\mathbf{Q})^{-1}$ can be calculated by the iterative application of (4) and (3). We illustrate the procedure with three macrostates only but the extension to a general number of macrostates is straightforward and can be performed easily by symbolic math packages. With three macrostates the matrix $s\mathbf{I}-\mathbf{Q}$ is of the form

$$\begin{vmatrix} s\mathbf{I}-\mathbf{Q}^{(1,1)} & -\mathbf{Q}^{(1,2)} & -\mathbf{Q}^{(1,3)} \\ -\mathbf{Q}^{(2,1)} & s\mathbf{I}-\mathbf{Q}^{(2,2)} & -\mathbf{Q}^{(2,3)} \\ -\mathbf{Q}^{(3,1)} & -\mathbf{Q}^{(3,2)} & s\mathbf{I}-\mathbf{Q}^{(3,3)} \end{vmatrix}$$

and we will denote its blocks by $\mathbf{R}_{i,j}, 1 \le i,j \le 3$. First we apply (4) with $\mathbf{A}=\mathbf{R}_{1,1}, \mathbf{B}=\mathbf{R}_{1,2}, \mathbf{C}=\mathbf{R}_{2,1}$ and $\mathbf{D}=\mathbf{R}_{2,2}$. The block $\mathbf{B}$ describes the way the process enters the second macrostate and on entering this macrostate all activities are restarted. For this reason all rows of $\mathbf{B}$ are multiples of each other and, consequently, the rank of $\mathbf{B}$ is 1. Accordingly, we calculate $(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}$ and $(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$ according to (3). The result is given in Table 2 where $c_1 = 1/(1-d_1)$ with d_1 being the trace of

$$\mathbf{R}_{1,2}\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{R}_{1,1}^{-1}$$

and we will denote by $\mathbf{S}_{i,j}, 1 \le i,j \le 2$ the blocks of the resulting matrix for further use.

The next step is to apply (4) again with

$$\mathbf{A} = \begin{vmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{1,2} \\ \mathbf{R}_{2,1} & \mathbf{R}_{2,2} \end{vmatrix}, \ \mathbf{B} = \begin{vmatrix} \mathbf{R}_{1,3} \\ \mathbf{R}_{2,3} \end{vmatrix},$$
$$\mathbf{C} = \begin{vmatrix} \mathbf{R}_{3,1} & \mathbf{R}_{3,2} \end{vmatrix}, \ \mathbf{D} = \begin{vmatrix} \mathbf{R}_{3,3} \end{vmatrix}$$

considering the inverse of $\mathbf{A}$ as given in Table 2. In this case as well, since all jobs are restarted on entry into the third macrostate, $\mathbf{B}$ is of rank one and we can apply (3) to calculate the necessary inverses. The inverse of $s\mathbf{I}-\mathbf{Q}$ in case of three macrostates is given in Table 3 where

$$\mathbf{F}_1 = \mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{S}_{1,1}, \ \mathbf{F}_2 = \mathbf{R}_{3,3}^{-1}\mathbf{R}_{3,1}\mathbf{S}_{1,1},$$
$$\mathbf{F}_3 = \mathbf{R}_{3,3}^{-1}\mathbf{R}_{3,2}\mathbf{S}_{2,1}, \ \mathbf{F}_4 = \mathbf{R}_{3,3}^{-1}\mathbf{R}_{3,1}\mathbf{S}_{1,2},$$
$$\mathbf{F}_5 = \mathbf{R}_{3,3}^{-1}\mathbf{R}_{3,2}\mathbf{S}_{2,2}, \ \mathbf{F}_6 = \mathbf{R}_{3,3}^{-1}\mathbf{R}_{3,2}\mathbf{F}_1,$$
$$\mathbf{F}_7 = \mathbf{S}_{1,1}\mathbf{R}_{1,3} + \mathbf{S}_{1,2}\mathbf{R}_{2,3}$$

and $c_2 = 1/(1-d_2-d_3-d_4-d_5), c_3 = 1/(1-d_4-d_5-d_6), c_4 = 1/(1-d_2-d_3-d_5+d_6)$ with d_2, d_3, d_4, d_5 and d_6 being, respectively, the traces of the matrices

$$\mathbf{R}_{1,3}\mathbf{F}_2, \ \mathbf{R}_{1,3}\mathbf{F}_3 \ \mathbf{R}_{2,3}\mathbf{F}_4, \ \mathbf{R}_{2,3}\mathbf{F}_5,$$
$$\mathbf{R}_{1,3}\mathbf{R}_{3,3}^{-1}\mathbf{R}_{3,2}\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{S}_{1,1} \ .$$

The calculation of the inverse $s\mathbf{I}-\mathbf{Q}$ results in complicated expressions but they can be easily obtained by symbolic math tools like Mathematica or Maple.

All that remains is to calculate the inverses

$$\mathbf{R}_{j,j}^{-1} = \left(s\mathbf{I}-\mathbf{Q}^{(j,j)}\right)^{-1}$$

where the matrix $\mathbf{Q}^{(j,j)}$ is given as the Kronecker sum of the matrices that describe the tasks in macrostate j, i.e., $\mathbf{Q}^{(j,j)} = \oplus_{i=1}^{A}\mathbf{Q}_i^{(j,j)}$. The following theorem, provided in [2], gives the possibility of performing the calculations based on the matrices $\mathbf{Q}_i^{(j,j)}$.

Theorem 1. *Given the Jordan normal form of the matrices* $\mathbf{Q}_i^{(j,j)}, 1 \le j \le N, 1 \le i \le A$ *such that*

$$\mathbf{Q}_i^{(j,j)} = \mathbf{V}_{j,i}\mathbf{J}_{j,i}\mathbf{V}_{j,i}^{-1}$$

we have

$$\left(s\mathbf{I}-\mathbf{Q}^{(j,j)}\right)^{-1} = \left(s\mathbf{I}-\oplus_{i=1}^{A}\mathbf{Q}_i^{(j,j)}\right)^{-1} =$$
$$\left(\otimes_{i=1}^{A}\mathbf{V}_{j,i}\right)\left(s\mathbf{I}-\oplus_{i=1}^{A}\mathbf{J}_{j,i}\right)^{-1}\left(\otimes_{i=1}^{A}\mathbf{V}_{j,i}^{-1}\right) \ . \quad (5)$$

In many applications special classes of PH distributions are used. In [2] the symbolic Jordan normal form for Erlang distributions and for acyclic PH distributions are provided. If the applied PH distributions do not fall into one of the above categories then the Jordan decomposition can be performed by numerical calculations.

For what concerns numerical calculations, the basic step of the approach proposed in this paper is exactly the one that was already analysed in [2]. In [2] also the numerical problems associated with the Jordan decomposition are discussed.

4. CONCLUSIONS

In this paper we proposed an analysis approach for structured Markov chains which can be carried out based on calculations on the building blocks of the model. Future activity must include implementation and extension to the more general case when not all activities are restarted on entry to a macrostate. Also different matrix decompositions will be considered in order to avoid the numerical problems associated with the Jordan decomposition.

5. REFERENCES

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* J. Wiley, 1995.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7424
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7424

$$\left|\begin{array}{cc}\mathbf{R}_{1,1} & \mathbf{R}_{1,2}\\ \mathbf{R}_{2,1} & \mathbf{R}_{2,2}\end{array}\right|^{-1} = \left|\begin{array}{cc}\mathbf{S}_{1,1} & \mathbf{S}_{1,2}\\ \mathbf{S}_{2,1} & \mathbf{S}_{2,2}\end{array}\right| =$$

$$\left|\begin{array}{cc}\mathbf{R}_{1,1}^{-1} + c_1\mathbf{R}_{1,1}^{-1}\mathbf{R}_{1,2}\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{R}_{1,1}^{-1} & -\mathbf{R}_{1,1}^{-1}\mathbf{R}_{1,2}\left(\mathbf{R}_{2,2}^{-1} + c_1\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{R}_{1,1}^{-1}\mathbf{R}_{1,2}\mathbf{R}_{2,2}^{-1}\right)\\ -\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\left(\mathbf{R}_{1,1}^{-1} + c_1\mathbf{R}_{1,1}^{-1}\mathbf{R}_{1,2}\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{R}_{1,1}^{-1}\right) & \mathbf{R}_{2,2}^{-1} + c_1\mathbf{R}_{2,2}^{-1}\mathbf{R}_{2,1}\mathbf{R}_{1,1}^{-1}\mathbf{R}_{1,2}\mathbf{R}_{2,2}^{-1}\end{array}\right|$$

Table 2: Case of three macrostates: inverse of submatrix of $s\mathbf{I} - \mathbf{Q}$ formed by $\mathbf{R}_{1,1}, \mathbf{R}_{1,2}, \mathbf{R}_{2,1}$ and $\mathbf{R}_{2,2}$

$$\left|\begin{array}{ccc}\mathbf{S}_{1,1} + c_2\mathbf{F}_7(\mathbf{F}_2 + \mathbf{F}_3) & \mathbf{S}_{1,2} + c_3\mathbf{F}_7(\mathbf{F}_4 + \mathbf{F}_5) & \begin{array}{c}\mathbf{F}_7(c_4((\mathbf{F}_6 - \mathbf{F}_2)\mathbf{R}_{1,3}-\\ (\mathbf{F}_4 + \mathbf{F}_5)\mathbf{R}_{2,3})\mathbf{R}_{3,3}^{-1} - \mathbf{R}_{3,3}^{-1})\end{array}\\ -\mathbf{F}_1 - c_3\mathbf{F}_8(\mathbf{F}_2 - \mathbf{F}_6) & \mathbf{S}_{2,2} - c_3\mathbf{F}_8(\mathbf{F}_4 + \mathbf{F}_5) & \begin{array}{c}\mathbf{F}_8(-c_4((\mathbf{F}_6 - \mathbf{F}_2)\mathbf{R}_{1,3}-\\ (\mathbf{F}_4 + \mathbf{F}_5)\mathbf{R}_{2,3})\mathbf{R}_{3,3}^{-1} - \mathbf{R}_{3,3}^{-1})\end{array}\\ \begin{array}{c}\mathbf{R}_{3,3}^{-1}(\mathbf{R}_{3,2}(\mathbf{F}_1 + c_3\mathbf{F}_8(\mathbf{F}_2 - \mathbf{F}_6))-\\ \mathbf{R}_{3,1}(\mathbf{S}_{1,1} + c_2\mathbf{F}_7(\mathbf{F}_2 + \mathbf{F}_3)))\end{array} & \begin{array}{c}-\mathbf{R}_{3,3}^{-1}(\mathbf{R}_{3,1}(\mathbf{S}_{1,2} + c_3\mathbf{F}_7(\mathbf{F}_4 + \mathbf{F}_5))+\\ \mathbf{R}_{3,2}(\mathbf{S}_{2,2} - c_3\mathbf{F}_8(\mathbf{F}_4 + \mathbf{F}_5)))\end{array} & \begin{array}{c}c_4((\mathbf{F}_2 - \mathbf{F}_6)\mathbf{R}_{1,3}+\\ (\mathbf{F}_4 + \mathbf{F}_5)\mathbf{R}_{2,3})\mathbf{R}_{3,3}^{-1} - \mathbf{R}_{3,3}^{-1}\end{array}\end{array}\right|$$

Table 3: Case of three macrostates: inverse of $s\mathbf{I} - \mathbf{Q}$

[2] P. Ballarini and A. Horváth. Memory efficient calculation of path probabilities in large structured markov chains. In *Proc. of the 5th International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 157–166, St. Malo, France, September 2008. IEEE Computer Society.

[3] L. Bodrog, A. Heindl, A. Horváth, G. Horváth, and M. Telek. Current results and open questions on PH and MAP characterization. In *Dagstuhl Seminar N.07461 on Numerical Methods for Structured Markov Chains*, 2008.

[4] P. Buchholz. On a Markovian process algebra. Technical report, Technical Report 500, Informatik IV, University of Dortmund, 1994.

[5] P. Buchholz. Iterative decomposition and aggregation of labelled GSPNs. In *Application and Theory of Petri Nets 1998: 19th International Conference, ICATPN'98*, 1998.

[6] P. Buchholz. Structured analysis approaches for large Markov chains. *Applied Numerical*, 31(4):375–404, 1999.

[7] P. Buchholz. An adaptive decomposition approach for the analysis of stochastic Petri nets. *Performance Evaluation*, 56(1–4):23–52, 2004.

[8] J. Campos, M. Silva, and S. Donatelli. Structured solution of asynchronously communicating stochastic modules. *IEEE Transactions on Software Engineering*, 25(2):147–165, 1999.

[9] G. Ciardo and A. S. Miner. A data structure for the efficient kronecker solution of gspns. In *Proc. of the 8th Int. Work. on Petri Nets and Performance Models*, pages 22–31, 1999.

[10] G. Ciardo, A. S. Miner, and S. Donatelli. Using the exact state space of a model to compute approximate stationary measures. In *Proc. ACM Sigmetrics*, 2000.

[11] P. Fernandes, B. Plateau, and W. J. Stewart. Efficient descriptor-vector multiplication in stochastic automata networks. *Journal of the ACM*, 45(3):381–414, 1998.

[12] A. Horváth and M. Telek. Markovian modeling of real data traffic: Heuristic phase type and map fitting of heavy tailed and fractal like samples. In *Tutorial of the IFIP WG 7.3 International Symposium on Computer Performance Modeling, Measurement and Evaluation (PERFORMANCE 2002)*, volume 2459 of *Lecture Notes in Computer Science*, Rome, Italy, Sept 2002.

[13] A. Lang and J. L. Arthur. Parameter approximation for phase-type distributions. In S. R. Chakravarty and A. S. Alfa, editors, *Matrix-analytic methods in stochastic models*, Lecture notes in pure and applied mathematics, pages 151–206. Marcel Dekker, Inc., 1996.

[14] K. S. Miller. On the inverse of the sum of matrices. *Mathematics Magazine*, 54(2):67–72, 1981.

[15] A. S. Miner and G. Ciardo. Efficient reachability set generation and storage using decision diagrams. In *Proceedings of the 20th Int. Conf. on Applications and Theory of Petri Nets*, volume 1639 of *Lecture Notes in Computer Science*, pages 6–25, Williamsburg, VA, USA, 1999.

[16] M. R. N. Goetz, U. Herzog. Multiprocessor and distributed system design: The integration of functional specication and performance analysis using stochastic process algebras. In *Proceedings of Performance Evaluation of Computer and Communication Systems*, volume 729 of *Lecture Notes in Computer Science*, pages 121–146, 1993.

[17] M. Neuts. Probability distributions of phase type. In *Liber Amicorum Prof. Emeritus H. Florin*, pages 173–206. University of Louvain, 1975.

[18] B. Plateau. On the stochastic structure of parallelism and synchronisation models for distributed. *Performance Evaluation Review*, 13:142–154, 1985.

[19] M. Scarpa and A. Bobbio. Kronecker representation of Stochastic Petri nets with discrete PH distributions. In *International Computer Performance and Dependability Symposium - IPDS98*, pages 52–61. IEEE CS Press, 1998.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7424
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7424