UNIVERSITÀ CA' FOSCARI DI VENEZIA
Dipartimento di Informatica
Technical Report Series in Computer Science

Rapporto di Ricerca CS-2010-8

Dicembre 2010

S. Balsamo, P. G. Harrison, A. Marin

Methodological Construction of Product-Form Stochastic Petri-Nets for Performance Evaluation

Dipartimento di Informatica, Università Ca' Foscari di Venezia
Via Torino 155, 30172 Mestre–Venezia, Italy

Methodological Construction of Product-Form Stochastic Petri-Nets for Performance Evaluation

S. Balsamo*, P. G. Harrison**, A. Marin*

***Abstract*—Product-forms in Stochastic Petri-Nets (SPNs) are obtained by a compositional technique for the first time, by combining small SPNs with product-forms in a hierarchical manner. In this way, performance engineering methodology is enhanced by the greatly improved efficiency endowed to the steady state solution of a much wider range of Markov models. Previous methods have relied on analysis of the whole net and so are not incremental – hence they are intractable in all but small models. We show that the product-form condition for open nets depends, in general, on the transition rates, whereas closed nets have only structural conditions for a product-form, except in rather pathological cases. Both the "building blocks" formed by the said small SPNs and their compositions are solved for their product-forms using the Reversed Compound Agent Theorem (RCAT), which, to date, has been used exclusively in the context of process-algebraic models. The resulting methodology provides a powerful, general and rigorous route to product-forms in large stochastic models and is illustrated by several detailed examples.**

I. Introduction

Quantitative models are vital for the design of efficient and reliable computer-communication networks and other operational systems such as software processes, biochemical pathways and healthcare resource scheduling. As well as being capable of representing large and complex systems, such models must also be accessible to system developers, rather than only to the performance specialist. Two popular higher-level formalisms for Markovian systems are Stochastic Petri-nets (SPN) and Stochastic Process Algebras (SPA, or MPA when referring to an SPA with the Markov property). In particular, SPNs, possibly with immediate transitions, have been widely used in software performance engineering. For example, a tool is presented in [8], [7] for the performance evaluation of software architectures defined in UML, by means of discrete event simulation of SPNs into which the UML is translated. Similar studies may be found in [19], [1], [21]. To some degree, SPA and SPN have been in competition with one another. Petri-nets are the more expressive in a natural, graphical way but tend to be harder to analyse structurally, relying on direct mappings to the underlying Markov chain. In contrast, a SPA includes a primitive operator that describes synchronisation between CTMCs (e.g. *cooperation* in PEPA [17]). This can lead to elegant, hierarchical (or *compositional*) structures, but apart from synchronisation, models tend to be specified at a very low level, near to that of individual state transitions on a space of possibly many millions of states. A SPN, on the other hand, is generally specified diagrammatically at a much higher level, in terms of workflows and constraints, but is usually solved by constructing the Markov chain in a "brute force" approach. The choice facing the system designer is, therefore, to use either a Petri-net diagram (or higher level abstraction like UML) that he can understand relatively easily, but which relies on an inefficient solution, or a set of mathematical equations that are akin to a complex, low-level program but which are more conducive to transformation into efficient solutions. In the present work, we show how to achieve the best of both worlds: a way of constructing hierarchically SPNs that are both easy to read and have solutions that are as efficient as those obtained from SPA specifications.

As far as solutions to Markov models are concerned, the immense advantage of SPAs has been that their compositionality has led to the *Reversed Compound Agent Theorem* (RCAT) [10]. This derives, in a purely mechanical way, the product-form solution for the state probabilities of a stationary CTMC defined as a cooperation between two sub-processes, under certain conditions. This approach has unified most of the commonly used product-forms, including multi-class (BCMP [3]) queueing networks, all variants of G-networks (queueing networks with negative customers [11]) and networks that interact via functional rates (where one network's rates depends on the state of another [13]).

Product-form SPNs allow the modeller to identify disjoint sets of interacting sub-models in a net's definition and study them in isolation. Then, the stationary probability distribution of the whole model can be computed as the (normalised) product of the stationary distributions of the sub-models.

A. *Related work*

SPNs with product-form solutions are not unknown. They were first studied in [20] and this result was generalised in [4]. According to this approach, two SPNs, whose stationary solutions are known, can be composed under a strict blocking discipline, to yield a product-form model.

One of the most significant results on product-form SPNs is due to Henderson, Taylor et al. (HT) in [16], [6], where a method was introduced to decide whether a SPN satisfying a set of structural conditions has a product-form, and then to represent each of the sub-models, into which the net is decomposed, by just one place. For this class of models, the following must be checked:

1) A structural condition on the SPN structure.
2) A certain system of linear equations that yields the invariant measure of the so-called *routing process* must have a non-trivial solution. Roughly speaking, these equations may be associated with the traffic equations of queueing networks.

*Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy
**Department of Computing, Imperial College, London, UK

3) The rank of a matrix, whose elements depend on the transition rates, must satisfy a certain condition (the *Rank Theorem*).

Obviously, the latter two, non-structural conditions may be hard to interpret and verify from the modeller's point of view; see the observations in [1]. Hence, substantial research effort has been devoted to relaxing them. In [27], it is shown that Condition 2 is equivalent to a structural condition on the invariants of an SPN. Regarding Condition 3, [9] defines a strict subset of HT's model class, in which the product-form conditions are rate-independent. A similar result has been obtained recently in [22], where it is proved that a class of SPNs with a rate-independent product-form condition is equivalent to the class of chemical reaction networks that fulfils the so-called *deficiency zero* property. It is also shown that a state machine possessing the deficiency zero property is a Jackson network. This suggests that the restrictions required to avoid rate-dependent product-form conditions are quite strong.

B. Contribution

The present work proposes and develops a new approach to finding product-form SPNs that is based on decomposition of a net into special, simple structures that we call *building blocks* (BBs). First, a new product-form is derived for a class of small Petri-net structures, using the most general version of RCAT [14]. These are the BBs that can be composed with each other to provide product-forms for successively larger SPN models by using RCAT again, typically in its simplest, unextended form. This approach treats SPNs with either rate-dependent or rate-independent product-form conditions in a uniform way. However, with respect to previous results, the decomposition of the model into small nets (the BBs) allows the modeller to give an easier interpretation to the rate conditions. In fact, whilst the ensuing result could be *verified* in particular instances by the HT test, based on the incidence matrix of a complete SPN and use of linear algebra, this approach is neither constructive nor compositional [16], [6]. The RCAT approach provides a systematic, programmable basis for constructing ever larger SPNs, with guaranteed product-forms, from primitive BBs. The non-compositional HT route is impractical for large models defined thus since it is not incremental, so that its cubic complexity would become punitive, and in any case, there would be no *a priori* reason to suspect a product-form might exist; most networks do not have product-forms. It is also worthy of note that, with our method, we can study open and closed nets in a uniform way. In the literature, mostly closed nets are considered exclusively [16], [6], [27], [9], [22].

RCAT was first introduced almost ten years ago and, although it has unified almost all the commonly used product-forms under one umbrella, its potential for automation and use in a constructive system design methodology remains unexploited. Software systems in particular are usually designed in a hierarchical way, yet still lack methodologies for managing their quantitative, or non-functional, aspects. Specification languages such as UML have been annotated with performance information and, as noted above, been translated into formalisms suitable for numerical analysis, such as Petri-nets. However, the resulting models are not inherently compositional and so lead to inefficient solutions; even when product-forms exist, they tend to be obscured in the generality of the whole net. This is highlighted in [1], where an algorithm is given that transforms a Generalised Stochastic Petri Net (GSPN) – i.e. a SPN in which immediate transitions are allowed – into a SPN, with the aim of providing a methodology for software performance engineering. Any product-form of the resulting SPN is determined by the HT test but the rank theorem tends to be expensive to apply and the authors point out that its results are often obscure to the modeller.

We advocate an explicitly compositional, non-functional part of the design process through the use of building blocks that can lead to product-forms under conditions that can be verified during that process. Simple queueing nodes have long been used in this way, but the addition of the BB as a new primitive greatly extends the domain of systems that can be constructed in this way. As a final strength of our methodology, observe that Boucherie's product-form condition, with full blocking (which generalises [20]), has been shown to fall under the umbrella of the same theorem that we use to derive the product-form conditions and expressions of HT SPNs [4], [12]. This unifies two approaches to product-form SPNs that up to now had been considered unrelated.

C. Structure of the paper.

We believe our contribution is a significant advance in the quantitative aspect of software design methodology and we illustrate this claim in section II with a class of applications of fork-join nature that do not fit into conventional queueing-style models. We then define the class of Petri-nets we consider in section III and motivate the methodology we are advocating with a small example, which is substantially extended in the remainder of the paper. In Section III-C, we define more general structures for the BBs and apply an extended version of RCAT (called ERCAT [12], [14]) inductively to derive their product-form probability functions at equilibrium, together with the conditions under which they exist. Section IV then illustrates how the BBs can be composed so as to generate larger product-form SPNs and how these may be identified within SPN models that satisfy certain structural properties. In this way, models consisting of compositions of BBs can be hierarchically combined whilst maintaining the product-form property. The methodology is illustrated by several detailed examples. The last section shows the performance analysis of a simple software architecture with synchronisation. The paper concludes with a brief summary and suggestions for future exploitation of the methodology's potential.

II. Illustrative example of hierarchical system construction

To illustrate the efficacy of the Petri net building block methodology in the software design process, we consider a generic class of applications with fork-join style operations. This could be specified as a strictly sequential fork-join, as

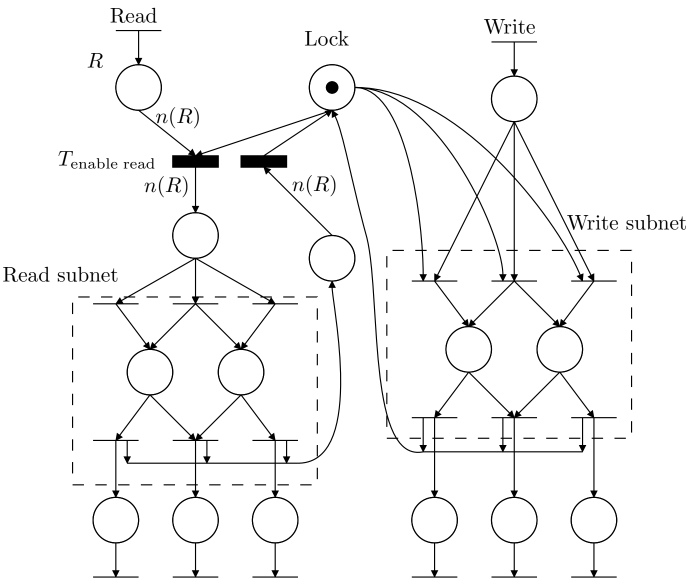

Fig. 1. SPN specifying concurrent access to a database. Solid transitions should be considered immediate transitions. $n(R)$ denotes that the arc weight is the marking of place R just before the last firing of immediate transition $T_{\text{enable read}}$.

in accesses to file segments striped over a set of storage devices, or concurrent selection of items of merchandise, from hoppers of identical items, to make up a client's order in a (perhaps partially automated) warehouse. In the latter case, the sequencing need not be strictly in the order of clients' order's arrivals.

To be concrete, we consider concurrent access to a database that is locked for writes, which occur one at a time. Reads are "gated" in the sense that they accumulate in a buffer whilst a write is progressing and when reads are next enabled, only those in the accumulated set are executed; any more that arrive during a read-sequence are deferred to the next time reads are enabled – possibly immediately after the current read sequence. Such a system can be described by the stochastic Petri net shown in figure 1. The meaning should be clear, but the semantics of SPNs is given in the next section. Notice, however, that the function n yields the number of tokens at the place indicated by its argument in the current marking of the net and that an inhibitor arc is denoted by a circle at its end. The use of n here is solely to allow a transition to clear all the tokens from a place when it fires.

Such a specification can be translated automatically into a simulation model or a Markov chain if the transitions have negative exponential firing times. In the latter case the Petri net model can be investigated at equilibrium by a linear equation solver, or over a finite time period by uniformization and a standard linear algebra package [26], [28].

As noted in the introduction, such models suffer from the state explosion problem, with a combinatorially increasing number of states as the token-population and numbers of places and transitions increase. Hence, decomposition methods are usually applied to find approximate (and sometimes exact) results in terms of solutions to the separate components. Under appropriate assumptions, the composition of the component-solutions yields a product-form and hence an efficient overall solution. Any specification with a rigorous top-down design can be decomposed in this way. Conversely, a bottom-up design deliberately constructed as successive compositions of components of increasing size must possess this hierarchical structure. The results reported in this paper seek decompositions of SPNs into the BBs referred to in the introduction. Moreover, whenever a perfect BB-decomposition exists, i.e. a net can be expressed entirely in terms of BBs, the procedure is automatic.

In the SPN of Figure 1, obvious components are the read and write sub-nets shown in the dashed rectangles. These would be composed as two 'super' places connected in a smaller SPN to the lock-place via the same two transitions as in the original Figure 1. The write sub-net is rather trivial, since it has a maximum of one token. The read sub-net, however, could be highly complex but is perfectly composed of BBs and hence amenable to the techniques of this paper. In fact, we will see several, much more complex, detailed numerical examples in later sections.

III. Stochastic Petri-nets

A stochastic Petri-net is a tuple, SPN $= (\mathcal{P}, \mathcal{T}, \chi(\cdot), \mathbf{I}(\cdot), \mathbf{O}(\cdot), \mathbf{m}_0)$ where:

- $\mathcal{P} = \{P_1, \ldots, P_N\}$ is a set of N places,
- $\mathcal{T} = \{T_1, \ldots, T_M\}$ is a set of M transitions,
- $\chi : \mathcal{T} \to \mathbb{R}^+$ is a positive valued function that associates a firing rate with every transition; we usually write χ_i as an abbreviation for $\chi(T_i)$,
- $\mathbf{I} : \mathcal{T} \to \mathbb{N}^N$ associates an input vector with every transition,
- $\mathbf{O} : \mathcal{T} \to \mathbb{N}^N$ associates an output vector with every transition,
- $\mathbf{m}_0 \in \mathbb{N}^N$ is the initial marking.

A state $\mathbf{m} \in \mathbb{N}^N$ of the model is called a *marking* and represents the numbers m_i of tokens in each place P_i, $i = 1, \ldots, N$. A transition T_i is *enabled* by $\mathbf{m}$ if $\mathbf{m} - \mathbf{I}(t_i)$ has non-negative components. An enabled transition T_i *fires* after an exponentially distributed random time with rate χ_i. In this case, the new state $\mathbf{m}'$ is $\mathbf{m} - \mathbf{I}(T_i) + \mathbf{O}(T_i)$. The net is called *ordinary* if the input and output vector domains are $\{0,1\}^N$.

Graphically, we draw places as circles and transitions as solid bars. If the j-th component of $\mathbf{I}(T_i)$ (respectively $\mathbf{O}(T_i)$) is $k > 0$ then we draw an arc from P_j (respectively T_i) to T_i (respectively P_j) and we label it with k (for ordinary nets we omit the labels).

The reachability set $RS(\mathbf{m}_0)$ is the set of all the possible states of the net, given the initial marking $\mathbf{m}_0$. In general, the problem of determining the reachability set of a SPN is exponential in space. The reachability graph is a graph whose nodes are the states of the reachability set, in which there is an arc from node $\mathbf{m}'$ to $\mathbf{m}''$ if there exists a transition T such that $\mathbf{m}'' = \mathbf{m}' - \mathbf{I}(T) + \mathbf{O}(T)$. The incidence matrix $\mathbf{A}$ of a SPN is an $M \times N$ matrix, row i of which is defined as $\mathbf{O}(T_i) - \mathbf{I}(T_i)$.

The reachability graph can be either finite or infinite and from it, the continuous time Markov chain (CTMC) underlying the SPN model can be derived simply (either lazily or in a parameterised way if the state space is infinite). Henceforth we consider models whose underlying CTMCs are ergodic and so admit a unique, equilibrium, state probability distribution. Calculating this can be a difficult computational task because of the state space explosion problem, which causes even a structurally small net to have a reachability set with high cardinality. In such cases, solution of the global balance equations rapidly becomes numerically intractable.

Some structural properties can be decided by analysis of the incidence matrix and, in particular, P-invariants and T-invariants play a pivotal role. They are defined as follows:

Definition 1 (P-invariant): Given an SPN with incidence matrin $\mathbf{A}$, a vector $\mathbf{P} = (p_1, \ldots, p_N) \in \mathbb{N}^N$ is a *P-invariant* if $\mathbf{AP} = \mathbf{0}$. A net that admits a P-invariant with all positive components is said *conservative* since the weighted sum of the tokens remains constant for each marking of its reachability set.

Definition 2 (T-invariant and support): Given an SPN with incidence matrix $\mathbf{A}$, a vector $\mathbf{X} = (x_1, \ldots, x_M) \in \mathbb{N}^M$ is a *T-invariant* if $\mathbf{A}^T\mathbf{X} = \mathbf{0}$.

The *support* $||\mathbf{X}||$ of a T-invariant $\mathbf{X}$ is the set of transitions corresponding to the non-zero entries of $\mathbf{X}$, i.e., those transitions T_i for which $x_i \neq 0$. A T-invariant $\mathbf{X}$ is *minimal* if there is no other T-invariant $\mathbf{X}'$ such that $x'_i \leq x_i$ for all $i = 1, \ldots, M$. A support of a T-invariant is minimal if no proper non-empty subset of the support is also a support. The *minimal support T-invariant* is the minimal T-invariant with minimal support.

Definition 3 (Closed support T-invariant): A T-invariant X has a *closed support* if for every transition $T_i \in ||X||$ there exists a transition $T_j \in ||X||$ whose input vector is the output vector of T_i, and a transition $T_k \in ||X||$ whose output vector is the input vector of T_i.

We call the fundamental structure that we use to analyse SPNs in product-form a *building block* (BB). We now formally define a BB and give an expression for its product-form solution, together with sufficient conditions for it to exist.

A. Building blocks

In general, a BB consists of a set of places $P_1, \ldots P_N$, a set $\mathcal{T}_I$ of input transitions whose input vectors are null (i.e. $\mathbf{0} = (0, \ldots, 0)$), and a set $\mathcal{T}_O$ of output transitions whose output vectors are null. All the arcs have multiplicity 1. The following definition points out the main restriction on the structure of the BBs, i.e., for each input transition T_y there must exist an output transition T'_y whose input vector is equal to the output vector of T_y.

Definition 4 (Building block (BB)): Given an ordinary (connected) SPN S with set of transitions $\mathcal{T}$ and set of N places $\mathcal{P}$, then S is a *building block* if it satisfies the following conditions:

1) For all $T \in \mathcal{T}$ then either $\mathbf{O}(T) = \mathbf{0}$ or $\mathbf{I}(T) = \mathbf{0}$. In the former case we say that $T \in \mathcal{T}_O$ is an *output transition* while in the latter we say that $T \in \mathcal{T}_I$ is an *input transition*. Note that $\mathcal{T} = \mathcal{T}_I \cup \mathcal{T}_O$ and $\mathcal{T}_I \cap \mathcal{T}_O = \emptyset$, where $\mathcal{T}_I$ is the set of input transitions and $\mathcal{T}_O$ is the set of output transitions.
2) For each $T \in \mathcal{T}_I$, there exists $T' \in \mathcal{T}_O$ such that $\mathbf{O}(T) = \mathbf{I}(T')$ and vice versa.
3) Two places $P_i, P_j \in \mathcal{P}$, $1 \leq i, j \leq N$, are connected, written $P_i \sim P_j$, if there exists a transition $T \in \mathcal{T}$ such that the components i and j of $\mathbf{I}(T)$ or of $\mathbf{O}(T)$ are non-zero. For all places $P_i, P_j \in \mathcal{P}$ in a BB, $P_i \sim^* P_j$, where $\sim^*$ is the transitive closure of $\sim$.

Condition 1 requires that all the transitions are either input or output transitions, while Condition 2 states that if there exists an input transition T_y feeding a subset of places y, then there must be a corresponding output transition T'_y that consumes the tokens from the same subset. Note that Condition 2 is the structural condition for the product-form model class defined by Coleman, Henderson *et al.* in [6]. Finally, Condition 3 simply requires the SPN to be connected.

Figure 2 illustrates an example of a BB consisting of 3 places $\mathcal{P} = \{P_1, P_2, P_3\}$, 3 input transitions $\mathcal{T}_I = \{T_3, T_{23}, T_{12}\}$ and 3 output transitions $\mathcal{T}_O = \{T'_3, T'_{23}, T'_{12}\}$.

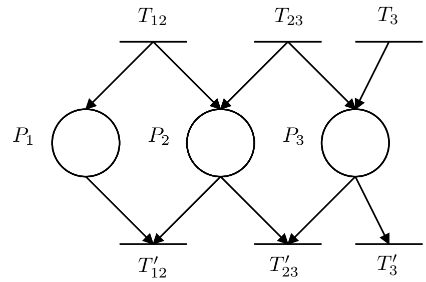

Fig. 2. Example of BB.

Note that if two or more input (output) transitions have the same output (input) vector, we can fuse them in one transition whose rate is the sum of the rates of the original transitions. Therefore, without loss of generality, we assume that all the input (output) transitions have different output (input) vectors.

Finally, to simplify the notation, we use T_y (T'_y) to denote an input (output) transition, where y is the set of place-indices of the non-zero components in the output (input) vector of T_y (T'_y). For instance, transition T_{23} (T'_{23}) in the net of Figure 2 is the transition that produces (consumes) the tokens in P_2 and P_3.

B. Illustrative example

Before deriving the product-form for a general BB, we first analyse a simple one, with just two places, to motivate our approach. The model is depicted in Figure 3. We assume the reader to be reasonably familiar with ERCAT, which forms the basis of the proof, whilst pointing out that the *application* of the result in larger SPNs is much simpler, using only the original form of RCAT, the conditions of which are easy to check. For completeness, we summarise ERCAT in the next section. We use the following conventions: transitions T_y, with null input vector, are always enabled; we call them *input transitions* and denote the set of all such T_y by $\mathcal{T}_I$. Transitions T'_y, with null output vector, are called *output transitions* and the set of all T'_y is denoted by $\mathcal{T}_O$. The subscript y is the set of indices of the output/input places for the input/output

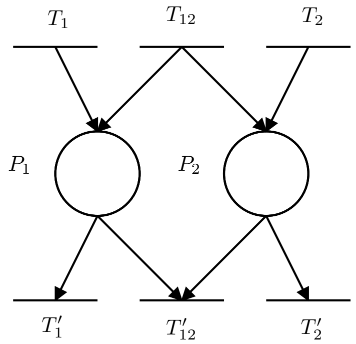

Fig. 3. Basic building block model (BBB).

transition T_y/T'_y. The rates for this SPN are written $\chi_y = \lambda_y$ and $\chi'_y = \mu_y$ where $y = \{1\}, \{2\}, \{1,2\}$.

Let P^1, P^2 be the Markov processes whose states represent the number of tokens in the places P_1, P_2, respectively. We use t_y to denote the action (name) associated with transition T_y, and t'_y similarly for T'_y. Figure 4 illustrates the state transition graph for P^1 and P^2. Note that P^1 controls the synchronised arrivals (t_{12} is active in P^1) while P^2 controls the synchronised departures (t'_{12} is active in P^2). RCAT cannot

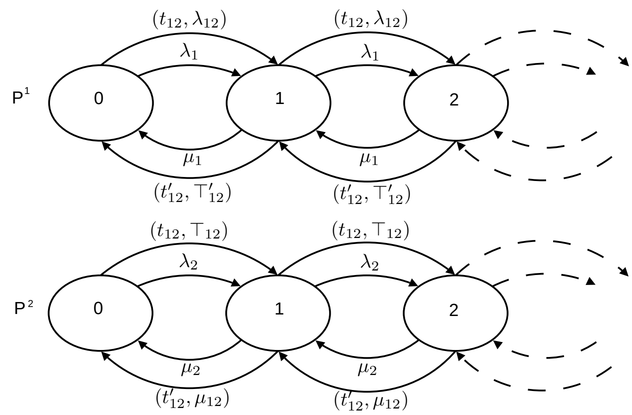

Fig. 4. State transition graphs of P^1 and P^2 in the BBB.

be applied because the passive actions t'_{12} are not enabled in state 0 of P^2, and neither is there an active action t_{12} coming in to state 0 of P^1. Therefore, we use ERCAT.

1) Salient features of ERCAT: When two transitions in different Markov chains synchronise, as in the present description of BBs, we consider one of them to be *active*, with a rate that becomes the rate of the joint transition (e.g., active action t_{12} has rate λ_{12} in P^1), and the other to be *passive*, with unspecified rate (usually denoted by a "top" symbol, e.g. $\top_{12}$ for passive action t_{12} in process P^2). The sets of outgoing, incoming, active and passive actions in each joint state are denoted according to the following notation (wherein $k = 1, 2$):

- $\mathcal{P}_k$: the set of passive action types in the process P^k – each must be passive at all its instances in P^k;
- $\mathcal{A}_k$: the set of active action types in P^k – each must be active at all its instances in P^k;
- $\mathcal{P}^{(i,j)\rightarrow}$: the set of action types that are passive and correspond to transitions out of joint state (i, j) in the synchronisation;
- $\mathcal{P}^{(i,j)\leftarrow}$: the set of action types that are passive and correspond to transitions into joint state (i, j) in the synchronisation;
- $\mathcal{A}^{(i,j)\rightarrow}$: the set of action types that are active and correspond to transitions out of joint state (i, j) in the synchronisation;
- $\mathcal{A}^{(i,j)\leftarrow}$: the set of action types that are active and correspond to transitions into joint state (i, j) in the synchronisation;
- $\alpha_a(i, j)$: the instantaneous transition rate out of state (i, j) corresponding to active action type a;
- $\beta_a(i, j)$: the instantaneous transition rate out of state (i, j) in the *reversed* joint Markov process, corresponding to passive action type a, with rate set to x_a in one of the forwards component-processes (in which passive a is therefore *incoming* to one of the local states i or j.

Under appropriate conditions, a product-form is given by the following theorem, which is a simplified version of the more general result of [14] that derives the reversed process (and so product-form) of pairwise synchronisations amongst any finite number of processes. Prior to application of the theorem, in each component-process, every unspecified rate $\top_a$ of a passive action with type a is replaced by x_a – i.e. we replace $(a, \top_a)$ by (a, x_a).

Theorem 1 (ERCAT): Suppose the following conditions hold in a synchronisation between two processes over the set of action types $L = \mathcal{P}_1 \cup \mathcal{P}_2 = \mathcal{A}_1 \cup \mathcal{A}_2$:

1) Every instance of a reversed action with type $\overline{a}$, of a forwards *active* action of type $a \in \mathcal{A}_k$, has the same rate $\overline{r_a}$ and $\{x_a\}$ satisfy the *rate equations*
$$\{x_a = \overline{r_a} \mid a \in \mathcal{A}_k, 1 \le k \le n\}.$$
2) The forward and reversed passive and active transition rates satisfy
$$\sum_{a \in \mathcal{P}^{(i,j)\rightarrow}} x_a - \sum_{a \in \mathcal{A}^{(i,j)\leftarrow}} x_a = \sum_{\substack{a \in \mathcal{P}^{(i,j)\leftarrow} \\ \setminus \mathcal{A}^{(i,j)\leftarrow}}} \beta_a(i,j) - \sum_{\substack{a \in \mathcal{A}^{(i,j)\rightarrow} \\ \setminus \mathcal{P}^{(i,j)\rightarrow}}} \alpha_a(i,j).$$

Then there is a product-form solution for the equilibrium state probabilities (when these exist), which is $\pi(i, j) \propto \pi_1(i)\pi_2(j)$ where $\pi_k(.)$ is the equilibrium probability function for the process P_k, $k = 1, 2$.

To apply the theorem it is important to be able to calculate the reversed rate of a transition. This is straightforward in any stationary Markov process in which the forward rate and the stationary state probabilities are known. Suppose a transition from state i to state j has rate λ. Then the reversed rate is (see [18], for example):
$$\overline{\lambda} = \frac{\pi(i)\lambda}{\pi(j)} \tag{1}$$

2) Application of ERCAT to the basic BB: We first consider the constraints on the transition rates imposed by the second condition of ERCAT. Note that the outgoing and incoming transitions and rates are the same for all states P_k^1 and P_k^2 with $k > 0$. Therefore, it is only necessary to analyse the four joint states $(0,0)$, $(0,k)$, (k,k), $(k,0)$ for some $k > 0$:

- $(0,0)$. For this state we have the following sets:

$$\mathcal{P}^{(0,0)\to} = \{t_{12}\} \qquad \mathcal{A}^{(0,0)\leftarrow} = \{t'_{12}\}$$
$$\mathcal{P}^{(0,0)\leftarrow} \setminus \mathcal{A}^{(0,0)\leftarrow} = \emptyset \qquad \mathcal{A}^{(0,0)\to} \setminus \mathcal{P}^{(0,0)\to} = \emptyset.$$

  In order to apply ERCAT, Condition 2 must be satisfied and so we require $x_{12} = x'_{12}$.
- $(0,k)$. The corresponding sets for this state are:

$$\mathcal{P}^{(0,k)\to} = \{t_{12}\} \quad \mathcal{A}^{(0,k)\leftarrow} = \{t'_{12}\}$$
$$\mathcal{P}^{(0,k)\leftarrow} \setminus \mathcal{A}^{(0,k)\leftarrow} = \{t_{12}, t'_{12}\} \setminus \{t'_{12}\} = \{t_{12}\}$$
$$\mathcal{A}^{(0,k)\to} \setminus \mathcal{P}^{(0,k)\to} = \{t_{12}, t'_{12}\} \setminus \{t_{12}\} = \{t'_{12}\}.$$

  To satisfy Condition 2, we require

$$x_{12} - x'_{12} = \beta_{12}(0k) - \alpha'_{12}(0k)$$

  where we use the abbreviations $\alpha_{12}(0k) \equiv \alpha_{t_{12}}(0k)$ and $\alpha'_{12}(0k) \equiv \alpha_{t'_{12}}(0k)$ and similarly for β, β'. But since $x_{12} = x'_{12}$, we conclude $\beta_{12}(0k) = \alpha'_{12}(0k) = \mu_{12}$, t'_{12} being active in process P^2.
- $(k,0)$ and (k,k). For these states the ERCAT Condition 2 is trivially satisfied since $\mathcal{P}^{(k,0)\to} = \mathcal{A}^{(k,0)\leftarrow} = \mathcal{P}^{(k,k)\to} = \mathcal{A}^{(k,k)\leftarrow} = L$. In fact, every passive action and every reversed action corresponding to an active action are always enabled, so that for these states the original RCAT structural conditions are satisfied [10].

The rate equations (Condition 1) for this model require, using Equation (1),

$$\begin{cases} x_{12} = \frac{\lambda_{12}(\mu_1 + x'_{12})}{\lambda_{12} + \lambda_1} \\ x'_{12} = \frac{\mu_{12}(\lambda_2 + x_{12})}{\mu_{12} + \mu_2} \end{cases}. \tag{2}$$

Since $x_{12} = x'_{12}$, we obtain:

$$\begin{cases} x_{12} = \frac{\lambda_{12}\mu_1}{\lambda_1} \\ x'_{12} = \frac{\mu_{12}\lambda_2}{\mu_2} \end{cases},$$

which gives the condition:

$$\lambda_1 \lambda_2 \mu_{12} = \lambda_{12} \mu_1 \mu_2. \tag{3}$$

Finally, we verify that

$$\beta_{12}(0k) = \frac{x_{12}(\mu_2 + \mu_{12})}{x_{12} + \lambda_2} = \mu_{12},$$

as required.

To sum up, when Condition (3) is satisfied, the steady-state probability function $\pi(m_1, m_2)$ is in product-form. Moreover, the Markov processes associated with P^1 and P^2 are simple birth-death processes with equilibrium probabilities (when they exist) proportional to

$$\left(\frac{\lambda_1 + \lambda_{12}}{\mu_1 + x'_{12}}\right)^{m_1} \quad \text{and} \quad \left(\frac{\lambda_2 + x_{12}}{\mu_2 + \mu_{12}}\right)^{m_2}.$$

Given the above conditions, the joint stationary probabilities therefore simplify to

$$\pi(m_1, m_2) \propto \left(\frac{\lambda_1}{\mu_1}\right)^{m_1} \left(\frac{\lambda_2}{\mu_2}\right)^{m_2}. \tag{4}$$

C. Product-form for arbitrary BBs

Arbitrary BBs can be analysed progressively by following exactly the same method as was used in the preceding case of a two-place BB. Ever larger BBs are constructed – and then solved using ERCAT – by successively adding one place with connections to any of the existing places. This leads to an inductive proof of our main result, which gives the conditions and the product-form solution for a BB of arbitrary structure and size.

We use essentially the same notation and conventions as for the basic two-place BB. To recap, the subscript y is the set of (output/input) place-indices for the (input/output) transition T_y/T'_y (with null input/output vector). $\mathcal{T}_I$ and $\mathcal{T}_O$ denote the sets of all input and output transitions respectively. The input and output rates are written $\chi_y = \lambda_y$ and $\chi'_y = \mu_y$ respectively, where $y \in 2^{\{1,\ldots,N\}} \setminus \emptyset$. Finally, we again write t_y/t'_y for the action name associated with transition T_y/T'_y.

Theorem 2: Consider a BB S with N places and let $\mathcal{N} \subseteq 2^{\{1,\ldots,N\}} \setminus \emptyset$. Let $\rho_y = \lambda_y/\mu_y$ for $T_y, T'_y \in \mathcal{T}, |y| \geq 1$. If the following system of equations has a unique solution ρ_i, $(1 \leq i \leq N)$:

$$\begin{cases} \rho_y = \prod_{i \in y} \rho_i & \forall y: \ T_y, T'_y \in \mathcal{T} \wedge |y| > 1 \\ \rho_i = \frac{\lambda_i}{\mu_i} & \forall i: \ T_i, T'_i \in \mathcal{T}, 1 \leq i \leq N \end{cases} \tag{5}$$

then the net's balance equations – and hence stationary probabilities when they exist – have product-form solution:

$$\pi(m_1, \ldots, m_N) \propto \prod_{i=1}^{N} \rho_i^{m_i}. \tag{6}$$

Note that System (5) have N unknowns $\rho_1, \ldots, \rho_N$. However, for each $1 \leq i \leq N$ such that $T_i, T'_i \in \mathcal{T}$ we straightforwardly have $\rho_i = \lambda_i/\mu_i$. Moreover, the system can be solved as a linear system by taking logarithms on both sides. Henceforth, we use lower case letters y, w, z to denote elements of $\mathcal{N}$ with arbitrary cardinality, letters i, j to denote singletons $\{i\}$ and $\{j\}$ in $\mathcal{N}$ and, finally, we write w, x instead of $w \cup x$.

Note also that a 2-place net with single input/output transition pair T_{12}, T'_{12} has only one equation $\rho_{12} = \rho_1 \rho_2$, which clearly does not have a unique solution. Since the two places would be exactly synchronised, with the same numbers of tokens in each at all times, a product-form solution cannot exist; rather the solution would be that for a duplicated single-place model. If one additional transition pair T_1, T'_1 or T_2, T'_2 (or both) were added, the equations would have a unique solution, corresponding to what is now an asynchronous model with a product-form.

1) Proof of Theorem 2: The proof is by induction on the number of places N. For $N = 1$, the net is a simple M/M/1 queue and the theorem states the standard geometric result for the stationary probabilities.

Now consider a model S consisting of $N+1$ places, with $N \geq 1$. Then S is a synchronisation between two Markov models:

- S', which is the model S with place P_{N+1} removed. Every pair of transitions $T_y, T'_y \in \mathcal{T}$ with $(N+1) \notin y$ is still present in S' with the same rates λ_y, μ_y. However, the transition pair $T_{y,N+1}, T'_{y,N+1} \in \mathcal{T}$ with $y \neq \emptyset$, from the point of view of S', has the same effect as T_y and T'_y, respectively, but with rates $\lambda_{y,N+1}$ and $\mu_{y,N+1}$. Therefore, in the analysis of S', their net rates are $\lambda_y + \lambda_{y,N+1}$ and $\mu_y + \mu_{y,N+1}$. Transitions $T_{y,N+1}$ are taken to be active in S' with rate $\lambda_{y,N+1}$, while transitions $T'_{y,N+1}$ are assumed to be passive in S' and, according to ERCAT convention, we replace their unknown rate by $x'_{y,N+1}$. Consequently, in S', T_y acquires rate $\lambda_y + \lambda_{y,N+1}$ and T'_y acquires rate $\mu_y + x'_{y,N+1}$. These are *parallel transitions* in the state-transition graph of S', the second component of each pair synchronising and the first components not.
- P^{N+1}, i.e., the model consisting of the single place P_{N+1}, together with its connected transitions. Here again, a transition from state k to state $k+1$ occurs due to a firing of T_{N+1} (if $T_{N+1} \in \mathcal{T}$) or of $T_{y,N+1}$ if $T_{y,N+1} \in \mathcal{T}$ and $y \neq \emptyset$; and similarly for a transition from state $k+1$ to state k. Transitions of the form $T'_{y,N+1}$ are active in P^{N+1}, with rates $\mu_{y,N+1}$, while transitions $T_{y,N+1}$ are passive and their rates are set to $x_{y,N+1}$. As in the model S', the transition pairs $(T_y, T_{y,N+1})$ and $(T'_y, T'_{y,N+1})$ are aggregated for each of the valid sets y, as depicted in Figure 5.

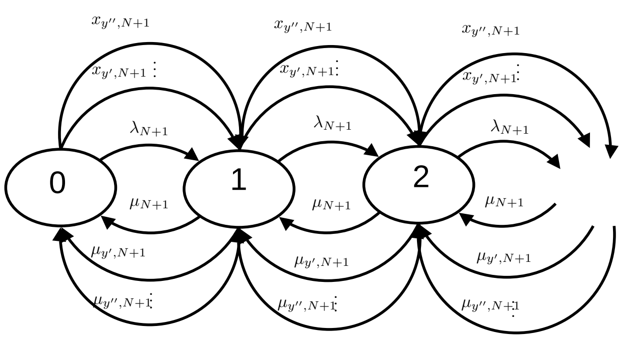

Fig. 5. Process P^{N+1} after replacing $\top$ with x

Model S can now be defined as the cooperation between S' and P^{N+1} over cooperation set $L \cup L'$ where $L = \{t_y \mid N+1 \in y, T_y \in \mathcal{T}\}$ and $L' = \{t'_y \mid N+1 \in y, T'_y \in \mathcal{T}\}$. The induction step may now be performed following the pattern of the analysis of the basic BB in section III-B2, beginning with (structural) condition 2.

Structural constraints: We write the states of S as $(\mathbf{m}, k)$, where $\mathbf{m}$ is a state of S' and the non-negative integer k is a state of P^{N+1}. We extend the notation of section III-B1 to simplify the application of Theorem 1 (ERCAT) by splitting the sets of action types into those in L and those in L'. Thus we write, for the sets of outgoing actions:

$$\begin{aligned}\mathcal{P}^{(k)\rightarrow} &= \mathcal{P}^{(\mathbf{m},k)\rightarrow} \cap L\\ \mathcal{P}'^{(\mathbf{m})\rightarrow} &= \mathcal{P}^{(\mathbf{m},k)\rightarrow} \cap L'\\ \mathcal{A}^{(\mathbf{m})\rightarrow} &= \mathcal{A}^{(\mathbf{m},k)\rightarrow} \cap L\\ \mathcal{A}'^{(k)\rightarrow} &= \mathcal{A}^{(\mathbf{m},k)\rightarrow} \cap L'\end{aligned}$$

and similarly for the sets of incoming actions by replacing $\rightarrow$ with $\leftarrow$. Because input transitions are always enabled and output transitions can leave the system in any state, $\mathcal{P}^{(k)\rightarrow} = \mathcal{A}^{(\mathbf{m})\rightarrow} = L$ and $\mathcal{A}'^{(k)\leftarrow} = \mathcal{P}'^{(\mathbf{m})\leftarrow} = L'$ for all states $(\mathbf{m}, k)$. Therefore, the right hand side of Condition 2 of ERCAT can be written

$$\sum_{\substack{t_{y,N+1} \in \mathcal{P}^{(k)\leftarrow}\\ \setminus \mathcal{A}^{(\mathbf{m})\leftarrow}}} \beta_{y,N+1}(\mathbf{m}, k) - \sum_{\substack{t'_{y,N+1} \in \mathcal{A}'^{(k)\rightarrow}\\ \setminus \mathcal{P}'^{(\mathbf{m})\rightarrow}}} \alpha'_{y,N+1}(\mathbf{m}, k).$$

For states with $k = 0$, we also have $\mathcal{A}'^{(k)\rightarrow} = \mathcal{P}^{(k)\leftarrow} = \emptyset$, so that the right hand side of Condition 2 becomes zero.

Let $\mathcal{O}(\mathbf{m}) = \{y \mid T'_y \text{ is enabled by marking } \mathbf{m}\} = \{y \mid T'_y \in \mathcal{T} \wedge m_i > 0 \forall i \in y\}$. Condition 2 of ERCAT now yields the following:

$k = 0$. In this case, Condition 2 simplifies to

$$\sum_{(y,N+1) \in \mathcal{O}(\mathbf{m})} x'_{y,N+1} + \sum_{\substack{T_{y,N+1} \in \mathcal{T}\\ y \neq \emptyset}} x_{y,N+1} = \sum_{(y,N+1) \in \mathcal{O}(\mathbf{m})} x_{y,N+1} + \sum_{\substack{T_{y,N+1} \in \mathcal{T}\\ y \neq \emptyset}} x'_{y,N+1}$$

Since this must hold for all markings $\mathbf{m}$, we can choose $\mathbf{m}$ successively to show that $x_{y,N+1} = x'_{y,N+1}$ for all y.

$k > 0$. The left hand side of Condition 2 is independent of k and so (again) equal to zero. Hence, noting that $\mathcal{A}'^{(k)\rightarrow} = L'$ and $\mathcal{P}^{(k)\leftarrow} = L$ for $k > 0$,

$$\sum_{(y,N+1) \notin \mathcal{O}(\mathbf{m})} \beta_{y,N+1}(\mathbf{m}, k) = \sum_{(y,N+1) \notin \mathcal{O}(\mathbf{m})} \alpha'_{y,N+1}(\mathbf{m}, k), \quad (7)$$

where $\alpha'_{y,N+1}(\mathbf{m}, k) = \mu_{y,N+1}$ for all $\mathbf{m}, k$.

Sub-model P^{N+1}: We now consider the rate equations and Condition (1) of ERCAT. In model P^{N+1}, we find the reversed rates of the active actions $t'_{y,N+1}$ (using the geometric stationary probabilities of P^{N+1}) to obtain the rate equation:

$$x'_{y,N+1} = \frac{\mu_{y,N+1}\left(\sum_{\substack{w \neq \emptyset\\ T_{w,N+1} \in \mathcal{T}_I}} x_{w,N+1} + \lambda_{N+1}\right)}{\mu_{N+1} + \sum_{\substack{w \neq \emptyset\\ T'_{w,N+1} \in \mathcal{T}_O}} \mu_{w,N+1}}, \quad (8)$$

where $y \neq \emptyset$ and $\lambda_{N+1} = \mu_{N+1} = 0$ if $T_{N+1}, T'_{N+1} \notin \mathcal{T}$. Recalling that $T_y \in \mathcal{T}_I$ if and only if $T'_y \in \mathcal{T}_O$, and that

$x'_{y,N+1} = x_{y,N+1} \forall y$ from the preceding analysis, summing over y and rearranging, we obtain

$$\sum_{\substack{w \neq \emptyset \\ T_{w,N+1} \in \mathcal{T}_I}} x_{w,N+1} = \rho_{N+1} \sum_{\substack{w \neq \emptyset \\ T_{w,N+1} \in \mathcal{T}_I}} \mu_{w,N+1}$$

and hence, substituting back into Equation (8), we obtain $x_{y,N+1} = x'_{y,N+1} = \rho_{N+1}\mu_{y,N+1} \;\forall T_{y,N+1}, T'_{y,N+1} \in \mathcal{T}, y \neq \emptyset$. The reversed rates are therefore constant, as required by ERCAT.

Furthermore, the reversed rate of $x_{y,N+1}$ is

$$\begin{aligned} &\beta_{y,N+1}(\mathbf{m}, k) \\ = \; & \frac{x_{y,N+1}\left(\mu_{N+1} + \sum_{\substack{w \neq \emptyset \\ T'_{w,N+1} \in \mathcal{T}_O}} \mu_{w,N+1}\right)}{\sum_{\substack{w \neq \emptyset \\ T_{w,N+1} \in \mathcal{T}_I}} x_{w,N+1} + \lambda_{N+1}} \\ = \; & \mu_{y,N+1}, \end{aligned}$$

for all $\mathbf{m}, k$, which is consistent with Equation (7).

Inductive step: sub-model S': In model S', let $\mathring{\mathcal{T}} = \mathring{\mathcal{T}}_I \cup \mathring{\mathcal{T}}_O$ be the set of the transitions of S'. To make use of the inductive hypothesis, we now show that Conditions (5) hold for S', i.e. that

$$\mathring{\rho}_z = \prod_{i \in z} \mathring{\rho}_i, \;\; \mathring{T}_z, \mathring{T}'_z \in \mathring{\mathcal{T}}, |z| > 1, \tag{9}$$

where:

$$\mathring{\rho}_z = \frac{\lambda_z + \lambda_{z,N+1}}{\mu_z + x'_{z,N+1}} \tag{10}$$

for $|z| \geq 1$, and $\lambda_z = \mu_z = 0$ if $T_z, T'_z \notin \mathcal{T}$ and $\lambda_{z,N+1} = \mu_{z,N+1} = x'_{z,N+1} = 0$ if $T_{z,N+1}, T_{z',N+1} \notin \mathcal{T}$.

Now, for all z such that $\mathring{T}_z, \mathring{T}'_z \in \mathring{\mathcal{T}}$, either $T_z, T'_z \in \mathcal{T}$ or $T_{z,N+1}, T'_{z,N+1} \in \mathcal{T}$ and so

$$\begin{aligned} \mathring{\rho}_z &= \frac{\lambda_z + \lambda_{z,N+1}}{\mu_z + \rho_{N+1}\mu_{z,N+1}} \\ &= \frac{\rho_z\mu_z + \mu_{z,N+1}\rho_{z,N+1}}{\mu_z + \rho_{N+1}\mu_{z,N+1}} \text{ by definition} \\ &= \frac{\rho_z\mu_z + \rho_z\rho_{N+1}\mu_{z,N+1}}{\mu_z + \rho_{N+1}\mu_{z,N+1}} \text{ by Equations (5)} \\ &= \rho_z = \prod_{i \in z}^{N} \rho_i. \end{aligned} \tag{11}$$

For $1 \leq i \leq N$, taking $z = \{i\}$, we have $\mathring{\rho}_i = \rho_i$ if $T_i, T'_i \in \mathring{\mathcal{T}}$ or $T_{i,N+1}, T'_{i,N+1} \in \mathring{\mathcal{T}}$. Otherwise, for i such that $T_i, T'_i, T_{i,N+1}, T'_{i,N+1} \notin \mathring{\mathcal{T}}$, choosing $\mathring{\rho}_i = \rho_i$ satisfies Equations (9), as has just been verified. Moreover, this solution is unique. To prove this, suppose that $\{\rho'_1, \ldots, \rho'_N\}$ is another solution to the Equations (9) and note that any y for which $T_y, T'_y \in \mathcal{T} \setminus \mathring{\mathcal{T}}$, $N+1 \in y$. Let $y = w \cup \{N+1\}$ where $T_w, T'_w \in \mathring{\mathcal{T}}$. Then, by Equations (5), $\rho_y = \rho_w\rho_{N+1} = \mathring{\rho}_w\rho_{N+1}$ by Equation (11). Therefore, $\{\rho'_1, \ldots, \rho'_N, \rho_{N+1}\}$ is a solution to Equations (5) and by the uniqueness hypothesis in the theorem, $\rho'_i = \rho_i$ for $1 \leq i \leq N$.

Having thus established Conditions (9)[1], we conclude by induction that the stationary probabilities of S' are

$$\pi(m_1, \ldots, m_N) \propto \prod_{i=1}^{N} \mathring{\rho}_i = \prod_{i=1}^{N} \rho_i.$$

and hence the reversed rate of an active action $t_{y,N+1}$, with forward rate $\lambda_{y,N+1}$, is $\lambda_{y,N+1}/\mathring{\rho}_y$. Hence, we write

$$\begin{aligned} x'_{y,N+1} &= \lambda_{y,N+1}/\mathring{\rho}_y = \lambda_{y,N+1}/\rho_y \\ &= \rho_{N+1}\lambda_{y,N+1}/\rho_{y,N+1} = \rho_{N+1}\mu_{y,N+1} \end{aligned}$$

consistent with the calculation for $x_{y,N+1}$ and the ERCAT constraints.

The conditions of ERCAT have now all been verified for the $(N+1)$-place net, which thereby has product-form solution

$$\begin{aligned} \pi(m_1, \ldots, m_{N+1}) &\propto \prod_{i=1}^{N} \mathring{\rho}_i^{m_i} \rho_{N+1}^{m_{N+1}} \\ &= \prod_{i=1}^{N} \rho_i^{m_i} \rho_{N+1}^{m_{N+1}} = \prod_{i=1}^{N+1} \rho_i^{m_i} \end{aligned}$$

as required. ♠

From Theorem 2 we can straightforwardly derive the following corollary that will be useful in what follows.

Corollary 1: Given a BB in product-form by Theorem 2, let $T'_y \in \mathcal{T}_O$. The reversed rate of transition T'_y is λ_y, i.e., the rate of the corresponding input transition.

Proof: By hypothesis the stationary distribution of the BB is given by Equation (6). Let $\mathbf{I}(T'_y)$ be the input vector of $T'_y \in \mathcal{T}_O$ and let $\mathbf{m}$ be a general reachable marking that enables T'_y, i.e., $\mathbf{m} > \mathbf{I}(T'_y)$. The reversed rate $\overline{\mu_y}$ of T'_y is:

$$\overline{\mu_y} = \frac{\pi(\mathbf{m})}{\pi(\mathbf{m} - \mathbf{I}(T'_y))}\mu_y = \mu_y \prod_{i \in y} \rho_i.$$

Since ρ_i are the solution of System (5), we straightforwardly have:

$$\overline{\mu_y} = \rho_y\mu_y = \lambda_y,$$

as required. ♠

D. Example: 3-place BB

Example 1: Consider the BB of Figure 2. Using Theorem 2 we have the following conditions for a product-form:

$$\begin{cases} \rho_{12} = \rho_1\rho_2 \\ \rho_{23} = \rho_2\rho_2 \\ \rho_3 = \lambda_3/\mu_3 \end{cases}$$

These give $\rho_2 = \rho_{23}/\rho_3$ and $\rho_1 = \rho_{12}\rho_3/\rho_{23}$. Therefore, the steady-state probabilities are given by:

$$\pi(m_1, m_2, m_3) \propto \left(\frac{\lambda_{12}\lambda_3\mu_{23}}{\mu_{12}\mu_3\lambda_{23}}\right)^{m_1} \left(\frac{\lambda_{23}\mu_3}{\mu_{23}\lambda_3}\right)^{m_2} \left(\frac{\lambda_3}{\mu_3}\right)^{m_3}$$

[1]Notice that we have not calculated the reversed passive rates $\beta'_{y,N+1}$ since these do not appear in the ERCAT constraint equations.

IV. Composition of BBs

Henceforth, the models we consider are assumed to be ergodic, i.e., to have a steady-state. We now take advantage of the RCAT approach by showing how Theorem 2 can be used to study complex product-form SPN models. Since the BBs themselves are in product-form (by ERCAT), we observe that:

1) the reversed rates of the reversed actions corresponding to the output transition firings are constant;
2) the input transitions are always enabled;
3) each state of the BB can be reached by the firing of any output transition.

Thus, we can use RCAT to combine two BBs, i.e., we do not need to check the conditions of the ERCAT theorem. Considering only pairwise compositions of several BBs for each action type, we can apply MARCAT to solve a model comprising more than two BBs in one step.

A. Semantics of BB composition

In general, we require that the firing of an output transition of one BB corresponds to the firing of an input transition of another. For example, let us consider two BBs S_1 and S_2 and suppose that we want the input transition $T_y^{(2)}$ of S_2 to fire when the output transition $T_x'^{(1)}$ of S_1 fires. We can model this behaviour by replacing transitions $T_x'^{(1)}$ and $T_y^{(2)}$ with a transition T whose input vector is $\mathbf{I}(T_x)$ and whose output vector is $\mathbf{O}(T_y)$.

A slightly more complicated situation is that in which we model an input transition $T_y^{(2)}$ of S_2, whose firing is synchronised with *two* output transitions of other BBs, e.g., $T_x'^{(1)}$ and $T_w'^{(1)}$, meaning that T_y fires when $T_x'^{(1)}$ fires *or* when $T_w'^{(1)}$ fires. In RCAT terminology, associate the passive action a with transition $T_y^{(2)}$ and the active action a with both transitions $T_x'^{(1)}$ and $T_w'^{(1)}$. Then, in the synchronisation between the BBs, there are transitions $u \xrightarrow{a} s$ and $v \xrightarrow{a} s$ due to a, for joint states u, v, s. Intuitively, we can define an equivalent net by introducing an extra "infinitely fast" state s' such that $u \xrightarrow{a'} s'$, $v \xrightarrow{a''} s'$ and $s' \xrightarrow{a} s$, where a', a'' are not in the cooperation set. RCAT now applies simply, where the reversed rate of the active action a is the sum of the reversed rates of the pair of original active actions a, i.e. of a' and a''.[2] Recall that, by Corollary 1, the reversed rate $\overline{\mu_y}$ of a transition T_y' in a BB is λ_y. This intuition is made rigorous by applying ERCAT when (at every instance) the active actions a are replaced by a' and a'' respectively and the passive action a is replaced by *parallel* actions (same source and destination states) a' and a''.

In this case we replace $T_x'^{(1)}$, $T_w'^{(1)}$ and T_y with two transitions whose output vectors are both $\mathbf{O}(T_y^{(2)})$ and input vectors are $\mathbf{I}(T_x^{(1)})$ and $\mathbf{I}(T_w^{(1)})$.

The question now is how to decide if a composition of BBs has a product-form and how the expression for its stationary state probability function can then be derived. The key idea is to consider the rates of the input transitions to be unknown, i.e., in the process definition they correspond to passive actions. Then, we derive a system of equations where each unknown rate is set to the reversed rate of the corresponding output transition. If the system admits a solution that is compatible with the product-form conditions of each BB, then the SPN has product-form.

[2] This is because the total outgoing rate from state s remains unchanged, but this is equal to the total *reversed* rate out of state s by Kolmogorov's criteria [18].

We now illustrate this technique with an example.

Example 2 (Composition of BBs): Consider now the composition of BBs depicted in Figure 6-A. According to the semantics of BB-composition, we need to solve the net of Figure 6-B.

The BB with P_1, P_2, P_3 has been solved already (in Section III-D), while the BB with P_4 and P_5 is in product-form under the condition:

$$x_{45}\mu_4\mu_5 = \mu_{45}\lambda_4\lambda_5$$

with stationary probability function

$$\pi(m_4, m_5) \propto \left(\frac{\lambda_4}{\mu_4}\right)^{m_4} \left(\frac{\lambda_5}{\mu_5}\right)^{m_5}.$$

The unknowns for the composed model are x_{45} and x_{23}, i.e., the rates of transitions T_{45} and T_{23}, respectively. The rate equations are:

$$\begin{cases} x_{45} = \overline{\mu_{12}} + \overline{\mu_{23}} = \lambda_{12} + x_{23} \\ x_{23} = \overline{\mu_5} = \lambda_5 \end{cases}$$

and the condition for a product-form solution becomes:

$$(\lambda_{12} + \lambda_5)\mu_4\mu_5 = \mu_{45}\lambda_4\lambda_5$$

which yields stationary product-form solution:

$$\pi(m_1, m_2, m_3, m_4, m_5) \propto \left(\frac{\lambda_{12}\lambda_3\mu_{23}}{\mu_{12}\mu_3\lambda_5}\right)^{m_1} \left(\frac{\lambda_5\mu_3}{\mu_{23}\lambda_3}\right)^{m_2} \left(\frac{\lambda_3}{\mu_3}\right)^{m_3} \left(\frac{\lambda_4}{\mu_4}\right)^{m_4} \left(\frac{\lambda_5}{\mu_5}\right)^{m_5}.$$

B. Ordinary SPNs in product-form

We now show how to use Theorem 2, combined with RCAT, to study a wider class of ordinary SPNs in product-form. The models we study have the property that the input vector of every transition is equal to the output vector of at least one transition, and vice versa. We call this class of SPN models *Product-form Ordinary SPN*s (POSPNs).

Definition 5 (POSPN): A Product-form Ordinary Stochastic Petri-net is a SPN in which $\{\mathbf{I}(T_i) \mid T_i \in \mathcal{T}\} = \{\mathbf{O}(T_i) \mid T_i \in \mathcal{T}\}$.

Not all POSPNs have a product-form solution since conditions on the rates of the constituent transitions may be required, as we shall see. Note that, although the structural condition meets that required by the HT approach [16], [6], we pursue a different analysis technique that exploits the modularity of product-form model specifications and allows an efficient hierarchical analysis. Indeed, our approach is constructive, building up ever larger SPNs from smaller ones, knowing that a product-form is guaranteed under conditions that are generated in the same procedure. The HT approach has to be applied to the ultimate SPN constructed, with

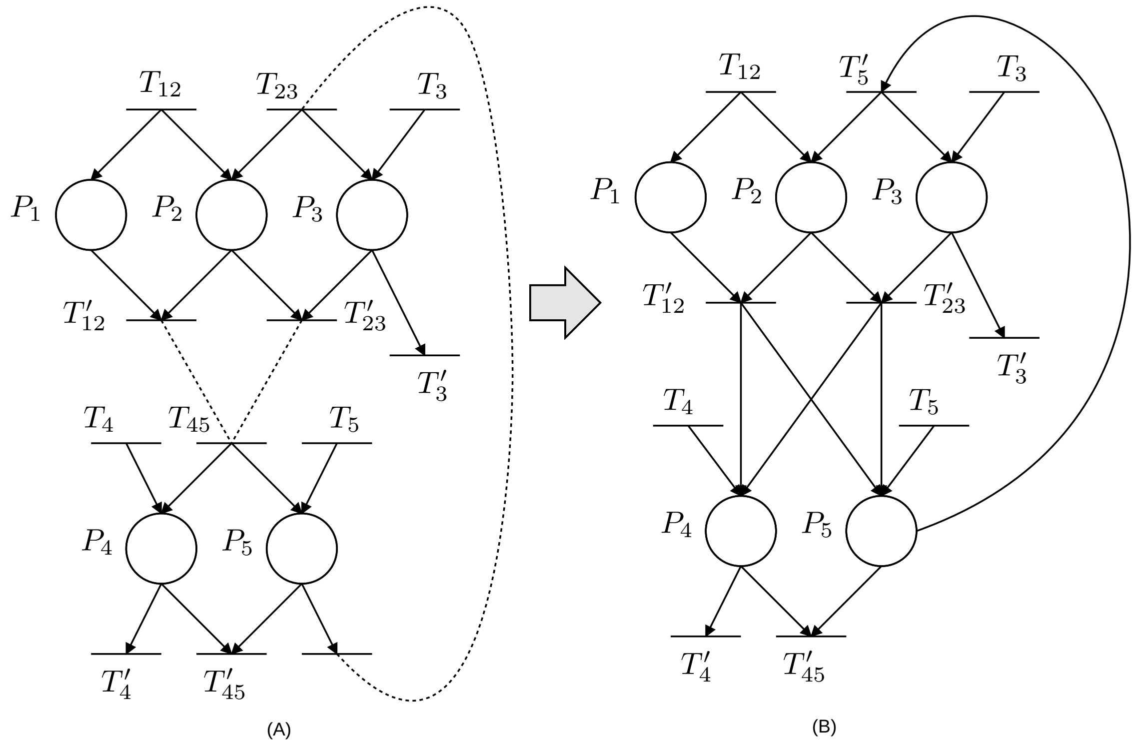

Fig. 6. Net considered in Example 2 of composition of BBs. (A) BBs to be composed. (B) Resulting net.

cubic computational complexity; a minor modification, e.g., composition with a small BB, would require a fresh analysis, in contrast to the present methodology, which would only require one application of RCAT.

POSPNs exhibit the important property that they can be decomposed into BBs. Let us consider the following relation between places: $P_i \sim P_j$ if there exists at least one transition T such that $\mathbf{I}(T)$ has both i-th and j-th positive components. Note that the relation $\sim$ is symmetric and reflexive but it is not transitive, as is shown by the counterexample of Figure 7, where $P_1 \sim P_2$, $P_2 \sim P_3$, but $P_1 \sim P_3$ is false. An

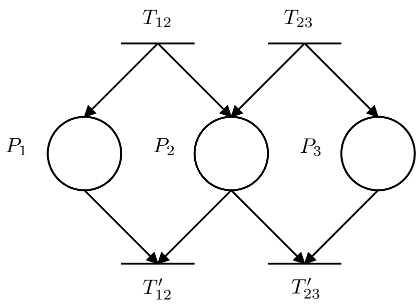

Fig. 7. Example of relations among transitions.

equivalence relation that induces a partition of the places in the net is the transitive closure $\sim'$ of $\sim$, defined by $P_i \sim' P_j$ if $P_i \sim P_j$ or if there exists $P_k \in \mathcal{P}$ such that $P_i \sim P_k$ and $P_k \sim' P_j$. Each equivalence class of $\sim'$ (forming a partition) consists of the places of a BB, as required.

Algorithm 1 identifies the BBs that form a given POSPN – supplied as its input. The output is in the form of a partition of the set of places, each class of which contains the places of a BB. The complexity of the algorithm is $\mathcal{O}(|\mathcal{P}| \cdot |\mathcal{A}|)$.

Input: POSPN structure: $\mathcal{P}, \mathcal{A}, \mathcal{T}$
/* Input: places, arcs and transitions */
Output: Building blocks: $\mathcal{P}_i$
/* P_i is a subset of $\mathcal{P}$ such that $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ with $i \neq j$ */
/* $\bigcup_i \mathcal{P}_i = \mathcal{P}$ */
/* The places in $\mathcal{P}_i$ belongs to the same building block i */
/* Identify the building blocks in a SPN */
foreach $t \in \mathcal{T}$ **do**
  /* $i-1$ denotes the number of classes identified */
  $i := 1$;
  /* Consider every output arc */
  $\mathcal{P}_t = \{P \in \mathcal{P}$ such that $(t, P) \in \mathcal{A}\}$;
  if *exists* $P \in \mathcal{P}_t$ *and* $j > 0$ *such that* $P \in \mathcal{P}_j$ **then**
    $\mathcal{P}_j := \mathcal{P}_j \cup \mathcal{P}_t$;
  end
  else
    /* Create a new class */
    $\mathcal{P}_i := \mathcal{P}_t$;
    $i := i + 1$;
  end
end

Algorithm 1: Algorithm which identifies the BBs of a SPN

Once a POSPN has been decomposed into a set of BBs, the rate equations (12) shown below have to be solved. x_i represents the rate of $T_i \in \mathcal{T}$ in the BB in which it is an input transition. By RCAT, this rate is the reversed rate of T_i in the BB in which T_i is an output transition. Using Corollary 1, x_i is therefore equal to the sum of the rates of the transitions whose output vector is $\mathbf{I}(T_i)$, weighted by the probability of T_i being the transition to fire out of all the transitions with the same input vector.

$$x_i = \left(\sum_{j:\mathbf{O}(T_j)=\mathbf{I}(T_i)} x_j \right) p_i, \tag{12}$$

where $x_j = \chi_j$ if $\mathbf{I}(T_j) = \mathbf{0}$ and $p_i = x_i / (\sum_{k:\mathbf{I}(T_k)=\mathbf{I}(T_i)} x_k)$.

Example 3: We now consider the example presented in [22], where it was shown that the product-form condition depends on the transition rates by applying Kelly's results on reversibility [18]. Here, we show that a routine application of our approach gives the expression for the rate condition straightforwardly and without the need to first establish the reversibility of the process. The net and its decomposition into BBs is shown in Figure 8. The rate equations are simply $x_1 = x_2$, $x_5 = x_6$ and $x_4 = x_3$. By applying Theorem 2, we find the product-form condition for the left hand BB (with places P_1 and P_3) to be $x_6\chi_1\chi_3 = x_2 x_4 \chi_5$ and, analogously, for the right hand BB, $x_5\chi_2\chi_4 = x_1 x_3 \chi_6$. The rate condition $\chi_2\chi_4\chi_5 = \chi_1\chi_3\chi_6$ follows mechanically. Observe also that the rate condition arises since both the BBs are complete and that it is sufficient to apply a local modification, e.g. in only the right hand BB (with P_2 and P_4), to obtain a product-form. For instance, a net with a rate-independent product-form condition would certainly be obtained if T_3 were to feed a new place P_5 instead of P_4 and P_5 were also the sole input place of T_4.

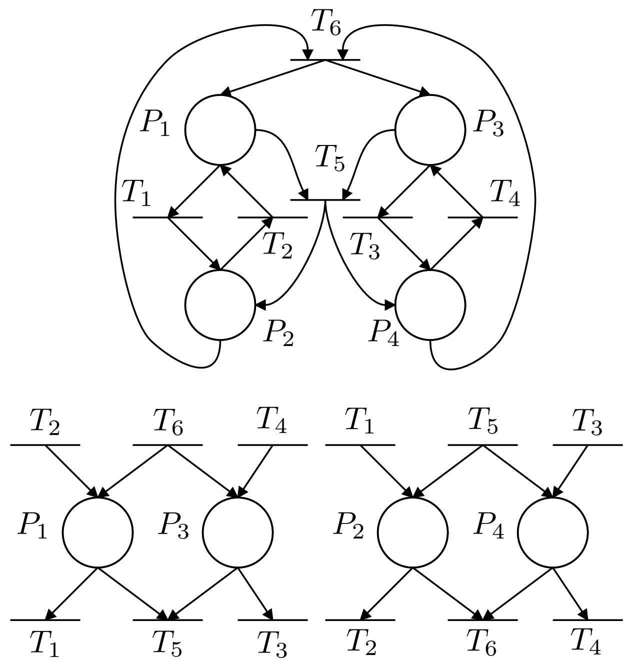

Fig. 8. SPN with rate-dependent product-form taken from [22].

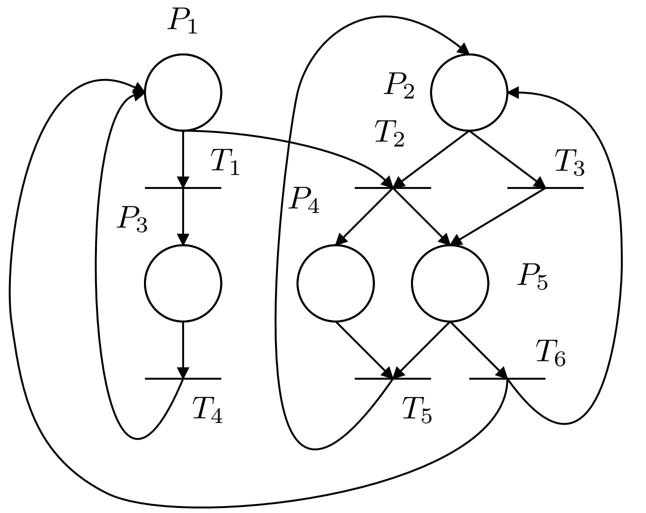

Fig. 9. The POSPN of Example 4 given as input.

Example 4: One of the examples considered in [6] is now solved by our method. The SPN structure is shown in Figure 9. Algorithm 1 generates 3 BBs, as shown in Figure 10.

We apply MARCAT to the following BBs provided by the algorithm:

BB 1 (Figure 10 (1)): The product-form is:

$$\left(\frac{x_4}{\chi_1}\right)^{m_1} \left(\frac{x_5}{\chi_3}\right)^{m_2}$$

under the condition $x_6\chi_1\chi_3 = \chi_2 x_4 x_5$. Applying the rate equations of RCAT, we find $x_1 = \overline{\chi_1} = x_4$, $x_2 = \overline{\chi_2} = x_6$ and $x_3 = \overline{\chi_3} = x_5$, by the result in the proof of Theorem 2 or just using the above product-form of BB 1.

BB 2 (Figure 10 (2)): This block unconditionally has the product-form:

$$\left(\frac{\chi_6 x_2}{x_3\chi_5}\right)^{m_4} \left(\frac{x_3}{\chi_6}\right)^{m_5},$$

where $x_5 = x_2$ and $x_6 = x_3$, similarly to BB 1.

BB 3 (Figure 10 (3)): Even more simply, this block is equivalent to an M/M/1 queue and unconditionally has the product-form $(x_1/\chi_4)^{m_3}$, where $x_4 = x_1$.

The above equations are easily solved to give $x_2 = x_3 = x_5 = x_6$, $x_1 = x_4 = \chi_1\chi_3/\chi_2$, so that the product-forms for BBs $1-3$ become

$$\left(\frac{\chi_3}{\chi_2}\right)^{m_1} \left(\frac{x_5}{\chi_3}\right)^{m_2}, \quad \left(\frac{\chi_6}{\chi_5}\right)^{m_4} \left(\frac{x_3}{\chi_6}\right)^{m_5}, \quad \left(\frac{\chi_1\chi_3}{\chi_2\chi_4}\right)^{m_3}$$

respectively. Hence, we get a product-form that may be written as:

$$x_2^{m_2+m_5} \left(\frac{\chi_3}{\chi_2}\right)^{m_1} \left(\frac{1}{\chi_3}\right)^{m_2} \left(\frac{\chi_6}{\chi_5}\right)^{m_4} \left(\frac{1}{\chi_6}\right)^{m_5} \left(\frac{\chi_1\chi_3}{\chi_2\chi_4}\right)^{m_3}.$$

There is only a unique solution for x_2, x_3, x_5, x_6 up to a multiplicative constant, but since the sum of the numbers of tokens at places P_2 and P_5 is fixed (as may be confirmed by inspection of the original net), the multiplicative constant is arbitrary. We choose $x_2 = x_3 = x_5 = x_6 = \chi_3$ to obtain the product-form:

$$\pi(\mathbf{m}) \propto \left(\frac{\chi_3}{\chi_2}\right)^{m_1} \left(\frac{\chi_1\chi_3}{\chi_2\chi_4}\right)^{m_3} \left(\frac{\chi_6}{\chi_5}\right)^{m_4} \left(\frac{\chi_3}{\chi_6}\right)^{m_5},$$

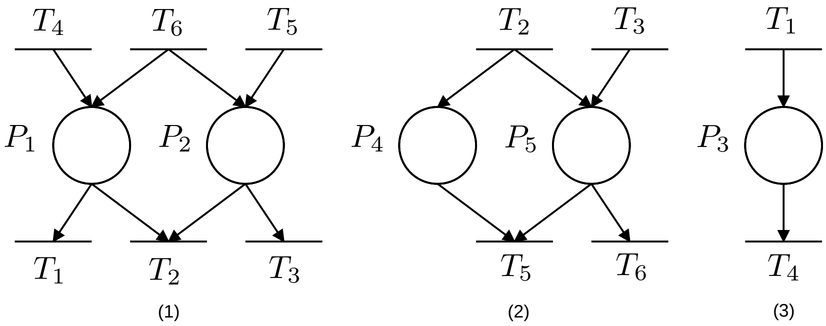

Fig. 10. Decomposition of the POSPN into BBs 1 – 3 of the net of Figure 9.

which reproduces the result of [6].

Example 5: Next we analyse one of the examples presented in [9]. Their SPN structure is shown in Figure 11. In the original paper it was studied using the HT approach [16], [6].

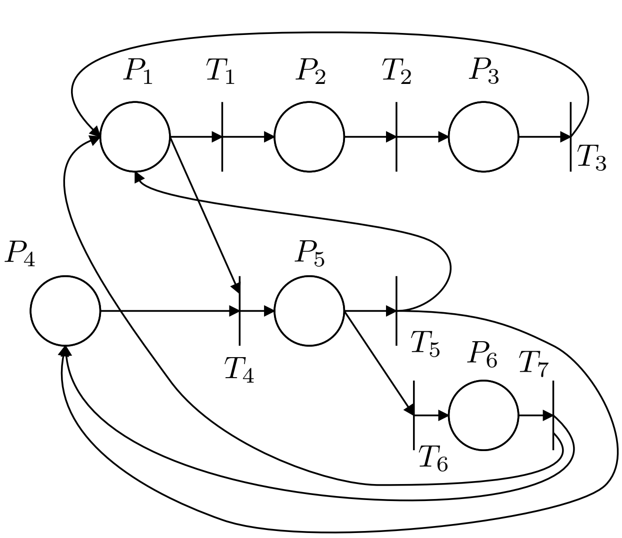

Fig. 11. Net studied in Example 5 (Sereno's example).

We first identify the BBs shown in Figure 12 using Algorithm 1.

We focus on BBs 1 and 4. BB 1 unconditionally has the product-form $(x_3/\chi_1)^{m_1}[((x_5+x_7)\chi_1)/(\chi_4 x_3)]^{m_4}$. The other BBs are simple M/M/1 queues. We study BB 4 because of the presence of T_5 and T_6. Obviously, in this case we have $x_5+x_6=x_4$, but we have to use the rate equation (in RCAT) for x_5 and x_6. As in the previous example (but much more simply in this case because BB 4 is an M/M/1 queue), we obtain $x_6=\chi_6/(\chi_6+\chi_5)x_4$ and $x_5=\chi_5/(\chi_6+\chi_5)x_4$. The (independent) system of equations for the x_i is therefore:

$$\begin{cases} x_1 = x_2 = x_3 \\ x_7 = x_6 \\ x_5 = \frac{\chi_5}{\chi_5+\chi_6} x_4 \\ x_6 = \frac{\chi_6}{\chi_5+\chi_6} x_4 \end{cases} .$$

This linear system is underdetermined and we have one solution for $x_1=x_2=x_3$ and one solution for $x_4, x_5, x_6 = x_7$, each unique up to a multiplicative constant which is arbitrary

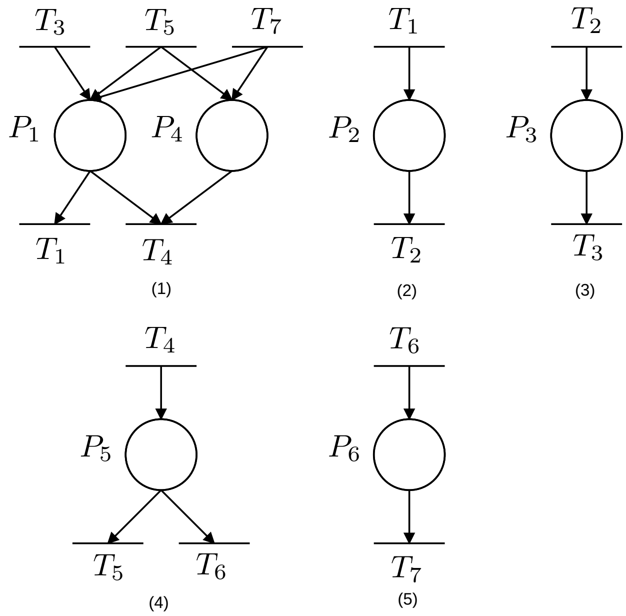

Fig. 12. Decomposition into BBs 1 – 5 of the net of Figure 11.

in each case. We obtain the product-form:

$$\left(\frac{x_3}{\chi_1}\right)^{m_1}\left(\frac{x_4\chi_1}{\chi_4 x_3}\right)^{m_4}\left(\frac{x_1}{\chi_2}\right)^{m_2}\left(\frac{x_2}{\chi_3}\right)^{m_3} \left(\frac{x_4}{\chi_5+\chi_6}\right)^{m_5}\left(\frac{x_6}{\chi_7}\right)^{m_6}$$

which, when normalised, gives the same probability function for *any* choices of x_1 and x_4; in fact we will choose $x_4=\chi_4$ and $x_1=\chi_1$ to get the simplest unnormalised form.

The product-form may be written

$$x_1^{m_1+m_2+m_3-m_4} x_4^{m_4+m_5+m_6}\left(\frac{1}{\chi_1}\right)^{m_1}\left(\frac{\chi_1}{\chi_4}\right)^{m_4} \cdot\left(\frac{1}{\chi_2}\right)^{m_2}\left(\frac{1}{\chi_3}\right)^{m_3}\left(\frac{1}{\chi_5+\chi_6}\right)^{m_5}\left(\frac{\chi_6}{(\chi_5+\chi_6)\chi_7}\right)^{m_6}.$$

As in the previous example, inspection of the net reveals that $m_4+m_5+m_6$ is fixed, so that x_4 is arbitrary; we choose $x_4=\chi_4$. Further inspection verifies that $m_1+m_2+m_3-m_4$ is also constant; transitions either leave both $m_1+m_2+m_3$ and m_4 fixed or increment both m_1 and m_4 by one. Hence, x_1 is also arbitrary and we choose $x_1=\chi_1$.

These choices for x_1 and x_4 lead to the unnormalised product-form:

$$\pi(\mathbf{m}) \propto \left(\frac{\chi_1}{\chi_2}\right)^{m_2}\left(\frac{\chi_1}{\chi_3}\right)^{m_3}\left(\frac{\chi_4}{\chi_5+\chi_6}\right)^{m_5} \cdot \left(\frac{\chi_4\chi_6}{(\chi_5+\chi_6)\chi_7}\right)^{m_6}.$$

It is important to note that there is no need to consider the dynamic properties of the net – our analysis of the rate equations *implies* dynamic properties such as we have just deduced by visual inspection, in an attempt to aid intuition.

Finally, the following proposition establishes a relationship between the MARCAT rate equations and the solution of the traffic processes of the HT approach.

Proposition 1: All transitions in a POSPN, for which the rate equations of MARCAT have a solution, are covered by a closed support T-invariant.

Proof: The proof is based on a result of Sereno [27], which shows that a SPN that admits an invariant measure on the traffic processes defined in [16], is covered by closed support T-invariants. Here, we simply observe that MARCAT's rate equations (12) are equivalent to the system of equations defined in [16] for the invariant measure of the traffic processes. ♠

C. Modular composition of POSPNs

Consider a POSPN in which there is a set of transitions $\mathcal{T}_I \subseteq \mathcal{T}$ whose input vectors are the null vector, and the symmetrical set of transitions $\mathcal{T}_O$ whose output vectors are the null vector. In this case we can interpret the POSPN as an open model, where the input transitions model the arrivals to the system, and the output transitions model the departures. In this section, we investigate product-form properties of a *composition of POSPN*s, by which we mean a model in which a subset of the input transitions of a block becomes output transitions of another block. Basically, the firing of an output transition corresponds to the firing of the corresponding input transition.

We first observe that Proposition 1 extends to networks of POSPNs, using the same proof. We state this as a corollary to that proposition.

Corollary 2: All transitions in a network of POSPNs, for which the rate equations of MARCAT have a solution, are covered by a closed support T-invariant.

In the underlying Markov processes of the POSPNs, we use passive actions to label state transitions corresponding to the firing of the input transitions and active actions for the state transitions corresponding to the firing of the output transitions. Note that an appropriate choice of labels allows one to represent the possibility of connecting more than one output transitions to the same input transition.

Proposition 2: A network of MARCAT-product-form POSPNs, defined as above, satisfies the conditions of MARCAT.

Proof: MARCAT's conditions are verified in the special case handled by the original RCAT as follows:

1) The passive actions are enabled in every state of the model because they correspond to the input transitions.
2) The second condition holds if, for all output transitions T_o, for every state $\mathbf{m}$ of the POSPN, there exists a state $\mathbf{m}'$ such that the state-transition $\mathbf{m}' \to \mathbf{m}$ takes place due to the firing of T_o. Suppose that, in the network of POSPNs, the rate equations of MARCAT have a solution; since each POSPN's local rate equations have a solution (by hypothesis), this allows each POSPN to be considered as a primitive component in the application of MARCAT. Thus, all the network's transitions are covered by a closed support T-invariant by Corollary 2. In the minimal closed support T-invariant covering T_o there is an input transition and is it easy to find a firing sequence that starts with the firing of an input transition and finishes with the firing of T_o, which takes the net from a marking $\mathbf{m}$ back to itself.
3) We have already proved that the reversed rates of output transitions are constant. ♠

Thus we have shown that, if we have a set of POSPNs that each admit a product-form solution, then we can compose them to obtain a new model that can be solved efficiently by RCAT. Note that this approach enhances the modularity of that proposed by Coleman *et al.* Moreover, our approach allows a hierarchical modelling technique – in fact, a product-form network of POSPNs still satisfies RCAT and so can be used as a block in larger networks.

The following section provides a detailed example.

D. Product-form in a network of POSPNs

As an example, we consider the composition of two POSPNs, which are depicted in Figures 13 and 14, along with their decomposition into building blocks. Considering the former, the BB with P_1 and P_2 is unconditionally in product-form:

$$\left(\frac{x_8+\chi_1}{\chi_5+\chi_4}\frac{\chi_3}{\chi_2}\right)^{m_1}\left(\frac{\chi_2}{\chi_3}\right)^{m_2}.$$

The reversed rates are $x_5 = (x_8+\chi_1)/(\chi_4+\chi_5)\chi_5$ and $x_4 = (x_8+\chi_1)/(\chi_4+\chi_5)\chi_4$. The other blocks are equivalent to M/M/1 queues: $x_3 = \chi_2$, $x_6 = x_5 = x_8$, $x_4 = x_7$. The solution of the equations for x_i is $x_6 = x_5 = x_8 = \frac{\chi_1\chi_5}{\chi_4}$, $x_4 = x_7 = \chi_1$. Therefore, the steady-state solution for the first POSPN model (BLOCK 1) is:

$$\pi_1(\mathbf{m}) \propto \left(\frac{\chi_1\chi_3}{\chi_4\chi_2}\right)^{m_1}\left(\frac{\chi_2}{\chi_3}\right)^{m_2}\left(\frac{\chi_1\chi_5}{\chi_4\chi_6}\right)^{m_3} \cdot \left(\frac{\chi_1\chi_5}{\chi_4\chi_8}\right)^{m_4}\left(\frac{\chi_1}{\chi_7}\right)^{m_5}. \quad (13)$$

The input transition set is $\mathcal{T}_I^{(1)} = \{T_1, T_2\}$ and the output transition set is $\mathcal{T}_O^{(1)} = \{T_7, T_3\}$. Note that minimal closed support T-invariants covering T_3 and T_7 are $(0,1,1,0,0,0,0,0)^T$ and $(1,0,0,1,0,0,1,0)^T$, respectively.

Now we consider the open POSPN depicted in Figure 14, in which the input transition set is $\mathcal{T}_I^{(2)} = \{T_1, T_2\}$ and the output transition set is $\mathcal{T}_O^{(2)} = \{T_6, T_8\}$. In this case there are 4 BBs, 3 of which are trivial. The block with places P_1, P_2, P_3 is unconditionally in product-form. Its steady-state probabilities are:

$$\left(\frac{\chi_1\chi_4}{\chi_3 x_7}\right)^{m_1}\left(\frac{x_7}{\chi_4}\right)^{m_2}\left(\frac{\chi_2\chi_4}{\chi_5 x_7}\right)^{m_3}.$$

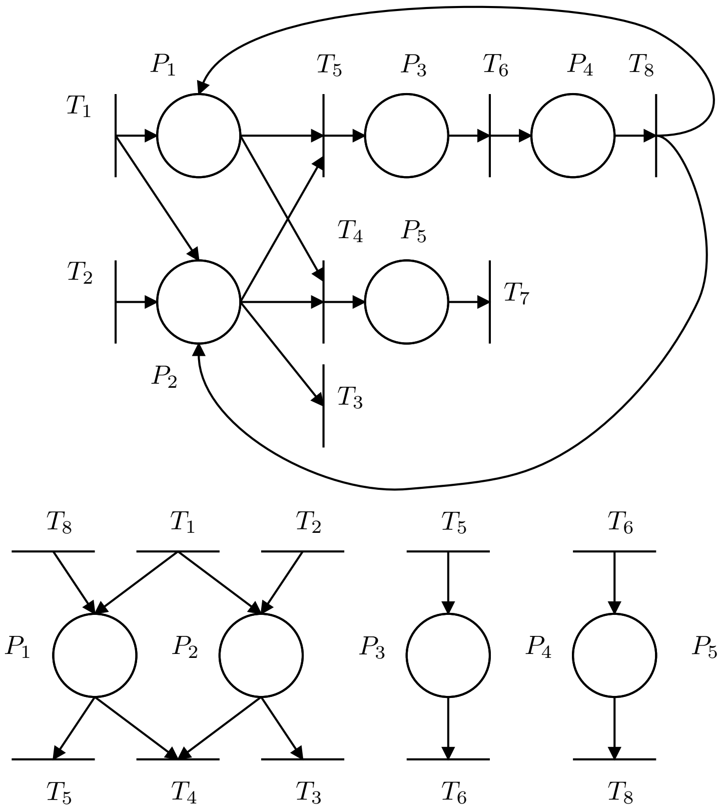

Fig. 13. BLOCK1: POSPN model and its BBs.

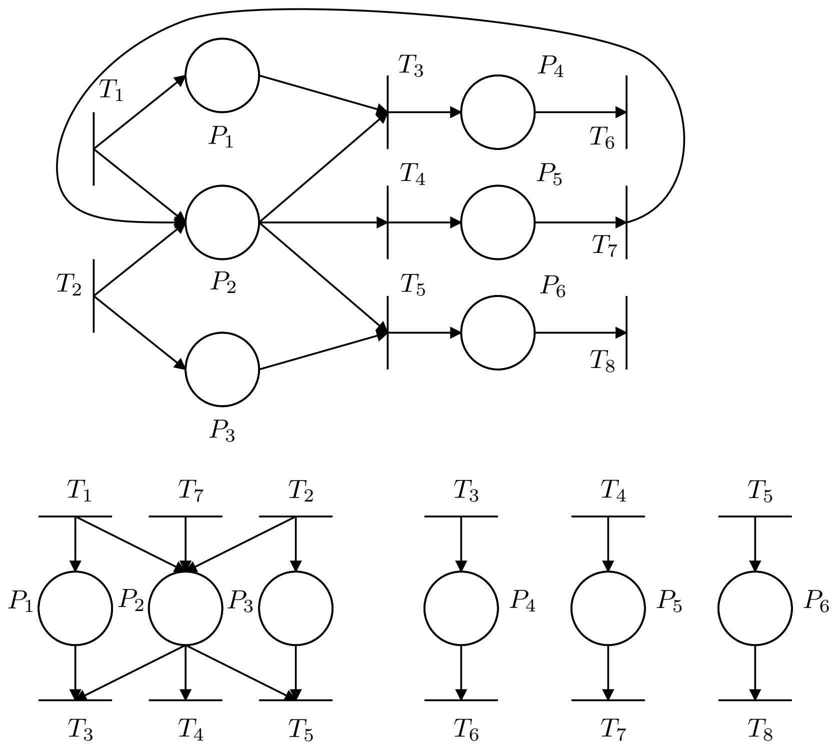

Fig. 14. BLOCK2: POSPN model and its BBs.

The reversed rates are given by $x_3 = \lambda_1$, $x_4 = x_7$ and $x_5 = \chi_2$. The analysis of the BB with P_5 gives the condition $x_7 = x_4$. We can choose $x_4 = x_7 = \chi_4$, since the rate equations are underdetermined, and so the product-form solution can be obtained straightforwardly as:

$$\pi_2(\mathbf{m}) \propto \left(\frac{\chi_1}{\chi_3}\right)^{m_1} \left(\frac{\chi_2}{\chi_5}\right)^{m_3} \left(\frac{\chi_1}{\chi_6}\right)^{m_4} \left(\frac{\chi_4}{\chi_7}\right)^{m_5} \cdot \left(\frac{\chi_2}{\chi_8}\right)^{m_6}. \quad (14)$$

We now want to compose these models, as shown in Figure 15. In order to avoid conflicts on the transition, place and rate names, we use the superscript (n) with $n = 1, 2$ in order to distinguish the models (e.g. $P_4^{(1)}$ and $P_4^{(2)}$ are the places 4 in models 1 and 2, respectively).

The rates of transitions $T_7^{(1)}$ and $T_3^{(1)}$ are unknown with respect to model BLOCK2, and similarly $T_6^{(2)}$ and $T_8^{(2)}$ with respect to BLOCK1. Therefore, we rewrite solutions (13) and (14) as:

$$\pi_1(\mathbf{m}^{(1)}) \propto \left(\frac{x_6^{(2)}\chi_3^{(1)}}{\chi_4^{(1)}x_8^{(2)}}\right)^{m_1^{(1)}} \left(\frac{x_8^{(2)}}{\chi_3^{(1)}}\right)^{m_2^{(1)}} \cdot \left(\frac{x_6^{(2)}\chi_5^{(1)}}{\chi_4^{(1)}\chi_6^{(1)}}\right)^{m_3^{(1)}} \left(\frac{x_6^{(2)}\chi_5^{(1)}}{\chi_4^{(1)}\chi_8^{(1)}}\right)^{m_4^{(1)}} \left(\frac{x_6^{(2)}}{\chi_7^{(1)}}\right)^{m_5^{(1)}}$$

and

$$\pi_2(\mathbf{m}^{(2)}) \propto \left(\frac{x_7^{(1)}}{\chi_3^{(2)}}\right)^{m_1^{(2)}} \left(\frac{\chi_2^{(2)}}{\chi_5^{(2)}}\right)^{m_3^{(2)}} \cdot \left(\frac{x_7^{(1)}}{\chi_6^{(2)}}\right)^{m_4^{(2)}} \left(\frac{\chi_4^{(2)}}{\chi_7^{(2)}}\right)^{m_5^{(2)}} \left(\frac{\chi_2^{(2)}}{\chi_8^{(2)}}\right)^{m_6^{(2)}}$$

The traffic equations for the composed net then become:

$$\begin{cases} x_6^{(2)} = x_7^{(1)}/\chi_6^{(2)}\chi_6^{(2)} = x_7^{(1)} \\ x_8^{(2)} = \chi_2^{(2)}/\chi_8^{(2)}\chi_8^{(2)} = \chi_2^{(2)} \\ x_7^{(1)} = x_6^{(2)}/\chi_7^{(1)}\chi_7^{(1)} = x_6^{(2)} \end{cases}.$$

These have solution $x_6^{(2)} = x_7^{(1)} = \chi_3^{(2)}$ and $x_8^{(2)} = \chi_2^{(2)}$, so that the stationary probability function is:

$$\pi(\mathbf{m}^{(1)};\mathbf{m}^{(2)}) \propto \left(\frac{\chi_3^{(2)}\chi_3^{(1)}}{\chi_4^{(1)}\chi_2^{(2)}}\right)^{m_1^{(1)}} \left(\frac{\chi_2^{(2)}}{\chi_3^{(1)}}\right)^{m_2^{(1)}} \cdot \left(\frac{\chi_3^{(2)}\chi_5^{(1)}}{\chi_4^{(1)}\chi_6^{(1)}}\right)^{m_3^{(1)}} \left(\frac{\chi_3^{(2)}\chi_5^{(1)}}{\chi_4^{(1)}\chi_8^{(1)}}\right)^{m_4^{(1)}} \left(\frac{\chi_3^{(2)}}{\chi_7^{(1)}}\right)^{m_5^{(1)}} \cdot \left(\frac{\chi_2^{(2)}}{\chi_5^{(2)}}\right)^{m_3^{(2)}} \left(\frac{\chi_3^{(2)}}{\chi_6^{(2)}}\right)^{m_4^{(2)}} \left(\frac{\chi_4^{(2)}}{\chi_7^{(2)}}\right)^{m_5^{(2)}} \cdot \left(\frac{\chi_2^{(2)}}{\chi_8^{(2)}}\right)^{m_6^{(2)}} \quad (15)$$

V. CONCRETE APPLICATION

We now give an example of the application of the preceding methodology in the performance evaluation and optimisation of a software architecture with fork and join constructs. Such constructs are important in many multi-tasking applications; for example, RAID[3] storage control systems.

[3]Redundant arrays of independent disks.

1) Model description: Consider the net of Figure 16. The

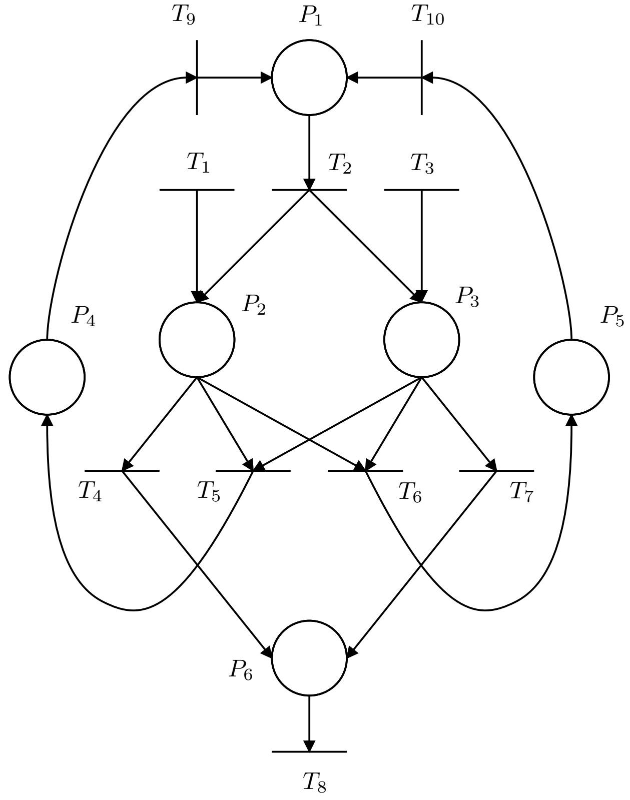

Fig. 16. Example of Section V.

system consists of two servers (e.g., databases) and two different classes of customers. Servers are exclusive, in the sense that each of them can work on only one of the classes. Customers arrive at the system according to independent Poisson processes (transitions T_1 and T_3 for classes 1 and 2, respectively) and wait in their own queues (represented by respective places P_2 and P_3) for service (represented by transitions T_4 and T_7). Once a customer has been served, it goes to a queue (place P_6), where it waits to be sent out of the system (transition T_8) – regardless of its class. The simultaneous service to customers of different classes may conflict such that two kinds of system-failures (represented by T_5 and T_6) may occur, which are subsequently repaired according to two different recovery procedures (T_9 and T_{10}). Once a recovery procedure is completed, customers are queued again for service (at respective places P_2 and P_3). Although a simplified example, it is easy to see how more realistic, multi-class systems, with interactions amongst the classes, can be constructed in this way, and then possibly composed with each other to form larger systems.

2) Analysis: We first derive a product-form solution for the net and show that it holds independently of the transitions' rates – since the POSPN is open, rate-dependent conditions may arise in general. The throughput and mean response time of the net is then obtained as a function of the transition rates. The decomposition into BBs is depicted in Figure 17

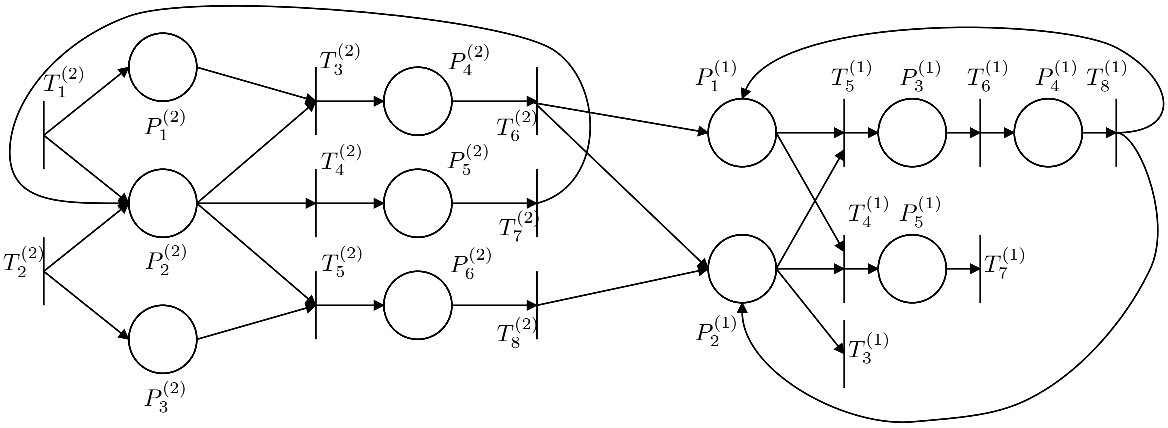

Fig. 15. BLOCK1 composed with BLOCK2.

and generates the following rate-equations:

$$\left\{\begin{array}{l} x_2 = x_9 + x_{10} \\ x_9 = x_5 \\ x_{10} = x_6 \\ x_5 = x_2 \frac{\chi_5}{\chi_5+\chi_6} \\ x_6 = x_2 \frac{\chi_6}{\chi_5+\chi_6} \\ x_4 = \chi_1 \\ x_7 = \chi_3 \\ x_2\chi_4\chi_7 = (\chi_5+\chi_6)\chi_1\chi_3 \end{array}\right. \tag{16}$$

the last equation being the product-form condition for BB1. These equations have a unique solution for any set of $\chi_i > 0$, $i = 1, \ldots, 10$:

$$\begin{array}{lll} x_2 = \frac{\chi_1\chi_3(\chi_5+\chi_6)}{\chi_4\chi_7}, & x_4 = \chi_1, & x_5 = \frac{\chi_1\chi_3\chi_5}{\chi_4\chi_7} \\ x_6 = \frac{\chi_1\chi_3\chi_6}{\chi_4\chi_7}, & x_7 = \chi_3, & \chi_9 = \frac{\chi_1\chi_3\chi_5}{\chi_4\chi_7}, \\ x_{10} = \frac{\chi_1\chi_3\chi_6}{\chi_4\chi_7} & & \end{array}.$$

The steady-state, product-form probabilities (under appropriate stability conditions) are then:

$$\pi(m_1, \ldots, m_6) = G \prod_{i=1}^{6} \rho_i^{m_i},$$

where G is a normalising constant and the terms ρ_i are:

$$\left\{\begin{array}{l} \rho_1 = \chi_1\chi_3(\chi_5+\chi_6)/(\chi_2\chi_4\chi_7) \\ \rho_2 = \chi_1/\chi_4 \\ \rho_3 = \chi_3/\chi_7 \\ \rho_4 = \chi_1\chi_3\chi_5/(\chi_4\chi_7\chi_9) \\ \rho_5 = \chi_1\chi_3\chi_6/(\chi_{10}\chi_4\chi_7) \\ \rho_6 = (\chi_1+\chi_3)/\chi_8 \end{array}\right. \tag{17}$$

In this example we observe that the state space is $\mathbb{N}^6$. Consequently, the normalising constant can be computed easily as the product of the normalising constants for each place i, considered as a standard M/M/1 queue with load-factor ρ_i:

$$G = \prod_{i=1}^{6}(1-\rho_i).$$

Note that, in equilibrium, the throughput of transition T_8 (the net throughput), as one might expect from a net invariant analysis, is $\chi_1 + \chi_3$, i.e. the sum of the customer arrival rates.

Let the mean number of tokens in place P_i, at equilibrium, be denoted by $\overline{N}_i$. Since the net has product-form, each place can be seen as an independent M/M/1 queue, so that $\overline{N}_i = \rho_i/(1-\rho_i)$. Observe that the net conserves the number of customers, meaning that each customer arriving from the outside eventually leaves the system, and there is no internal creation of customers. This may be proved formally by considering a closure of the model, where the input place of transitions T_1 and T_3 is P_6 and T_8 is not present. It is easy to see that a weighted sum of the numbers of tokens in each place is constant, since there exists a P-invariant $\mathbf{P} = (2, 1, 1, 2, 2, 1)$. Roughly speaking, this P-invariant implies that

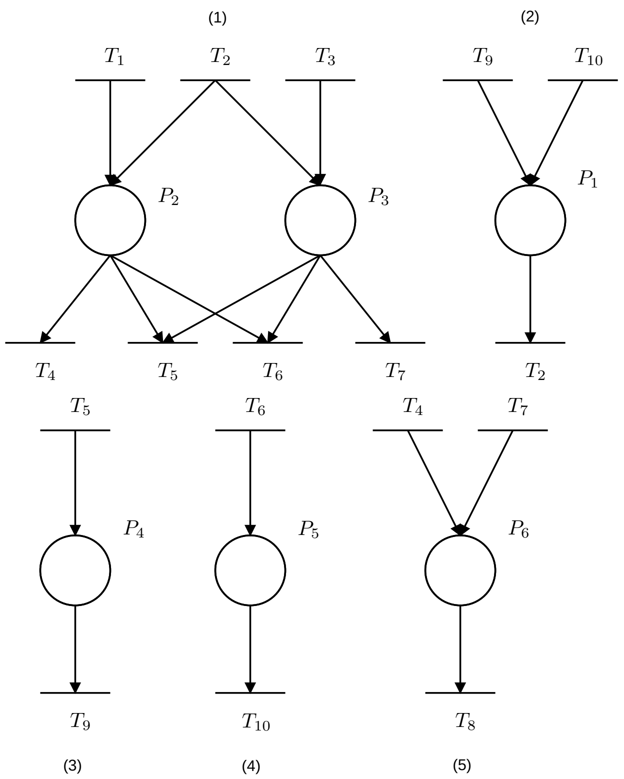

Fig. 17. Decomposition into BB of the net of Figure 16.

Transition	Rate
T_1	2.0
T_2	1.7
T_3	1.0
T_4	2.2
T_5	1.0
T_6	1.5
T_7	1.8
T_8	4.0
T_9	.60
T_{10}	1.0

TABLE I
TRANSITION RATES IN THE NET OF FIGURE 16.

every token in any of P_1, P_4, P_5 should be interpreted as a pair of customers that, in our model, represents the two customers (one from each class) that initiated a failure. The mean number of all customers (of either class) in the network is therefore $\overline{N} = \sum_{i=1}^{6} P_i \overline{N}_i$, where P_i is the ith component of the P-invariant $\mathbf{P}$, and the mean system response time is $\overline{R} = \overline{N}/(\chi_1 + \chi_3)$ by Little's result.

The mean response times for the two classes separately are, similarly:

$$\overline{R}^{(1)} = \overline{N}^{(1)}/\chi_1 \quad \text{and} \quad \overline{R}^{(2)} = \overline{N}^{(2)}/\chi_3,$$

where

$$\overline{N}^{(1)} = \overline{N}_1 + \overline{N}_2 + \overline{N}_4 + \overline{N}_5 + \overline{N}_6\chi_1/(\chi_1 + \chi_3)$$
$$\overline{N}^{(2)} = \overline{N}_1 + \overline{N}_3 + \overline{N}_4 + \overline{N}_5 + \overline{N}_6\chi_3/(\chi_1 + \chi_3)$$

As a numerical instance, we use the transition rates shown in Table I. With this parameterisation, we obtain the following mean values for the performance indices: $\overline{R}^{(1)} = 11.666, R^{(2)} = 13.581$ and $\overline{N} = 36.912$. Assume now that we would like to improve the recovery procedures in order to reduce the system response time. Suppose that the rates of transitions T_9 and T_{10} are linear functions of a scalar factor $\alpha > 0$, i.e.

$$\chi_9(\alpha) = \chi_9\alpha, \quad \chi_{10}(\alpha) = \chi_{10}\alpha.$$

We can easily express the mean response times as functions of α and obtain the plots shown in Figure 18.

3) Further considerations: To conclude this example, we make some remarks about the consequences of the compositionality properties of our methodology. According to SPN semantics, we considered the delay distributions of transitions T_9 and T_{10} as independent, negative exponential random variables with rates χ_9 and χ_{10}, respectively. However, observe that if a processor-sharing discipline were assumed for the tokens in P_4 and P_5, then the exponential distributions may be replaced by Coxians. This is due to the fact that P_4 with T_9 (and hence P_5 with T_{10}) may be interpreted as a single class BCMP queueing station [3], and this is known to be quasi-reversible [25]. In [23] it is proved that the combination of RCAT and quasi-reversible models gives a product-form solution. As a consequence, the repair time may have any distribution with rational Laplace transform, which can approximate any distribution arbitrarily closely. This changes the distribution of the performance indices but not their average values.

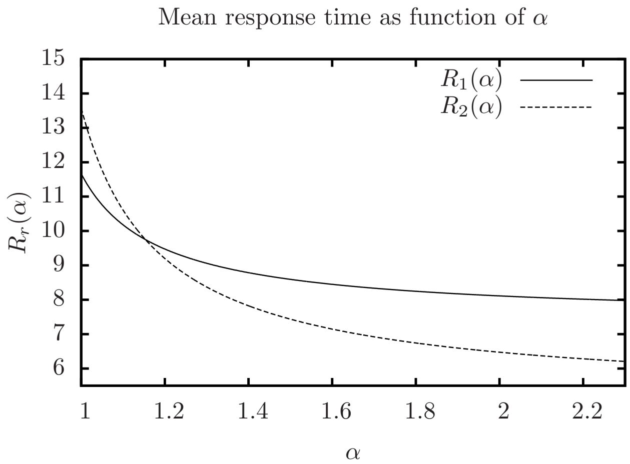

Fig. 18. Mean response time for class 1 and 2 customers as function of α.

VI. CONCLUSIONS

The constructive method described in this paper holds the promise of producing the first practical performance tools to provide efficient quantitative solutions in the design of a diverse range of systems and networks under one umbrella. Exact (separable) solutions can be identified when they can be shown to exist, and approximate separable solutions computed otherwise. The problem with existing generalist approaches has been that specialised efficient solutions have been sacrificed for generality, but this is not the case here. Conversely, in specialised models like queueing networks, the range of solvable problems is very narrow.

Although product-form SPN models may require strict conditions, possibly even on the transition rates, their importance in the practice of (software) performance engineering should not be underestimated. In fact, several works, such as [15], [24], [5], have shown that product-form models may be used as efficient approximations of non-product-form systems, providing bounds on or approximate values for meaningful performance indices.

There are various ways in which the potential of this work can be exploited:

- The techniques described should be automated to provide a genuine, practical methodology for compositional SPNs for the first time;
- In our analysis, all transitions have constant rates. State-dependency in rates of the type considered in [16], [6], [22] can be achieved by applying the result presented in [2], but more general forms of state-dependent firing rates may be amenable to our methodology;
- An important performance metric that has not been considered in the present contribution (apart from its mean value) is *response time*, or *passage time* between two (sets of) markings. Complementary to RCAT, a new approach to finding joint node-sojourn time probability distributions in Markovian networks is to consider *reversed processes*. The idea is to split a path into (at least) two segments and analyse the reversed process in the first segment and the forwards process in the rest.

Our compositional analysis of SPNs is conducive to this approach.

References

[1] Balbo, G., Bruell, S. C., and Sereno, M. Product form solution for Generalized Stochastic Petri Nets. *IEEE Trans. on Software Eng. 28*, 10 (2002), 915–932.

[2] Balsamo, S., and Marin, A. Product-form solutions for models with joint-state dependent transition rates. In *LNCS 6148, Proc. of Int. Conf. ASMTA* (Cardiff, UK, 2010), Springer, pp. 87–101.

[3] Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, F. G. Open, closed, and mixed networks of queues with different classes of customers. *J. ACM 22*, 2 (1975), 248–260.

[4] Boucherie, R. J. A characterisation of independence for competing Markov chains with applications to stochastic Petri nets. *IEEE Trans. on Software Eng. 20*, 7 (1994), 536–544.

[5] Buchholz, P. Product form approximations for communicating Markov processes. *Perf. Eval. 67*, 9 (2010), 797 – 815. Special Issue: QEST 2008.

[6] Coleman, J. L., Henderson, W., and Taylor, P. G. Product form equilibrium distributions and a convolution algorithm for Stochastic Petri nets. *Perf. Eval. 26*, 3 (1996), 159–180.

[7] Distefano, S., Scarpa, M., and Puliafito, A. From UML to Petri Nets: the PCM-Based Methodology. *IEEE Trans. on Software Eng. 99*, PrePrints (2010).

[8] Gómez-Martínez, E., and Merseguer, J. ArgoSPE: Model-based software performance engineering. In *LNCS 4024, 27th Int. Conf. on Applications and Theory of Petri Nets and Other Models of Concurrency* (Turku, Finland, June, 2006), Springer, pp. 401–410.

[9] Haddad, S., Moreaux, P., Sereno, M., and Silva, M. Product-form and stochastic Petri nets: a structural approach. *Perf. Eval. 59*, 4 (2005), 313–336.

[10] Harrison, P. G. Turning back time in Markovian process algebra. *Theoretical Computer Science 290*, 3 (January 2003), 1947–1986.

[11] Harrison, P. G. Compositional reversed Markov processes, with applications to G-networks. *Perf. Eval. 57*, 3 (2004), 379–408.

[12] Harrison, P. G. Reversed processes, product forms and a non-product form. *Linear Algebra and Its Applications 386* (July 2004), 359–381.

[13] Harrison, P. G. Product-forms and functional rates. *Perf. Eval. 66*, 11 (2009), 660–664.

[14] Harrison, P. G., and Lee, T. T. Separable equilibrium state probabilities via time reversal in Markovian process algebra. *Theoretical Computer Science 346*, 1 (2005), 161–182.

[15] Harrison, P. G., and Vigliotti, M. G. Response time distributions and network perturbation into product-form. In *VALUETOOLS '09: Proc. of the Fourth Int. ICST Conf. on Perf. Eval. Meth. and Tools* (Pisa, Italy, 2009), ICST, pp. 1–9.

[16] Henderson, W., Lucic, D., and Taylor, P. G. A net level performance analysis of Stochastic Petri Nets. *J. Austral. Math. Soc. Ser. B 31* (1989), 176–187.

[17] Hillston, J. *A Compositional Approach to Performance Modelling*. PhD thesis, Department of Computer Science, University of Edinburgh, 1994.

[18] Kelly, F. *Reversibility and stochastic networks*. Wiley, New York, 1979.

[19] King, P., and Pooley, R. Using UML to Derive Stochastic Petri Net Models. In *Dep. of Computer Science, Univ. of Bristol* (1999), pp. 45–56.

[20] Lazar, A. A., and Robertazzi, T. G. Markovian Petri Net Protocols with Product Form Solution. *Perf. Eval. 12*, 1 (1991), 67–77.

[21] López-Grao, J. P., Merseguer, J., and Campos, J. From UML activity diagrams to Stochastic Petri nets: application to software performance engineering. In *Proc. of the 4th int. workshop on Soft. and Perf.* (Redwood Shores, California, 2004), WOSP '04, ACM, pp. 25–36.

[22] Mairesse, J., and Nguyen, H.-T. Deficiency Zero Petri Nets and Product Form. In *Proc. of the 30th Int. Conf. on App. and Theory of Petri Nets* (Paris, France, 2009), PETRI NETS '09, Springer-Verlag, pp. 103–122.

[23] Marin, A., and Vigliotti, M. G. A general result for deriving product-form solutions of markovian models. In *Proc. of First Joint WOSP/SIPEW Int. Conf. on Perf. Eng.* (San Josè, CA, USA, 2010), ACM, pp. 165–176.

[24] Marin, A., and Vigliotti, M. G. On product-form approximations of cooperating stochastic models. In *Proc. of 25th Int. Symp. on Computer and Information Sciences* (The Royal Society, London, 2010), LNEE, Springer, pp. 65–70.

[25] Muntz, R. R. Poisson departure processes and queueing networks. Tech. Rep. IBM Research Report RC4145, Yorktown Heights, New York, 1972.

[26] Reibman, A., and Trivedi, K. Numerical transient analysis of Markov models. *Comp. and Op. Res. 15*, 1 (1988), 19–36.

[27] Sereno, M., and Boucherie, R. On closed support T-invariant and the traffic equations. *J. Appl. Probab. 35*, 2 (1988), 473–481.

[28] Steward, W. J. *Probability, Markov Chains, Queues, and Simulation*. Princeton University Press, UK, 2009.

Appendix

In this appendix, we point out, by means of a simple example, an application of the result presented in this paper, which aims to illustrate that Theorem 2 is certainly not obvious and gives, even in simple cases, results that seem to contradict the literature on product-form SPNs. Consider the BB of Figure 3. Theorem 2 states that it is in product-form if $\lambda_{12}\mu_1\mu_2 = \mu_{12}\lambda_1\lambda_2$. Consider now the BB depicted in Figure 19. If we apply the composition of BBs described

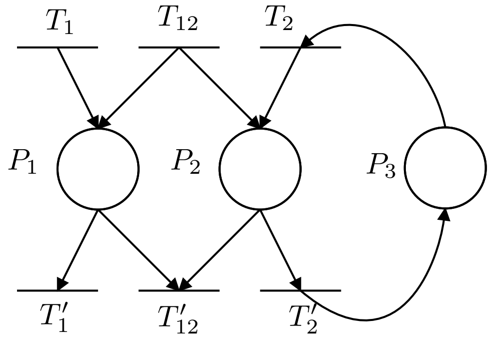

Fig. 19. Example of Appendix.

in Section IV, we obtain the unconditional product-form:

$$\pi(m_1, m_2, m_3) \propto \left(\frac{\lambda_1}{\mu_1}\right)^{m_1} \left(\frac{\lambda_{12}\mu_1}{\lambda_1\mu_{12}}\right)^{m_2} \left(\frac{\lambda_{12}\mu_1\mu_2}{\lambda_1\mu_{12}\lambda_2}\right)^{m_3}.$$

Hence, one may suspect that the condition required for the BB of Figure 3 is not really needed. However, this argument is flawed. Suppose that the BB has a product-form solution:

$$\pi(m_1, m_2) \propto \rho_1^{m_1}\rho_2^{m_2}, \tag{18}$$

for some $\rho_i > 0$, $i = 1, 2$. The global balance equation for state $(0, 0)$ is:

$$\pi(0,0)(\lambda_1 + \lambda_2 + \lambda_{12}) = \pi(1,0)\mu_1 + \pi(0,1)\mu_2 + \pi(1,1)\mu_{12},$$

which leads to the following constraint on ρ_i:

$$\lambda_1 + \lambda_2 + \lambda_{12} = \rho_1\mu_1 + \rho_2\mu_2 + \rho_1\rho_2\mu_{12}. \tag{19}$$

Using the global balance equation for state $(1, 0)$, we derive the following condition:

$$\lambda_1 + \lambda_2 + \lambda_{12} + \mu_1 = \rho_1\mu_1 + \rho_1\rho_2\mu_{12} + \rho_2\mu_2 + \frac{\lambda_1}{\rho_1},$$

which, by equation (19) gives $\rho_1 = \lambda_1/\mu_1$. Symmetrically, we obtain $\rho_2 = \lambda_2/\mu_2$. Finally, consider the global balance equation at state $(1, 1)$:

$$\pi(1,1)(\lambda_1+\lambda_2+\lambda_{12}+\mu_1+\mu_2+\mu_{12}) = \pi(0,0)\lambda_{12}+\pi(0,1)\lambda_1 + \pi(1,0)\lambda_2 + \pi(2,1)\mu_1 + \pi_{1,2}\mu_2 + \pi(2,2)\mu_{12},$$

This is satisfied if and only if

$$\rho_1\rho_2 = \frac{\lambda_{12}}{\mu_{12}} \Rightarrow \lambda_{12}\mu_1\mu_2 = \mu_{12}\lambda_1\lambda_2.$$

Hence, the BB of Figure 3 does *not* have a product-form solution of the form (18) unless some constraints on the rates are satisfied. The net of Figure 19 has the same transitions as the net of Figure 3 and is also open. However, it yields an unconditional product-form. This follows straightforwardly from the application of our result, which gives the expression for the product-form in a few lines. Although we omit the details, one may check that the given product-form solution does hold for arbitrary transition rates, by solving the global balance equations.