Structured characterization of the Markov chain of phase-type SPN

Susanna Donatelli[1]*, Serge Haddad[2], and Patrice Moreaux[2,3]

[1] Università di Torino, Dipartimento di Informatica susi@di.unito.it
[2] LAMSADE {haddad, moreaux}@lamsade.dauphine.fr
[3] Université de Reims Champagne-Ardenne

Abstract. This paper presents a characterization of the Markovian state space of a Stochastic Petri Nets with phase-type distribution transitions as a union of Cartesian products of a set of "components" of the net. The method uses an abstract view of the net based on the vectors of enabling degrees of phase-type transitions, as well as on the sets of "interrupted clients". Following the decomposition used for the state space characterization, a tensor algebra expression for the infinitesimal generator (actually for its rate matrix) is given, that allows the steady state probability to be computed directly from a set of matrices of the size of the components, without the need of storing the whole infinitesimal generator.

1 Introduction

Since the introduction of Stochastic Petri Nets (SPN) [17]), the need for more general distributions of firing time have been recognized as necessary to adequately model many real life phenomena. Two general directions have been taken in this area: the first one [1, 10, 2] introduces general distributions of firing time with specific conditions which allows one to extract "subordinated" Markov chains and compute steady state and transient probabilities of the states. The second approach [17, 11, 9] introduces phase-type (PH) distribution firing time without further restrictions, and computes the resulting embedded Continuous Time Markov Chain (CTMC). Two problems then arise: to precisely define the stochastic semantics of phase-type (PH-)transitions and to cope with the state space problem since we are faced with a "bi-dimensional" complexity: exponential with respect to the size of the net, and that can be exponential under specific hypotheses with respect to the number of stages of the PH distributions.

Previous works on SPN with PH-transitions (PH-SPN for short) have taken into account the stochastic semantics problem either at the net level [9], or at the underlying CTMC level directly [11].

For what concerns the net level approach, the works reported in [17, 9] *alter* the initial SPN with PH-transitions: each phase type (including special cases like

* This work has been developed within the project HCM CT94-0452 (MATCH) of the European Union. At time of writing S. Donatelli was visiting professor at Lamsade, at the University of Paris-Dauphine.

Erlang, Coxian) transition is replaced in the net with a sub-net which translates its stochastic semantics. This expansion may be automated and integrated in a tool. However only the single server policy is studied and, moreover, each PH-transition must have only one input and one output place in [9].

The second approach [11] expands the embedded Markov chain, and the net is not modified. The PH distributions are taken into account during the reachability graph (RG) computation. This method allows the management of any combination of the three elements of the execution policy (that we shall define later), and may be also integrated in a tool.

It is now well-known [19, 4, 12, 14] that tensor expressions applied to queuing networks and SPN solution can provide substantial memory savings which enable management of Markov chains with large state space. In fact, it was recognized (for example in [16]) that the main memory bottleneck with tensor methods is the size of the steady state probabilities vector, and not of the infinitesimal generator.

In a first attempt to reduce the complexity of the *resolution* of the Markov chain derived from an SPN with PH-distributions, the work in [15] presents a method based on structural decomposition of the net, leading to a tensor expression of the generator of the chain. Such a decomposition builds a superset of the state space of the chain as a *Cartesian product* of smaller state spaces and classify state transitions as local, that is to say changing only one component of the state, or global, changing simultaneously several components of the state.

A serious drawback of the work in [15] is in the number of elements of the Cartesian product that are not legal states of the system. This may lower the efficacy of the method since a lot of space and time can be lost in those states that are part of the Cartesian product. This problem is common also to other approaches using tensor algebra.

In this paper we show instead how tensor algebra can be used to *exactly* characterize the state space of a PH-SPN, in a way that does not depend on the structure of the net, by following the approach based on high level state information, introduced in the context of marked graphs [6], and used for hierarchical GSPN [3], and, recently, to the asynchronous composition of components[8, 7].

Using this characterization, we derive a tensor expression of the rate matrix of the Markov chain of the net which allows the computation of the steady state probabilities coping with the growth of the Markovian state space induced by the PH-transitions.

The method is presented in the context of exponentially distributed, transition timed, Petri nets, without immediate transitions and inhibitor arcs and assumes that the net is small enough for the tangible reachability set TRS, and for the tangible reachability graph TRG, to be built and stored in memory.

The paper is organized as follows: Section 2 introduces the definition of PH distributions, the semantics of PH-transitions, the definition of PH-SPN, and discusses the definition of Markovian state of an SPN which is at the root of the construction of the Markovian state space presented in Section 3 for single server transitions. Section 4 introduces the tensor expressions for the corresponding rate matrix. Section 5 outlines the results in the more complicated case of PH-transitions with multiple servers semantics. The method presented is summarized and evaluated in Section 6 that concludes the paper. The reader will find in [13] an extended version of the paper with detailed examples and proofs.

2 Phase-type SPN

PH distributions have been defined [18] as the time until absorption of a CTMC with $s+1$ states and a single absorbing state. A PH distribution can also be seen as a network of exponential servers, the initial distribution of the chain being $(c_{O,j})_{1\leq j\leq s}$ and the absorbing state being the state 0. Leaving a stage i, a client may enter stage j with probability $C_{i,j}$ or enter the absorbing state with probability $C_{i,0}$ ($\sum_{j=0}^{j=s} C_{i,j} = 1$).

We firstly study stochastic semantics of PH-transitions and we give the definition of PH-SPN; then we discuss how to define a state of the system that is consistent with the given specifications.

2.1 Stochastic semantics of PH-transitions

In order to provide a full definition of PH-SPN, a number of additional specifications should be added to each non-exponential transitions.

• *Memory policy:* the memory policy specifies what happens to the time associated to transition when a change of state occurs.

With the *Enabling Memory* policy, the elapsed time is lost only if t is not enabled in the new marking. Transitions with enabling memory therefore do not loose the past work, as long as they are enabled. Time-out transition usually have this memory policy.

With *Age Memory* policy the elapsed time is always kept (of course unless it is transition t itself that causes the change of state) whatever the evolution of the system: the next time the transition will be enabled, the remaining firing time will be the *residual time* from the last disabling and not a new sampling. Time-sharing service may be modelled by age memory policies.

In this paper we do not consider *resampling policy* which realizes a conflict at the temporal specification level, that may not have a counterpart at the structure level.

The memory policy of transition t is denoted by $mp(t)$, and the possible values are E, for enabling, or A, for age.

The enabling memory policy of PH-transitions needs a precise definition, giving how its internal state is modified when another transition fires. Since a firing is considered as an "atomic" action, in this paper we use the following interpretation of enabling memory: the memory of a transition t_h, enabling memory, is reset when either in the new state the transition is disabled, or in the new state the transition has kept its enabling, but it is the transition t_h itself that has caused the change of state.

• *Service semantics:* the service semantics specifies how many clients can be served in parallel by a transition. The use of the term "client" is imported from queuing networks, and we use it in the paper as a synonymous of active server.

The service semantics is defined around the concept of *enabling firing degree* $\mathrm{edf}(\mathrm{t}, \mathbf{m})$ of t in $\mathbf{m}$ which gives the number of time the transition t may be fired consecutively from $\mathbf{m}$:

$$\mathrm{edf}(\mathrm{t}, \mathbf{m}) = k \qquad \text{iff } \begin{cases} \forall p \in {}^\bullet t,\ \mathbf{m}(p) \geq k \cdot \mathbf{Pre}(p,t) \\ \exists p \in {}^\bullet t,\ \mathbf{m}(p) < (k+1) \cdot \mathbf{Pre}(p,t) \end{cases}$$

The service policy of transition t is denoted by $sp(t)$. We have the three classical policies: single server (S), multiple (K) and infinite (I) server.

We shall denote by $\mathrm{ed}(\mathrm{t}, \mathbf{m})$ the enabling degree of transitions, *according with the service policy*, that is to say: $\mathrm{ed}(\mathrm{t}, \mathbf{m}) = \min(K, \mathrm{edf}(\mathrm{t}, \mathbf{m}))$, where $K = 1$ for single server, $K = K$ for K-multiple server, and $K = +\infty$ for infinite server.

As usual, the firing rate of an exponential transition t in $\mathbf{m}$ is $\mathrm{ed}(\mathrm{t}, \mathbf{m})) \cdot \mathbf{w}(t, \mathbf{m})$. For single server PH-transition, the firing rate is $\mu_s \cdot C(t)[s, 0]$ if a client in stage s ends its service and no new client enters in service, and $\mu_s \cdot C(t)[s, 0] \cdot C(t)[0, s']$ if another client enters immediately in service in stage s'. The multiple and infinite server cases introduce several complex problems so that we have chosen to present the method first for the single server case, then to generalize the results to the multiple and infinite server cases in Section 5.

• *Clients and interrupt/resume policies:* two other parameters have to be defined for multiple and infinite server policies: the Clients policy (cp) and the interrupt/resume policy (ip). For the single server case, we suppose the interrupt/resume semantics which resumes the interrupted client (if any), when a transition firing increases the enabling degree (from 0 to 1). The study of cp and ip for multiple and infinite server PH-transitions is outlined in Section 5.

The specification of the stochastic semantics of a PH-transition is then $\theta = (s, \mu, C, mp, sp, cp, ip)$, with $s(t)$ the number of stages of t and $\forall 1 \leq i \leq s(t)$:

- $\mu_i(t)$ the rate of the ith stage of t;
- $C(t)[i, j]$ the probabilities of routing a client from the ith stage to the jth $(0 \leq j \leq s(t))$ stage after the end of service of the ith stage. $C[i, 0] = 1 - \sum_{j=1}^{j=s(t)} C(t)[i, j]$ is the probability to leave the service of the transition. $C[0, j]$, the probabilities for an incoming client to enter the system in stage j, with $\sum_{j=1}^{j=s(t)} C(t)[0, j] = 1$ (it is not possible to leave the transition in zero time);
- $mp(t)$ the memory policy of t (E or A), $sp(t)$ the service policy of t (S, K or I), $cp(t)$ the clients policy of t, if needed (O or U), and $ip(t)$ the interrupt policy of t, if needed.

Let us now introduce the definition of PH-SPN.

Definition 1. *A PH-SPN is a tuple* $\mathcal{N} = (P, \mathrm{TX}, \mathrm{TH}, \mathbf{Pre}, \mathbf{Post}, \mathbf{w}, \theta)$*:*

- P *is the set of places;*
- TX *is the set of exponential transitions;*

– *TH is the set of phase-type (PH-)transitions;* $\mathrm{TX} \cap \mathrm{TH} = \emptyset$ *and* $\mathrm{TX} \cup \mathrm{TH} = T$ *is generically called the set of transitions.*
– $\mathbf{Pre}$ *and* $\mathbf{Post}$: $P \times T \to \mathbb{N}$ *are the incidence functions:* $\mathbf{Pre}$ *is the input function, and* $\mathbf{Post}$ *the output function*
– $\mathbf{w} : T \times \mathrm{Bag}(P) \to \mathbb{R}^+$*:* $\mathbf{w}(t, \mathbf{m})$ *is the firing rate of the exponential distribution associated to transition t in marking*[1] $\mathbf{m}$*, if t is exponential, and it is instead equal to 1 for PH-transitions.*
– *θ is a function that associates to each transition $t_h \in \mathrm{TH}$ its specification θ_h.*

The choice of an initial marking $\mathbf{m_0}$, defines a marked PH-SPN also called a PH-SPN *system* $\mathcal{S} = (\mathcal{N}, \mathbf{m_0})$. We denote with TRS and TRG the reachability set and graph.

The definition of a Markovian state of a PH-SPN is therefore an $(H+1)$-tuple,

$$(\mathbf{m}, \mathrm{d}(t_1, \mathbf{m}), \ldots, \mathrm{d}(t_H, \mathbf{m}))$$

where $\mathbf{m}$ is the marking, and $\mathrm{d}(t_h, \mathbf{m})$ is the "descriptor" of phase-type transition t_h in marking $\mathbf{m}$. In the next section we shall define the descriptors in such a way for the state to be Markovian, and we shall see that, for the age memory case, the descriptors depend on information that can be computed only from the reachability graph, so that it may be more precise to write $(\mathbf{m}, \mathrm{d}(t_1, \mathbf{m}, \mathrm{TRG}), \ldots, \mathrm{d}(t_H, \mathbf{m}, \mathrm{TRG}))$. We call MS the set of Markovian states of a PH-SPN.

2.2 Descriptors for single server PH-transitions

When there are PH-transitions the marking process of the net is not Markovian any longer, but it is still possible to build a Markovian state by considering more detailed information [18, 11]. We shall consider the enabling memory and the age memory case separately.

• *Descriptors for enabling memory:* for an enabling memory transition the only relevant information is the stage of the current client. Let $d(t, \mathbf{m})$ (d for short, if t and $\mathbf{m}$ are implicit from the context) denote the descriptor of a PH- transition t. We have

$$d = i \in \{0, 1, \ldots, s(t)\}$$

where i is the index of the stage in which the client is receiving service. $i = 0$ means no client is currently in service.

• *Descriptors for age memory:* for the case of age memory we need the additional information on the interrupted client giving in which stage it has been interrupted. This information can be stored explicitly or implicitly. We can add to the descriptor a number indicating the stage at which the client has been interrupted (explicit info), or we can encode this information into the descriptor already defined (implicit info): since a single server transition has either a client interrupted, or a client in service, but never the two together, then $d(t, \mathbf{m}) = i$

[1] $\mathrm{Bag}(P)$ is the set of multi-sets on P.

means: client in service in stage i if $\mathrm{ed}(\mathrm{t}, \mathbf{m}) > 0$ and client interrupted in stage i if $\mathrm{ed}(\mathrm{t}, \mathbf{m}) = 0$.

In this paper we choose the implicit encoding. Note however, that we need to explicitly encode the descriptor for no client at all (either in service or interrupted) with the value $d(t, \mathbf{m}) = 0$.

3 Characterization of the Markovian state space of a PH-SPN in the single server case

In this section we show the first main result of the paper which gives an exact structured description of the MS of a PH-SPN. The method, presented and used in [6] for marked graphs, in [8] for DSSP systems, and, more recently, in [7] exploits two ideas: the first one is a description of the MS as Cartesian product of K subspaces following the classical tensor based approach [3, 12, 15]. The second one is the introduction of a high level (abstract) description of the MS, used in the asynchronous context in [4, 6–8], "compatible" with the previous subspaces product. This leads to an expression of MS as a disjoint union of Cartesian products with the same high level description. In a formal way, if $\mathcal{A}$ is the set of abstract views $av(s)$ of the Markovian states s, we have:

$$MS = \biguplus_{a \in \mathcal{A}} S_1(a) \times \ldots \times S_K(a)$$

where $S_k(a)$ denotes the subset of states of the kth component such that

$$(\forall\, 1 \leq k \leq K, s_k \in S_k(a)) \Leftrightarrow \begin{cases} s = (s_1, \ldots, s_K) \in MS \\ av(s) = a \end{cases}$$

The existence of an high level description of the system allows to appropriately pre-select the subsets of the states that should enter in the Cartesian product: only states of the different components that share the same high level view of the system are multiplied in the Cartesian product. Without such a high level description, Cartesian products provide only Potential Markovian state Spaces (PMS) with $MS \subset PMS$ and, in the general case, $|PMS| \gg |MS|$.

Since a Markovian state of a PH-SPN is a $H+1$ tuple $(\mathbf{m}, \mathrm{d}(t_1, \mathbf{m}), \ldots \mathrm{d}(t_H, \mathbf{m}))$, we consider MS as a set of $H+1$ components, one component defines the marking of the net and the other H the state of the H phase-type-transitions.

For what concerns the high level view, although we might use the marking $\mathbf{m}$ of a state (any other “high level” information computable from the TRS in linear time and space, may also be a choice), it is not the most efficient choice. Indeed, as we have seen in the previous section, the descriptor of a PH-transition only depends on its enabling degree for enabling memory, and on the enabling degree and on the number of interrupted clients for age memory. It is then suitable to define a coarser high level view exploiting these weaker dependencies. We shall consider the two cases enabling and age separately.

• *Enabling memory case:* from the definition of the descriptors, the set $D_h(\mathbf{m})$ of legal descriptors of a transition t_h for a marking $\mathbf{m}$ only depends on the enabling degree $\mathrm{ed(t_h}, \mathbf{m}) = \mathrm{e}_h$ of t_h in $\mathbf{m}$: $D_h(\mathbf{m}) = D_h(e_h) = \{i : 1 \leq i \leq s(t_h)\}$ if $e_h = 1$ and $D_h(e_h) = \{0\}$ if $e_h = 0$. This key observation leads us to define the equivalence relation ρ over the set of markings in TRS:

$$\mathbf{m} \; \rho \; \mathbf{m}' \quad \text{iff} \quad \forall t_h \in \mathrm{TH} \;\; \mathrm{ed(t_h}, \mathbf{m}) = \mathrm{ed(t_h}, \mathbf{m}')$$

and to take the vector $\mathbf{ed} = (\mathrm{e}_1, \ldots, \mathrm{e}_H)$ of enabling degrees of transitions in TH as the high level description of Markovian states. We use as representative of a class the vector $\mathbf{ed}$ and indicate the set of markings of equivalence class $\mathbf{ed}$ with $[\mathbf{ed}]$ and the set of distinct vectors of enabling degrees of a given TRS with $\mathcal{ED}$. The following proposition gives the structure of MS.

Proposition 1. *If all phase-type transitions of a PH-SPN $\mathcal{S}$ are enabling memory then*

$$\mathrm{MS}(\mathcal{S}) = \biguplus_{\mathbf{ed} \in \mathcal{ED}} [\mathbf{ed}] \times \mathrm{D}_1(\mathrm{e}_1) \times \cdots \times \mathrm{D}_H(\mathrm{e}_H) \tag{1}$$

Proof. The proof is based on the property of the sets $D_h(\mathbf{m})$. By definition,

$$\mathrm{MS}(\mathcal{S}) = \biguplus_{\mathbf{ed} \in \mathcal{ED}} \{(\mathbf{m}, \mathrm{d}(t_1, \mathbf{m}), \ldots \mathrm{d}(t_H, \mathbf{m})) \in \mathrm{MS}(\mathcal{S}); \mathbf{m} \in [\mathbf{ed}]; \forall h,\; \mathrm{d}(t_h, \mathbf{m}) = e_h\}$$

Since, $\forall m \in [\mathbf{ed}] :\; \mathrm{d}(t_h, \mathbf{m}) = e_h$ for all h, and $\forall \mathbf{m}, \mathbf{m}' \in \mathrm{TRS}(\mathcal{S})$ such that $\mathbf{m} \, \rho \, \mathbf{m}'$:

$$(\mathbf{m}, \mathrm{d}^1, \ldots \mathrm{d}^H) \in \mathrm{MS}(\mathcal{S}) \;\; \text{iff} \;\; (\mathbf{m}', \mathrm{d}^1 \ldots \mathrm{d}^H) \in \mathrm{MS}(\mathcal{S})$$

we can decompose the $(H+1)$ tuple to obtain the result.

• *The age memory case:* in this case, the set of legal descriptors of a transition t_h in the marking $\mathbf{m}$ is $\{s_1, \ldots, s_{t_h}\}$ if the enabling degree of t_h in $\mathbf{m}$ is 1 and is fully determined by the number of interrupted clients, which may be 0 or 1, if the enabling degree of t_h in $\mathbf{m}$ is 0.

In fact, for a given marking, the number of interrupted clients in t_h may depend on the different possible ways to get to that marking, and for makings with the same enabling degree for t_h, the number of interrupted clients of t_h may differ (see [13] for an example). We now partition the MS accordingly to the enabling degree and to the set of the possible numbers of interrupted clients in each phase-type transition.

Let $\mathcal{IC}(t_h, \mathbf{m})$ indicate the set of possible numbers of interrupted client in transition t_h for marking $\mathbf{m}$ ($\mathcal{IC}(t_h, \mathbf{m}) = \{0\}$ or $\{1\}$ or $\{0, 1\}$). We have developed an algorithm which may be applied for both single and multiple server cases, providing the $\mathcal{IC}(t_h, \mathbf{m})$ sets in polynomial time with respect to the size of the reachability graph (see [13] for details).

Since the set $\mathrm{D}_h(\mathbf{m})$ of legal descriptors of a transition t_h for a marking $\mathbf{m}$ only depends on the enabling degree $\mathrm{ed(t_h}, \mathbf{m}) = \mathrm{e}_h$ of t_h in $\mathbf{m}$ and the number of interrupted clients in $\mathbf{m}$, we can write $\mathrm{D}_h(\mathbf{m}) = \mathrm{D}_h(\mathrm{e}_h, \mathrm{i}_h)$ with i_h the set of possible numbers of interrupted clients in $\mathbf{m}$. Consequently, the equivalence relation ρ over the set of markings in TRS is now defined by:

$$\mathbf{m}\ \rho\ \mathbf{m}' \text{ iff } \forall t_h \in \mathrm{TH}\ \ \mathrm{ed}(\mathrm{t_h}, \mathbf{m}) = \mathrm{ed}(\mathrm{t_h}, \mathbf{m}') \text{ and } \mathcal{IC}(t_h, \mathbf{m}) = \mathcal{IC}(t_h, \mathbf{m}')$$

and the high level description of Markovian states is the pair of vectors $(\mathbf{ed}, \mathbf{ic})$ where $\mathbf{ic} = (\mathrm{i}_1, \ldots, \mathrm{i}_H)$ is the vector of possible numbers of interrupted clients in phase-type transitions (remember that each element i_h is a set), and $\mathbf{ed}$ is the vector of their enabling degrees, as before. We shall use as representative of a class the pair of vectors $(\mathbf{ed}, \mathbf{ic})$, and we shall indicate the set of markings of equivalence class $(\mathbf{ed}, \mathbf{ic})$ with $[\mathbf{ed}, \mathbf{ic}]$.

Let $\mathcal{EI}$ be the set of pairs $(\mathbf{ed}, \mathbf{ic})$; if all phase-type transitions of a PH-SPN $\mathcal{S}$ are age memory then

$$\mathrm{MS}(\mathcal{S}) = \biguplus_{(\mathbf{ed},\mathbf{ic}) \in \mathcal{EI}} [\mathbf{ed}, \mathbf{ic}] \times \mathrm{D}_1(\mathrm{e}_1, \mathrm{i}_1) \times \cdots \times \mathrm{D}_H(\mathrm{e}_H, \mathrm{i}_H) \tag{2}$$

where $\mathrm{D}_h(\mathrm{e}_h, \mathrm{i}_h)$ is the set of descriptors of transition t_h compatible with $(\mathrm{e}_h, \mathrm{i}_h)$.

• *General expression for the Markovian state space of PH-SPN:* obviously there is no problem in mixing age memory transitions with enabling memory ones in the same PH-SPN $\mathcal{S}$. Indeed we can rewrite MS in more general form in the following theorem that summarize the two previous results:

Theorem 1. *Given a PH-SPN $\mathcal{S}$, we have*

$$\mathrm{MS}(\mathcal{S}) = \biguplus_{\mathbf{ei} \in \mathcal{EI}} [\mathbf{ei}] \times \mathrm{D}_1(\mathrm{ei}_1) \times \cdots \times \mathrm{D}_H(\mathrm{ei}_H) \tag{3}$$

where $\mathbf{ei} = (\mathrm{ei}_1, \ldots, \mathrm{ei}_H)$*, and* $\mathrm{ei}_h = \mathrm{e}_h$ *if* t_h *is enabling memory, and* $\mathrm{ei}_h = (\mathrm{e}_h, \mathrm{i}_h)$ *if* t_h *is age memory.*

4 Expression of the infinitesimal generator and rate matrix

This section shows the second main result of the paper, namely how the rate matrix of a PH-SPN with H phase-type transitions can be characterized through a tensor algebra expression of matrices of the size of the $(H+1)$ components that have been used to characterize the state space in the previous section. The rate matrix $\mathbf{R}$ is defined as:

$$\mathbf{Q} = \mathbf{R} - \mathbf{\Delta} \tag{4}$$

where $\mathbf{Q}$ is the infinitesimal generator, $\mathbf{\Delta}$ is a diagonal matrix and $\mathbf{\Delta}[i,i] = \sum_{k \neq i} \mathbf{Q}[i,k]$; $\mathbf{R}$ can therefore be obtained from $\mathbf{Q}$ by putting to null all diagonal elements. The use of the rate matrix $\mathbf{R}$ instead of the infinitesimal generator $\mathbf{Q}$, allows for a simpler tensorial expression, as pointed out in numerous papers [16, 3], at the cost of either computing the diagonal elements on the fly, or of explicitly storing the diagonal.

Following the approach presented in [3, 8, 6], a characterization of the set of reachable states as the disjoint union of Cartesian products naturally leads to an organization of $\mathbf{R}$ in block form (disjoint union) and to a tensor expression for each block (Cartesian product). Since we have considered the vector of enabling degrees and, for the age memory case, the vector of sets of numbers of interrupted clients, as high level states, the blocks are determined by the equivalence classes built on these vectors denoted by $\mathbf{ei}$ in a generic form. The structure of $\mathbf{R}$ is given by the next theorem.

Theorem 2. *The block matrices of* $\mathbf{R}$ *are:*

$$\begin{aligned}\mathbf{R}(\mathbf{ei},\mathbf{ei}') = {} & \mathbf{K}_0(\mathrm{TX})(\mathbf{ei},\mathbf{ei}') \bigotimes_{h=1}^{H} \mathbf{Dc}_h(\mathrm{ei}_h,\mathrm{ei}'_h) \\ & + \textstyle\sum_{t_h \in T(\mathbf{ei},\mathbf{ei}')\cap \mathrm{TH}} \mathbf{K}_0(t_h)(\mathbf{ei},\mathbf{ei}') \\ & \qquad \bigotimes_{l=1}^{h-1} \mathbf{Dc}_l(\mathrm{ei}_l,\mathrm{ei}'_l) \otimes \mathbf{Dr}_h(\mathrm{ei}_h,\mathrm{ei}'_h) \bigotimes_{l=h+1}^{H} \mathbf{Dc}_l(\mathrm{ei}_l,\mathrm{ei}'_l)\end{aligned} \tag{5}$$

$$\begin{aligned}\mathbf{R}(\mathbf{ei},\mathbf{ei}) = {} & \bigoplus_{h=1}^{H} \mathbf{R}_h(\mathrm{ei}_h,\mathrm{ei}_h) \\ & + \textstyle\sum_{t_h \in T(\mathbf{ei},\mathbf{ei})\cap \mathrm{TH}} \mathbf{K}_0(t_h)(\mathbf{ei},\mathbf{ei}) \bigotimes_{l=1}^{h-1} \mathbf{I}_{\mathrm{ei}_l} \otimes \mathbf{Dr}_h(\mathrm{ei}_h,\mathrm{ei}_h) \bigotimes_{l=h+1}^{H} \mathbf{I}_{\mathrm{ei}_l}\end{aligned} \tag{6}$$

Due to lack of space, we omit in the present paper, technical details and proofs (which may be found in [13]) about the expression of $\mathbf{R}$ and we give only an intuitive explanation of the theorem.

We have two types of blocks in the matrix, the $(\mathbf{ei},\mathbf{ei})$ diagonal blocks and the off-diagonal $(\mathbf{ei},\mathbf{ei}')$.

The off-diagonal block matrices correspond to a change of marking either due to the firing of an exponential transition (gathered in $\mathbf{K}_0(\mathrm{TX})(\mathbf{ei},\mathbf{ei}')$) which simultaneously (hence the $\bigotimes$ operator) produces modifications of the descriptors of the PH-transitions ($\mathbf{Dc}_h(\mathrm{ei}_h,\mathrm{ei}'_h)$); or due to the external firing (α-firings in [15], the firing of the stage is followed by the choice to leave the transition) of a phase-type transition ($\mathbf{K}_0(t_h)(\mathbf{ei},\mathbf{ei}')$ and $\mathbf{Dr}_h(\mathrm{ei}_h,\mathrm{ei}'_h)$) which also produces simultaneous modifications of other descriptors ($\mathbf{Dc}_l(\mathrm{ei}_l,\mathrm{ei}'_l)$).

The diagonal block matrices come either from internal (or "local", hence the $\bigoplus$ operator) state changes (β-firings in [15], the firing of the stage is followed by the choice of moving to another stage of the same transition) of PH-transitions ($\mathbf{R}_h(\mathrm{ei}_h,\mathrm{ei}_h)$) or from an external firing of a PH-transition which must leave the descriptors of other PH-transitions unchanged ($\mathbf{K}_0(t_h)(\mathbf{ei},\mathbf{ei})$, $\mathbf{Dr}_h(\mathrm{ei}_h,\mathrm{ei}_h)$ and $\mathbf{I}_{\mathrm{ei}_l}$).

The reader will find examples of matrices involved in the theorem in [13].

5 Extension to the multiple server case

In this section, we only briefly review the consequences of the introduction of multiple server PH-transitions; a detailed presentation is given in [13].

The first problem is to precisely define the stochastic semantics of these transitions. In particular, the clients policy (cp) together with the interrupt/resume policy (ip) must be refined so that the following property holds: starting from a Markovian state, the decrease of k in the enabling degree of the corresponding marking $\mathbf{m}$ of the net, leading to interrupt k clients, immediately followed by the increase of k in the enabling degree, leading to activate k clients, restores the initial Markovian state. We propose two types of clients ("ordered" and "unordered" with respect to time) and three adapted interrupt/resume policies ("FIFO", "LIFO" and "Static") covering many encountered modelling needs.

From the definition of these policies we derive new descriptors for multiple server PH-transitions in the enabling as well as in the age memory case.

With these new descriptors, we are able to extend the results of sections 3 and 4. The Markovian state space may still be exactly described as an union of Cartesian products and we present an algorithm providing the sets $\mathcal{IC}(t_h, \mathbf{m})$ (these sets are more complex to compute than for the single server case). The rate matrix $\mathbf{R}$ has the same structure as in Theorem 2 (obviously the $\mathbf{R}_h, \mathbf{Dr}_h$ and $\mathbf{Dc}_h$ matrices are modified accordingly).

6 Evaluation of the method and conclusions

In this paper we have presented a characterization of the state space of SPN with H PH-transitions as a disjoint union of Cartesian products of $H+1$ terms, and an associated tensor expression for the rate matrix of the underlying Markovian process. The memory policies considered are enabling and age memory, with single server as well as multiple/infinite server semantics.

The approach followed is inspired to the tensor algebra method for hierarchical Petri nets [3], and for DSSP [8], but we were able to find a definition of abstract view of the system that leads to a precise characterization of the state space (all and only the reachable states are generated), which was not possible for the mentioned papers. The abstract view is the vector of enabling degree of PH-transitions for enabling memory, enriched by the information on the set of possible numbers of interrupted clients for each age memory transition. Both information can be computed from the reachability graph of the net, the enabling degree computation is straightforward, and we have given an algorithm for the computation of the set of possible interrupted clients.

Tensor algebra has already been applied to PH-SPN in [15], following an SGSPN-like approach [12]: unfortunately the number of non reachable states generated could be very high, a problem that we have completely solved with the method here presented, at the price of a slightly more complicated tensorial expression.

Following the characterization of the state space a tensor formula for the rate matrix has been derived. This formula allows the computation of the state space probability vector without the need to explicitly compute and store the infinitesimal generator of the CTMC underlying the PH-SPN, resulting in a saving in storage, and thus in the possibility of solving more complex models. The probability vector need still to be stored in full, and *this is the real limitation of the approach proposed.* At present state of hardware, the tensor based approach can solve systems with up to some millions states on a workstation, depending on the memory available on the machine [16]; in [5] it is also shown that the amount of memory required for storing the component matrices is negligible with respect to that required for the probability vector.

The contribution of the method proposed is, we hope, twofold. Practical: once implemented and integrated in an SPN tool, it shall enlarge the set of solvable PH-SPN models, thus providing a greater applicability of the model. Theoretical: the characterization of the state space given shows the dependencies between markings, different types of transitions and different types of firing of the same transition, descriptors and memory policies, thus providing, we hope, a step forwards a deeper understanding of PH-SPN.

Acknowledgements We would like to thank Andrea Bobbio for fruitful discussions and explanations on PH-SPN.

References

1. M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In G. Rozenberg, editor, *Advances in Petri Nets 1987*, number 266 in LNCS, pages 132–145. Springer–Verlag, 1987.
2. 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.
3. P. Buchholz. A hierarchical view of GCSPN's and its impact on qualitative and quantitative analysis. *Journal of Parallel and Distributed Computing*, 15(3):207–224, July 1992.
4. P. Buchholz. Aggregation and reduction techniques for hierarchical GCSPN. In *Proc. of the 5th International Workshop on Petri Nets and Performance Models*, pages 216–225, Toulouse, France, October 19–22 1993. IEEE Computer Society Press.
5. P. Buchholz, G. Ciardo, S Donatelli, and P. Kemper. Complexity of Kronecker operations on sparse matrices with applications to the solution of Markov models, 1997. submitted for publication.
6. P. Buchholz and P. Kemper. Numerical analyisis of stochastic marked graphs. In *Proc. 6^{th} Intern. Workshop on Petri Nets and Performance Models*, pages 32–41, Durham, NC, USA, October 1995. IEEE-CS Press.
7. J. Campos, S. Donatelli, and M. Silva. Structured solution of asynchronously communicating stochastic modules: from DSSP to general P/T systems, 1997. submitted for publication.
8. J. Campos, S. Donatelli, and M. Silva. Structured solution of stochastic DSSP systems. In *Proc. of the 7th International Workshop on Petri Nets and Performance Models*, pages 91–100, Saint-Malo, France, June 3–6 1997. IEEE Computer Society Press.
9. P. Chen, S. C. Bruell, and G. Balbo. Alternative methods for incorporating non-exponential distributions into stochastic timed Petri nets. In *Proc. of the third International Workshop on Petri Nets and Performance Models*, pages 187–197, Kyoto, Japan, December 11–13 1989. IEEE Computer Society Press.

10. G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. In *Proc. of the 5th International Workshop on Petri Nets and Performance Models*, pages 170–179, Toulouse, France, October 19–22 1993. IEEE Computer Society Press.
11. A. Cumani. ESP - a package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In *Proc. of the International Workshop on Timed Petri Nets*, pages 144–151, Torino, Italy, July 1-3 1985. IEEE Computer Society Press.
12. S. Donatelli. Superposed generalized stochastic Petri nets: definition and efficient solution. In Robert Valette, editor, *Proc. of the 15th International Conference on Application and Theory of Petri Nets*, number 815 in LNCS, pages 258–277, Zaragoza, Spain, June 20–24 1994. Springer–Verlag.
13. S. Donatelli, S. Haddad, and P. Moreaux. Srtuctured characterization of the markov chains of phase-type SPN. http://www.lamsade.fr/˜ moreaux/tools98/extended.ps, June 1998.
14. S. Haddad and P. Moreaux. Asynchronous composition of high level Petri nets: a quantitative approach. In *Proc. of the 17th International Conference on Application and Theory of Petri Nets*, number 1091 in LNCS, pages 193–211, Osaka, Japan, June 24–28 1996. Springer–Verlag.
15. S. Haddad, P. Moreaux, and G. Chiola. Efficient handling of phase-type distributions in generalized stochastic Petri nets. In *Proc. of the 18th International Conference on Application and Theory of Petri Nets*, number 1248 in LNCS, pages 175–194, Toulouse, France, June 23–27 1997. Springer–Verlag.
16. P. Kemper. Numerical analysis of superposed GSPNs. In *Proc. of the 6th International Workshop on Petri Nets and Performance Models*, pages 52–61, Durham, NC, USA, October 3–6 1995. IEEE Computer Society Press.
17. M. K. Molloy. *On the integration of delay and throughput in distributed processing models.* PhD dissertation, University of California, Los Angeles, CA, USA, September 1981.
18. M. F. Neuts. *Matrix-geometric solutions in stochastic models - an algorithmic approach.* The John Hopkins University Press, London, 1981.
19. B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. In *Proc. of the 1985 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 147–154, Austin, Texas, USA, August 1985. ACM.