Structured Solution of Stochastic DSSP Systems*

J. Campos, M. Silva

Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
jcampos, silva@posta.unizar.es

S. Donatelli

Dip.to di Informatica
Università di Torino
susi@di.unito.it

Abstract

Deterministically Synchronized Sequential Processes (DSSP) are essentially states machines that communicate, may be in complex forms but under some restricted patterns, through buffer places; their definition is compositional by nature. This paper considers the problem of exploiting this compositionality to generate the state space and to find the steady state probabilities of a stochastic extension of DSSP in a net-driven, efficient way.

Essentially, we give an expresion of an auxiliary matrix, **G***, which is a supermatrix of the infinitesimal generator of a DSSP.* **G** *is a tensor algebra [9] expression of matrices of the size of the components for which it is possible to numerically solve the characteristic equation* $\boldsymbol{\pi} \cdot \mathbf{G} = \mathbf{0}$*, without the need to explicitly compute* **G***. Therefore, we obtain a method that computes the steady state solution of a DSSP without ever explicitly computing and storing its infinitesimal generator, and therefore without computing and storing the reachability graph of the system.*

1 Introduction and motivations

Generalized Stochastic Petri Nets (GSPN's) [1] are a well-known interpreted extension of autonomous Petri net (PN) models that allow the use of the formalism to deal with performance aspects in the design of complex concurrent systems, in addition to the PN's capability for the validation of functional properties. In general, numerical solution of the embedded Continuous Time Markov Chain (CTMC) must be performed to get exact performance indices. Product-form expressions and efficient algorithms for the computation of the steady-state distribution are known only for some particular classes of GSPN's. At this point, in general the *state explosion problem* makes intractable the evaluation of large systems due to the storage cost for the infinitesimal generator matrix and to the time complexity of solution algorithms.

Net-driven techniques deal with the reduction of both memory and time complexity of solution algorithms using structure information from the net model. Contributions already present in the literature that can be included under this epigraph are, for instance, the techniques for removing immediate transitions from the original net [6] or the exploitation of symmetries in the analysis of Stochastic Well-Formed Coloured Nets [7].

A different topic in which net structure can be useful to drive the solution technique is the *tensor algebra* approach to express the infinitesimal generator $\mathbf{Q}$ of a GSPN in terms of $\mathbf{Q}_i$ matrices coming from smaller systems and to implement the solution of the characteristic equation $\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$ without computing $\mathbf{Q}$ [2, 10, 11, 4, 13].

In this paper we concentrate on the Petri net subclass of *Deterministically Synchronized Sequential Processes* (DSSP) [21, 17]. They are obtained by the application of a simple modular design principle: several functional units (in this case, sequential processes modelled with *state machines*, SM's) execute concurrently and cooperate using asynchronous communication by message passing through a set of buffers (places with possibly weighted input and output arcs). Buffers are destination private (i.e., they go to a single sequential process). Thus competition among functional units is prevented. Moreover, buffers do not represent side conditions in conflicts of the functional units (i.e., choices are free, local in the SM's). DSSP are interesting enough from a modelling point of view since they include, in a controlled way, features for the modelling of *concurrency*, *decisions*, *synchronization*, *blocking*, and *bulk movements of jobs*. On the other hand, a well-developed theory exists for the analysis of qualitative behaviour of these systems [21, 17, 18, 19] that allows, in particular, to check necessary and suf-

*This work has been developed within the project HCM CT94-0452 (MATCH) of the European Union. The work of J. Campos and M. Silva is also supported by project PRONTIC 94-0242 of the Spanish CICYT.

ficient conditions for *finiteness* and *ergodicity* of the embedded CTMC. Moreover, the compositional definition of DSSP leads in a quite natural way to the construction of smaller modules that can be used in the tensor algebra approach for expressing the infinitesimal generator matrix.

From the components we can build, for a given DSSP model, a set of auxiliary, simpler, models. We show in this paper that the reachability set (RS) of a DSSP is included in a set PS that is the union of the Cartesian product of subsets of the reachability sets of the auxiliary models. From the reachability graphs of the auxiliary models, it is instead possible to build an auxiliary matrix $\mathbf{G}$ which is a supermatrix of the infinitesimal generator of a DSSP. $\mathbf{G}$ is a tensor algebra expression of matrices of the size of the components for which it is possible to numerically solve the characteristic equation $\boldsymbol{\pi} \cdot \mathbf{G} = \mathbf{0}$, without the need to explicitly compute $\mathbf{G}$. By showing that the $\boldsymbol{\pi}$ vector is the steady state solution of the DSSP, for an appropriate choice of the initial probability vector, we obtain a method that computes the steady state solution of a DSSP without ever explicitly computing and storing its infinitesimal generator, and therefore without computing and storing the reachability graph of the system.

Ergodicity is one weak point of the solution approaches based on compositionality. Ergodicity can be proved in stochastic bounded Petri nets (under Markovian interpretation) by checking whether the reachability graph has home states. This approach is definitely not possible in this case, since the complete reachability graph is never built, but for this particular class of systems we show that it is possible to prove ergodicity without computing the reachability graph and checking the existance of a home space.

The paper is organised as follows: Section 2 reviews the definition and main results for DSSP needed here, and proposes a stochastic interpretation similar to the GSPN one. The method is formally presented in Section 3 and in Section 4. All the concepts are explained on a simple running example. Finally, some concluding remarks are stressed in Section 5.

2 DSSP and their stochastic interpretation

In this section, we recall the definition and several important properties, and we introduce a reduction technique for stochastic DSSP, that allows to build a set of auxiliary models.

We assume that the reader is familiar with concepts and notation of P/T nets [20], moreover we adopt the classical notation of writing in bold vectors and matrices, and of using square brackets to indicate elements of vectors and matrices (in this paper sub-indices are never used for this aim).

2.1 DSSP definition and properties

State machines (SM's) are ordinary PN's such that every transition has only one input and only one output place ($\forall t \in T$: $|{}^\bullet t| = |t^\bullet| = 1$). SM's allow the modelling of sequences, decisions (or conflicts), and reentrance (when they are marked with more than one token) but not synchronization. SM's marked with a single token model sequential processes.

Deterministically Synchronized Sequential Processes (DSSP) [21] are used for the modelling and analysis of distributed systems composed by sequential processes communicating through output-private buffers. Each sequential process is modelled by a *safe* (1–bounded) strongly connected SM. The communication among them is described by *buffers* (places) which contain *products/messages* (tokens), that are produced by some processes and consumed by others. Each buffer is *output-private* in the sense that it is an input place of only one SM.

Definition 1 *A PN system,* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0} \rangle$, *is a* Deterministically Synchronized Sequential Processes *system, or simply a DSSP, if:*
1. $P_i \cap P_j = \emptyset$, $T_i \cap T_j = \emptyset$, $P_i \cap B = \emptyset$, $\forall i, j \in \{1, \ldots, K\}, i \neq j$;
2. $\langle \mathcal{SM}_i, \mathbf{m_0}_i \rangle = \langle P_i, T_i, \mathbf{Pre}_i, \mathbf{Post}_i, \mathbf{m_0}_i \rangle$, $\forall i \in \{1, \ldots, K\}$ *is a strongly connected and 1–bounded state machine (where* $\mathbf{Pre}_i$,$\mathbf{Post}_i$, *and* $\mathbf{m_0}_i$ *are the restrictions of* $\mathbf{Pre}$, $\mathbf{Post}$, *and* $\mathbf{m_0}$ *to* P_i *and* T_i*);*
3. *The set* B *of buffers is such that* $\forall b \in B$*:*
(a) $|{}^\bullet b| \geq 1$ *and* $|b^\bullet| \geq 1$,
(b) $\exists i \in \{1, \ldots, K\}$ *such that* $b^\bullet \subset T_i$,
(c) $\forall p \in P_1 \cup \ldots \cup P_K$*:* $t, t' \in p^\bullet \Rightarrow \mathbf{Pre}[b, t] = \mathbf{Pre}[b, t']$.
Transitions belonging to the set $\mathrm{TI} = {}^\bullet B \cup B^\bullet$ *are called* interface transitions. *The remaining ones* ($(T_1 \cup \ldots \cup T_K) \setminus \mathrm{TI}$) *are called* internal transitions.

An example of DSSP with two buffers and two state machines is depicted in Figure 1. Some basic properties for DSSP (as defined here) can be found in [21].

In this paper we consider a stochastic extension of DSSP: independent, exponentially distributed random variables are associated to the firing of transitions as in classical *stochastic PN's* [14].

Definition 2 *A stochastic DSSP (St-DSSP) is a pair* $\langle \mathcal{S}, w \rangle = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}, w \rangle$*, where* $\mathcal{S}$ *is a DSSP*

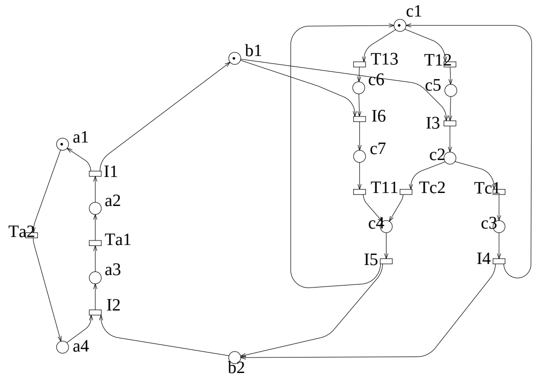

Figure 1: A DSSP system.

and $w : T \to \mathbb{R}^+$ *is a positive real function that associates to each transition* $t \in T$ *an exponentially distributed firing time with rate* $w(t)$.

The reader should notice that considering stochastic PN's without *immediate* transitions instead of *Generalized Stochastic PN's* [1] is not a major constraint in the case of DSSP. Immediate transitions that are not in TI (transitions internal to the SM's) can be eliminated with the technique in [6], and the resulting net (without immediate transitions) would still be a DSSP; transitions that are in TI (interface transitions) instead cannot in general be eliminated without violating some of the DSSP rules, but we must remember that transitions in TI are either *behaviourally conflict-free* or in *equal conflict* (same pre-incidence function) therefore the use of immediate transitions instead of timed would not add a significant contribution from a modelling point of view.

Our goal is to compute the steady-state probability distribution of markings of St-DSSP. In order for the computation to make sense, the embedded CTMC should be *ergodic*. Additionally, we assume the system to be *bounded*, thus the CTMC being finite, to be able to compute the limit distribution. Obviously, we are interested in *deadlock-free* models otherwise all transitions of our models will have null throughput. In the next property several interesting results on DSSP are summarized. All these results will be useful to check the functional properties that must be validated *before* performance analysis.

Property 3 *The following statements hold:*
1. *Let* $\mathcal{N}$ *be a (general) PN and* $\mathbf{C}$ *its* $n \times m$ *incidence matrix. Then* $\mathcal{N}$ *is structurally bounded iff* $\exists \mathbf{y} \in \mathbb{N}^n : \mathbf{y} \geq \mathbf{0} \wedge \mathbf{y} \cdot \mathbf{C} \leq \mathbf{0}$.
2. [17] *Let* $\mathcal{N}$ *be the net of a DSSP.* $\mathcal{N}$ *is structurally live and structurally bounded iff* $\mathcal{N}$ *is consistent (i.e.,* $\exists \mathbf{x} > \mathbf{0}, \mathbf{C} \cdot \mathbf{x} = \mathbf{0}$*), conservative (i.e.,* $\exists \mathbf{y} > \mathbf{0}, \mathbf{y} \cdot \mathbf{C} = \mathbf{0}$*) and* $\text{rank}(\mathbf{C}) = |\mathcal{E}| - 1$*, where* $\mathcal{E}$ *is the set of equal conflict sets of* $\mathcal{N}$ *(two transitions* t *and* t' *belong to the same equal conflict set iff* $\mathbf{Pre}[\cdot, t] = \mathbf{Pre}[\cdot, t'] \neq \mathbf{0}$*).*
3. [21] *Let* $\mathcal{S}$ *be a bounded, strongly connected DSSP. Then* $\mathcal{S}$ *is live iff it is deadlock-free.*
4. [18] *Let* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *be a DSSP. Deadlock-freeness of* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is characterised by the existance of no integer solution for a integer (linear) programming problem that can be derived from* $\mathcal{N}$ *and* $\mathbf{m_0}$.
5. [19] *Let* $\mathcal{S}$ *be a live DSSP. Then* $\mathcal{S}$ *is bounded iff it is structurally bounded.*
6. [18] *Live and bounded DSSP have home states.*

From a practical point of view, the previous results on DSSP could be used in the following way:
Step 1: check structural boundedness using statement *1*;
Step 2: check the characterization for structural liveness and structural boundedness using statement *2*;
Step 3: check deadlock-freeness (thus, by statement *3*, liveness) using statement *4*.

The answer to all the previous questions is affirmative if and only if the St-DSSP is live and bounded (statement *5*) thus it has home state (statement *6*), therefore the embedded CTMC is finite and ergodic, moreover, all transitions have non-null throughput, provided all firing rates are non-null.

2.2 A reduction rule

In this subsection we define a reduction rule for the internal behaviour of SM's of a DSSP (*internal behaviour* means firing of internal transitions). Using that reduction, we introduce a decomposition of DSSP into a collection of *low level systems* and a *basic skeleton* (along the lines used in [5]). In each low level system, only one of the different SM's of the original system is kept while the internal behaviours of the others are reduced. In the basic skeleton, the internal behaviours of all the SM's are reduced. In the next sections, we shall use the low level systems and basic skeleton introduced here for an structured construction of the reachability set of the original model and also for an structured computation of its steady-state probabilities.

A place is *implicit* [8] (under interleaving semantics) if it can be deleted without changing the firing sequences. Since we are considering a Markovian interpretation of PN's, implicit places are redundant in the following sense: the reachability graph (thus the embedded CTMC) of a system is preserved if implicit places are removed (only the state representation is slightly modified, by deleting from the states the information on the number of tokens in the implicit places).

In the following definition an *extended system* $\mathcal{ES}$ is built from the original system $\mathcal{S}$, by adding some sets of implicit places, H_i (standing for "high-level places" that are computed as non-negative summations of subsets of places of the original model), thus the reachability graphs of $\mathcal{S}$ and $\mathcal{ES}$ are identical (only the redundant marking of implicit places is added in the state representation of $\mathcal{ES}$).

Definition 4 *Let* $\mathcal{S} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}\rangle$ *be a DSSP with* $P = P_1 \cup \ldots \cup P_K \cup B$ *and* $T = T_1 \cup \ldots \cup T_K$*. Let* R *be the equivalence relation defined on all the places in* $P \setminus B$ *by:* $\langle p_1^i, p_2^i\rangle \in \mathrm{R}$ *for* $p_1^i, p_2^i \in P_i$ *iff there exists a non-directed path* np *in* $\mathcal{SM}_i$ *from* p_1^i *to* p_2^i *such that* $\mathrm{np} \cap \mathrm{TI} = \emptyset$ *(i.e., containing only internal transitions). Let* $[p_1^i], \ldots, [p_{r(i)}^i]$ *be the different equivalence classes defined in* P_i *by the relation* R*,* $i = 1, \ldots, K$*. The extended system of* $\mathcal{S}$ *is* $\mathcal{ES} = \langle P_{\mathcal{ES}}, T_{\mathcal{ES}}, \mathbf{Pre}_{\mathcal{ES}}, \mathbf{Post}_{\mathcal{ES}}, \mathbf{m_0}^{\mathcal{ES}}\rangle$*, defined as:*
i) $P_{\mathcal{ES}} = P \cup H_1 \cup \ldots \cup H_K$*, with* $H_i = \{h_1^i, \ldots, h_{r(i)}^i\}$*,* $i = 1, \ldots, K$*, such that* $H_i \cap P = \emptyset$ *and* $H_i \cap H_j = \emptyset$*, for* $j \neq i$*,* $i = 1, \ldots, K$*;*
ii) $T_{\mathcal{ES}} = T$*;*
iii) $\mathbf{Pre}_{\mathcal{ES}}|_{P \times T} = \mathbf{Pre}$*;* $\mathbf{Post}_{\mathcal{ES}}|_{P \times T} = \mathbf{Post}$*;*
iv) $\mathbf{Pre}[h_j^i, t] = \sum_{p \in [p_j^i]} \mathbf{Pre}[p, t]$ *and* $\mathbf{Post}[h_j^i, t] = \sum_{p \in [p_j^i]} \mathbf{Post}[p, t]$*, for* $t \in \mathrm{TI} \cap T_i$*;* $\mathbf{Pre}[h_j^i, t] = 0$ *and* $\mathbf{Post}[h_j^i, t] = 0$ *for* $t \notin \mathrm{TI} \cap T_i$*; for* $j = 1, \ldots, r(i)$*, for* $i = 1, \ldots, K$*;*
v) $\mathbf{m_0}^{\mathcal{ES}}[p] = \mathbf{m_0}[p]$*, for all* $p \in P$*;*
vi) $\mathbf{m_0}^{\mathcal{ES}}[h_j^i] = \sum_{p \in [p_j^i]} \mathbf{m_0}[p]$*, for* $j = 1, \ldots, r(i)$*, for* $i = 1, \ldots, K$*.*

As an example, the extended system of the DSSP of Figure 1 is depicted in Figure 2 where place C_{34} summarizes places c_2, c_3, c_4, and c_7, place C_{56} summarizes places c_1, c_5, and c_6, place A_{14} summarizes places a_1 and a_4, and place A_{23} summarizes places a_2 and a_3. In the example, we use the convention of naming implicit places with upper case letters, with an index that is a composition of two of the indices of the implicated places. In the DSSP of the example we have $H_1 = \{A_{14}, A_{23}\}$ and $H_2 = \{C_{34}, C_{56}\}$.

From the extended system, K different *low level systems* $\mathcal{LS}_i$ can be obtained deleting all places in P_j, $j \neq i$, and transitions in $T_j \setminus \mathrm{TI}$, $j \neq i$ $(i = 1, \ldots, K)$.

Definition 5 *Let us define a DSSP* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}\rangle$ *and* $\mathcal{ES}$ *its corresponding extended system.*
i) *The low level system* $\mathcal{LS}_i$ *(*$i = 1, \ldots, K$*) of* $\mathcal{S}$ *is the system obtained from* $\mathcal{ES}$ *deleting all the nodes in* $\bigcup_{j \neq i}(P_j \cup (T_j \setminus \mathrm{TI}))$ *and their adjacent arcs.*

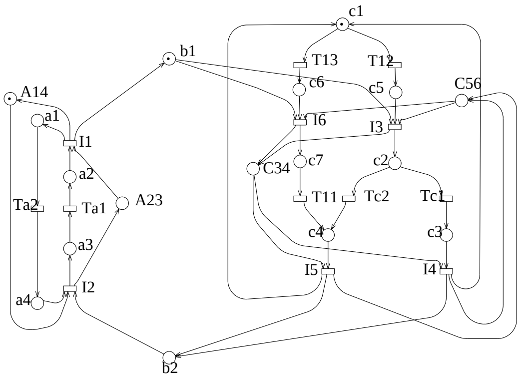

Figure 2: The DSSP example model with additional implicit places.

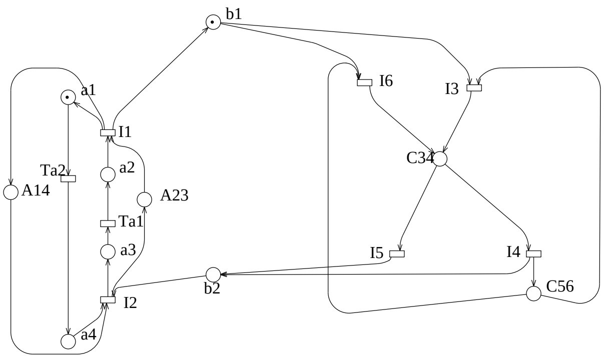

Figure 3: The $\mathcal{LS}_1$ system corresponding to the DSSP of Fig. 1 obtained by reducing all state machines ($\mathcal{SM}_2$ in this case) but $\mathcal{SM}_1$.

ii) *The basic skeleton* $\mathcal{BS}$ *of* $\mathcal{S}$ *is the system obtained from* $\mathcal{ES}$ *deleting all the nodes in* $\bigcup_j (P_j \cup (T_j \setminus \mathrm{TI}))$ *and their adjacent arcs.*

In other words, in each $\mathcal{LS}_i$ all the states machines $\mathcal{SM}_j$, $j \neq i$, are reduced to their interface transitions and to the implicit places that were added in the extended system, while $\mathcal{SM}_i$ is fully preserved. In the basic skeleton all the SM's are reduced in the same way. Figure 3 shows the low level system $\mathcal{LS}_1$ for the DSSP in Figure 1, and Figure 4 shows the low level system $\mathcal{LS}_2$. Figure 5 shows instead the corresponding $\mathcal{BS}$ system (please note that we preserve the names that transitions and places have in $\mathcal{S}$ when they are also present in other systems like $\mathcal{ES}$ and $\mathcal{LS}_i$ $(i = 1, \ldots, K)$).

In general, the reduction technique presented here does not remove but eventually adds new paths between interface transitions (see, for instance, the path from $I6$ to $I4$ that is present in $\mathcal{BS}$ (Figure 5), while it

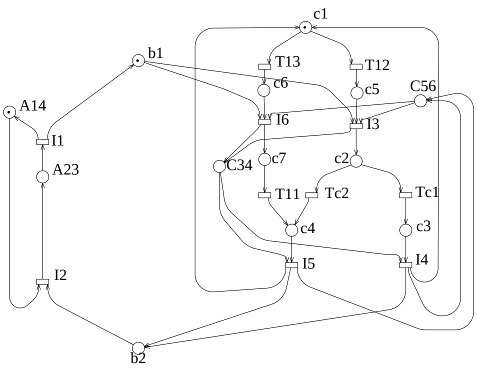

Figure 4: The $\mathcal{LS}_2$ system corresponding to the DSSP of Fig. 1 obtained by reducing all state machines ($\mathcal{SM}_1$ in this case) but $\mathcal{SM}_2$.

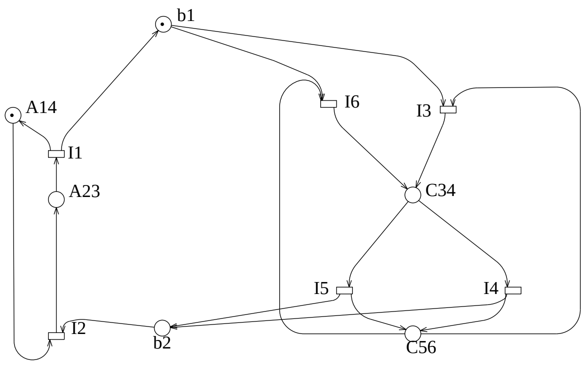

Figure 5: The $\mathcal{BS}$ of the DSSP of Fig. 1.

does not exist in the original model of Figure 1); from this fact, the next property can be proved.

Property 6 *Let us define a DSSP* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0} \rangle$, $\mathcal{LS}_i$ *its low level systems* ($i = 1, \ldots, K$)*,* $\mathcal{BS}$ *its basic skeleton, and* $\mathrm{L}(\mathcal{S})$ *the language of* $\mathcal{S}$*. Then:*
1. $\mathrm{L}(\mathcal{S})|_{T_i \cup \mathrm{TI}} \subseteq \mathrm{L}(\mathcal{LS}_i)$*, for* $i = 1, \ldots, K$*.*
2. $\mathrm{L}(\mathcal{S})|_{\mathrm{TI}} \subseteq \mathrm{L}(\mathcal{BS})$*.*

Proof:
Consider the extended system $\mathcal{ES}$ of $\mathcal{S}$. By definition, all the places in $H_1 \cup \ldots \cup H_K$ are implicit in $\mathcal{ES}$. Therefore $\mathrm{L}(\mathcal{S}) = \mathrm{L}(\mathcal{ES})$. Now, consider the system $\mathcal{LS}_i$ for an arbitrary i. Obviously, $\mathrm{L}(\mathcal{ES})|_{T_i \cup \mathrm{TI}} \subseteq \mathrm{L}(\mathcal{LS}_i)$, and the property follows. The same argument is valid for the basic skeleton (statement *2*). $\diamondsuit$

3 The structured construction of the reachability set

In this section, a result is presented that allows to construct a superset of the reachability set $\mathrm{RS}(\mathcal{S})$ of a DSSP in a structured way. Our approach is closely related to the *asynchronous composition* technique for a tensor algebra solution of queueing networks [3], coloured Petri nets [2, 12], and marked graphs [4]. Before introducing that result, we define a partition on the reachability sets of original ($\mathrm{RS}(\mathcal{S})$), extended ($\mathrm{RS}(\mathcal{ES})$), and low level systems ($\mathrm{RS}(\mathcal{LS}_i)$) according to the projection of the marking on the places of the basic skeleton.

Definition 7 *Let us define a DSSP* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0} \rangle$, $\mathcal{ES}$ *its extended system,* $\mathcal{LS}_i$ *its low level systems* ($i = 1, \ldots, K$)*, and* $\mathcal{BS}$ *its basic skeleton. Then, the following subsets of the reachability sets* $\mathrm{RS}(\mathcal{S})$*,* $\mathrm{RS}(\mathcal{ES})$*, and* $\mathrm{RS}(\mathcal{LS}_i)$ *are defined for each* $\mathbf{z} \in \mathrm{RS}(\mathcal{BS})$*:*

$$\begin{array}{l}
\mathrm{RS}_{\mathbf{z}}(\mathcal{ES}) = \{\mathbf{m} \in \mathrm{RS}(\mathcal{ES}) \mid \mathbf{m}|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}\} \\
\mathrm{RS}_{\mathbf{z}}(\mathcal{S}) = \{\mathbf{m} \in \mathrm{RS}(\mathcal{S}) \mid \exists \mathbf{m}' \in \mathrm{RS}_{\mathbf{z}}(\mathcal{ES}) : \\
\qquad\qquad \mathbf{m}'|_{P_1 \cup \ldots \cup P_K \cup B} = \mathbf{m}\} \\
\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_i) = \{\mathbf{m}_i \in \mathrm{RS}(\mathcal{LS}_i) \mid \mathbf{m}_i|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}\}
\end{array}$$

From the definition, the following obvious partitions are obtained: $\mathrm{RS}(\mathcal{S}) = \biguplus_{\mathbf{z} \in \mathrm{RS}(\mathcal{BS})} \mathrm{RS}_{\mathbf{z}}(\mathcal{S})$, $\mathrm{RS}(\mathcal{ES}) = \biguplus_{\mathbf{z} \in \mathrm{RS}(\mathcal{BS})} \mathrm{RS}_{\mathbf{z}}(\mathcal{ES})$, and $\mathrm{RS}(\mathcal{LS}_i) = \biguplus_{\mathbf{z} \in \mathrm{RS}(\mathcal{BS})} \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_i)$ ($i = 1, \ldots, K$), where symbol $\uplus$ denotes disjoint union of sets.

The following result essentially states that each reachable marking of a DSSP can be expressed as a composition of conveniently selected markings of all the low level systems built from the original model.

Theorem 8 *Let us define a DSSP* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0} \rangle$, $\mathcal{LS}_i$ *its low level systems* ($i = 1, \ldots, K$), *and* $\mathcal{BS}$ *its basic skeleton. Denoting* $\mathrm{PS}_{\mathbf{z}}(\mathcal{S}) = \{\mathbf{z}|_B\} \times \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_1)|_{P_1} \times \cdots \times \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_K)|_{P_K}$, *for each* $\mathbf{z} \in \mathrm{RS}(\mathcal{BS})$, *and* $\mathrm{PS}(\mathcal{S}) = \bigcup_{\mathbf{z} \in \mathrm{RS}(\mathcal{BS})} \mathrm{PS}_{\mathbf{z}}(\mathcal{S})$, *then:*

$$\mathrm{RS}(\mathcal{S}) \subseteq \mathrm{PS}(\mathcal{S}) = \biguplus_{\mathbf{z} \in \mathrm{RS}(\mathcal{BS})} \mathrm{PS}_{\mathbf{z}}(\mathcal{S})$$

(where symbol $\uplus$ *denotes disjoint union of sets). Even more:*

$$\mathrm{RS}_{\mathbf{z}}(\mathcal{S}) \subseteq \mathrm{PS}_{\mathbf{z}}(\mathcal{S})$$

Proof:
First, we prove the inclusion.

Let $\mathbf{m} \in \mathrm{RS}_{\mathbf{z}}(\mathcal{S})$. Then there exists a sequence σ such that $\mathbf{m_0} \xrightarrow{\sigma} \mathbf{m}$. By Property 6, there exist sequences σ_i with $\sigma_i = \sigma|_{T_i \cup \mathrm{TI}}$ ($i = 1, \ldots, K$) and $\sigma_{\mathcal{BS}}$ with $\sigma_{\mathcal{BS}} = \sigma|_{\mathrm{TI}}$ that may be fired in $\mathcal{LS}_i$ ($i = 1, \ldots, K$) and in $\mathcal{BS}$, respectively: $\mathbf{m_0^{\mathit{i}}} \xrightarrow{\sigma_i} \mathbf{m}_i$ ($i = 1, \ldots, K$) and $\mathbf{z_0} \xrightarrow{\sigma_{\mathcal{BS}}} \mathbf{z}$, where $\mathbf{m_0^{\mathit{i}}}$ ($i = 1, \ldots, K$) and $\mathbf{z_0}$ denote the initial markings of $\mathcal{LS}_i$ ($i = 1, \ldots, K$) and $\mathcal{BS}$, respectively.

Then: $\mathbf{m}_i|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}$, thus $\mathbf{m}_i \in \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_i)$, $i = 1, \ldots, K$ (this is because $\mathbf{m_0^{\mathit{i}}}|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z_0}$, $\sigma_i|_{\mathrm{TI}} = \sigma_{\mathcal{BS}}$, and $H_1 \cup \ldots \cup H_K \cup B \subset {}^\bullet\mathrm{TI} \cup \mathrm{TI}^\bullet$).

From analogous arguments, $\mathbf{z}|_B = \mathbf{m}|_B$ (because $\mathbf{z_0}|_B = \mathbf{m_0}|_B$, $\sigma|_{\mathrm{TI}} = \sigma_{\mathcal{BS}}$, and $B \subset {}^\bullet\mathrm{TI} \cup \mathrm{TI}^\bullet$).

Then, $\mathbf{m} = (\mathbf{z}|_B, \mathbf{m}_1|_{P_1}, \ldots, \mathbf{m}_K|_{P_K})$, because $\mathbf{m_0}|_{P_i} = \mathbf{m_0^{\mathit{i}}}|_{P_i}$, $\sigma|_{T_i} = \sigma_i|_{T_i}$, and $P_i \subset {}^\bullet T_i \cup T_i{}^\bullet$, $i = 1, \ldots, K$.

Now, we prove by contradiction that the union is disjoint.

Assume that there exist two different states $\mathbf{z}, \mathbf{z}' \in \mathrm{RS}(\mathcal{BS})$ such that: $(\mathbf{z}|_B, \mathbf{m}_1|_{P_1}, \ldots, \mathbf{m}_K|_{P_K}) = (\mathbf{z}'|_B, \mathbf{m}'_1|_{P_1}, \ldots, \mathbf{m}'_K|_{P_K})$, with $\mathbf{m}_i \in \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_i)$, $\mathbf{m}'_i \in \mathrm{RS}_{\mathbf{z}'}(\mathcal{LS}_i)$ ($i = 1, \ldots, K$).

Then: $\mathbf{z}|_B = \mathbf{z}'|_B$ (obvious), $\mathbf{m}_i|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}$ (by definition of $\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_i)$) and $\mathbf{m}'_i|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}'$ (by definition of $\mathrm{RS}_{\mathbf{z}'}(\mathcal{LS}_i)$).

Since $\mathbf{m}_i|_{P_i} = \mathbf{m}'_i|_{P_i}$ (obvious) and places H_i are implicit in $\mathcal{LS}_i$, $\mathbf{m}_i|_{H_i} = \mathbf{m}'_i|_{H_i}$. Therefore, $\mathbf{z}|_{H_i} = \mathbf{z}'|_{H_i}$, $i = 1, \ldots, K$.

Then, $\mathbf{z} = \mathbf{z}'$ and the result follows. $\diamond$

Concerning the example, Table 1 lists the reachability set of the DSSP of Figure 1, that consists of 26 states, $\mathbf{v}_i$. Table 2 lists instead the reachability sets of $\mathcal{BS}$ ($\mathbf{z}_i$), $\mathcal{LS}_1$ ($\mathbf{x}_j$), and $\mathcal{LS}_2$ ($\mathbf{y}_k$) respectively, where for each state of $\mathcal{LS}_1$ ($\mathcal{LS}_2$) we have indicated the partition $\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_1)$ ($\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_2)$) to which the state

RS of $\mathcal{S}$		RS of $\mathcal{S}$	
$\mathbf{v}_1$	a1, b1, c1	$\mathbf{v}_{14}$	a1, c1, b2
$\mathbf{v}_2$	a1, b1, c6	$\mathbf{v}_{15}$	a4, c3
$\mathbf{v}_3$	a1, b1, c5	$\mathbf{v}_{16}$	a4, c1, b2
$\mathbf{v}_4$	a4, b1, c1	$\mathbf{v}_{17}$	a1, b2, c6
$\mathbf{v}_5$	a1, c7	$\mathbf{v}_{18}$	a1, b2, c5
$\mathbf{v}_6$	a4, b1, c6	$\mathbf{v}_{19}$	a4, b2, c6
$\mathbf{v}_7$	a4, b1, c5	$\mathbf{v}_{20}$	a4, b2, c5
$\mathbf{v}_8$	a1, c2	$\mathbf{v}_{21}$	a3, c1
$\mathbf{v}_9$	a1, c4	$\mathbf{v}_{22}$	a3, c6
$\mathbf{v}_{10}$	a4, c7	$\mathbf{v}_{23}$	a3, c5
$\mathbf{v}_{11}$	a4, c2	$\mathbf{v}_{24}$	a2, c1
$\mathbf{v}_{12}$	a1, c3	$\mathbf{v}_{25}$	a2, c6
$\mathbf{v}_{13}$	a4, c4	$\mathbf{v}_{26}$	a2, c5

Table 1: RS of the DSSP of Figure 1.

RS of $\mathcal{BS}$	
$\mathbf{z}_1$	A14, C56, b1
$\mathbf{z}_2$	A14, C34
$\mathbf{z}_3$	A14, C56, b2
$\mathbf{z}_4$	A23, C56

RS of $\mathcal{LS}_1$		
$\mathbf{x}_1$	a1, b1, C56, A14	$\mathbf{z}_1$
$\mathbf{x}_2$	a4, b1, C56, A14	$\mathbf{z}_1$
$\mathbf{x}_3$	a1, C34, A14	$\mathbf{z}_2$
$\mathbf{x}_4$	a4, C34, A14	$\mathbf{z}_2$
$\mathbf{x}_5$	a1, b2, C56, A14	$\mathbf{z}_3$
$\mathbf{x}_6$	a4, b2, C56, A14	$\mathbf{z}_3$
$\mathbf{x}_7$	a3, C56, A23	$\mathbf{z}_4$
$\mathbf{x}_8$	a2, C56, A23	$\mathbf{z}_4$

RS of $\mathcal{LS}_2$		
$\mathbf{y}_1$	A14, b1, c1, C56	$\mathbf{z}_1$
$\mathbf{y}_2$	A14, b1, c6, C56	$\mathbf{z}_1$
$\mathbf{y}_3$	A14, b1, c5, C56	$\mathbf{z}_1$
$\mathbf{y}_4$	A14, c7, C34	$\mathbf{z}_2$
$\mathbf{y}_5$	A14, c2, C34	$\mathbf{z}_2$
$\mathbf{y}_6$	A14, c4, C34	$\mathbf{z}_2$
$\mathbf{y}_7$	A14, c3, C34	$\mathbf{z}_2$
$\mathbf{y}_8$	A14, b2, c1, C56	$\mathbf{z}_3$
$\mathbf{y}_9$	A14, b2, c6, C56	$\mathbf{z}_3$
$\mathbf{y}_{10}$	A14, b2, c5, C56	$\mathbf{z}_3$
$\mathbf{y}_{11}$	A23, c1, C56	$\mathbf{z}_4$
$\mathbf{y}_{12}$	A23, c6, C56	$\mathbf{z}_4$
$\mathbf{y}_{13}$	A23, c5, C56	$\mathbf{z}_4$

Table 2: RS's of the DSSP of Figures 3, 4, and 5.

belongs (third column). The elements of the partition are identified by the corresponding high level marking in $\mathcal{BS}$. As proved above

$$\mathrm{RS}(\mathcal{S}) \subseteq \biguplus_{\mathbf{z} \in \mathrm{RS}(\mathcal{BS})} \{\mathbf{z}|_B\} \times \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_1)|_{P_1} \times \mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_2)|_{P_2}$$

and in this case we actually have an equality. As an example consider the case of $\mathbf{z} = \mathbf{z}_1$: then the cross product of $\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_1) = \{\mathbf{x}_1, \mathbf{x}_2\}$ (markings of $\mathcal{LS}_1$) and $\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_2) = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$ (markings of $\mathcal{LS}_2$) produces the states of PS: $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_6$, and $\mathbf{v}_7$.

It is important to remember that in general $\mathrm{PS}(\mathcal{S}) \neq \mathrm{RS}(\mathcal{S})$ even for live and bounded DSSP; unreachable states may be present in systems in which there is a global sequentialization that enforce a non local order in the firing of the transitions of a single component.

4 The structured solution of St-DSSP

In this section we define the infinitesimal generator of a St-DSSP system in terms of matrices derived from the infinitesimal generator of the $\mathcal{LS}_i$ systems.

Let $\mathbf{Q}$ be the infinitesimal generator of a St-DSSP. We can rewrite $\mathbf{Q}$ as

$$\mathbf{Q} = \mathbf{R} - \mathbf{\Delta} \tag{1}$$

where $\mathbf{\Delta}$ is a diagonal matrix and $\mathbf{\Delta}[i,i] = \sum_{k \neq i} \mathbf{Q}[i,k]$. The same definitions hold for the $\mathcal{LS}_i$ components, for which we can define

$$\mathbf{Q}_i = \mathbf{R}_i - \mathbf{\Delta}_i$$

If we arrange the order of states according to the high level state $\mathbf{z}$, then we can describe the matrices $\mathbf{Q}$ and $\mathbf{R}$ (respectively, $\mathbf{Q}_i$ and $\mathbf{R}_i$) in terms of blocks $(\mathbf{z}, \mathbf{z}')$, of size $|\mathrm{RS}_{\mathbf{z}}(\mathcal{S})| \cdot |\mathrm{RS}_{\mathbf{z}'}(\mathcal{S})|$ (respectively, $|\mathrm{RS}_{\mathbf{z}}(\mathcal{LS}_i)| \cdot |\mathrm{RS}_{\mathbf{z}'}(\mathcal{LS}_i)|$). We shall indicate with $\mathbf{Q}(\mathbf{z},\mathbf{z}')$ and $\mathbf{R}(\mathbf{z},\mathbf{z}')$ ($\mathbf{Q}_i(\mathbf{z},\mathbf{z}')$ and $\mathbf{R}_i(\mathbf{z},\mathbf{z}')$, respectively) such blocks.

Blocks $\mathbf{R}_i(\mathbf{z},\mathbf{z}')$ with $\mathbf{z} = \mathbf{z}'$ have non null entries that are due *only* to the firing of transitions in $T_i \setminus$ TI (internal behaviour), while blocks $\mathbf{R}_i(\mathbf{z},\mathbf{z}')$ with $\mathbf{z} \neq \mathbf{z}'$ have non null entries due *only* to the firing of transitions in TI.

Let $\mathrm{TI}_{\mathbf{z},\mathbf{z}'}$ with $\mathbf{z} \neq \mathbf{z}'$, be the set of transitions $t \in$ TI such that $\mathbf{z} \xrightarrow{t} \mathbf{z}'$ in the basic skeleton $\mathcal{BS}$. From a matrix $\mathbf{R}_i(\mathbf{z},\mathbf{z}')$, with $\mathbf{z} \neq \mathbf{z}'$ we can build additional matrices $\mathbf{K}_i(t)(\mathbf{z},\mathbf{z}')$, for each $t \in \mathrm{TI}_{\mathbf{z},\mathbf{z}'}$, according to the following definition:

$$\mathbf{K}_i(t)(\mathbf{z},\mathbf{z}')[\mathbf{m},\mathbf{m}'] = \begin{cases} 1 & \text{if } \mathbf{m} \xrightarrow{t} \mathbf{m}' \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{m}$ and $\mathbf{m}'$ are two of the states with a high level view equal to $\mathbf{z}$ and $\mathbf{z}'$ respectively: $\mathbf{m}|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}$, and $\mathbf{m}'|_{H_1 \cup \ldots \cup H_K \cup B} = \mathbf{z}'$

We can now build the following matrices $\mathbf{G}(\mathbf{z},\mathbf{z}')$ of size $|\mathrm{PS}_{\mathbf{z}}(\mathcal{S})| \cdot |\mathrm{PS}_{\mathbf{z}'}(\mathcal{S})|$:

$$\begin{array}{rcl} \mathbf{G}(\mathbf{z},\mathbf{z}) & = & \bigoplus_{i=1}^{K} \mathbf{R}_i(\mathbf{z},\mathbf{z}) \\ \mathbf{G}(\mathbf{z},\mathbf{z}') & = & \sum_{t \in \mathrm{TI}_{\mathbf{z},\mathbf{z}'}} w(t) \bigotimes_{i=1}^{K} \mathbf{K}_i(t)(\mathbf{z},\mathbf{z}') \end{array} \tag{2}$$

The following theorem states that a St-DSSP system can be solved for the steady-state distribution using the $\mathbf{G}$ matrix defined by the $\mathbf{G}(\mathbf{z},\mathbf{z})$ and $\mathbf{G}(\mathbf{z},\mathbf{z}')$ blocks of equation (2).

Theorem 9 *Let define a St-DSSP* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}, w \rangle$, $\mathbf{Q}$ *its infinitesimal generator,* $\mathcal{LS}_i$ *its low level systems* ($i = 1, \ldots, K$)*, and* $\mathcal{BS}$ *its basic skeleton. Let* $\mathbf{R}$ *be the matrix defined by equation (1), and* $\mathbf{G}$ *the one defined by equations (2). Then:*
1. $\forall \mathbf{z}$ *and* $\mathbf{z}' \in \mathrm{RS}(\mathcal{BS})$*:* $\mathbf{R}(\mathbf{z},\mathbf{z}')$ *is a submatrix of* $\mathbf{G}(\mathbf{z},\mathbf{z}')$*.*
2. $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S})$ *and* $\forall \mathbf{m}' \in \mathrm{PS}(\mathcal{S}) \setminus \mathrm{RS}(\mathcal{S})$ *:* $\mathbf{G}[\mathbf{m},\mathbf{m}'] = 0$*.*

Proof:
In all cases in which $\mathbf{m}$ is a reachable state of $\mathrm{RS}(\mathcal{S})$ we know, by Theorem 8, that $\mathbf{m}$ can be rewritten in terms of the places of the single automata, of the buffers, and of the implicit places, as:

$$\mathbf{m} = \mathbf{l}_1, \ldots, \mathbf{l}_K, \mathbf{b}, \mathbf{H}$$

where $\mathbf{l}_i = \mathbf{l}|_{P_i}$, $\mathbf{b} = \mathbf{z}|_B$, and $\mathbf{H} = \mathbf{z}|_{H_1 \cup \ldots \cup H_K}$, which implies that $\mathbf{m}$ has been obtained from the product of the K local states $\mathbf{l}_i, \mathbf{b}, \mathbf{H}$, for $i \in \{1, \ldots, K\}$.

We first prove that:

$$\forall\, \mathbf{m}, \mathbf{m}' \in \mathrm{RS}(\mathcal{S}): \quad \mathbf{R}[\mathbf{m},\mathbf{m}'] = \mu \implies \mathbf{G}[\mathbf{m},\mathbf{m}'] = \mu$$

(since from $\mathbf{m}$ and $\mathbf{m}'$ the high level states $\mathbf{z}$ and $\mathbf{z}'$ are uniquely determined, we usually omit the specification of $\mathbf{z}$ and $\mathbf{z}'$). Considering that $\mathcal{S}$ and $\mathcal{ES}$ have the same behaviour, and the same set of states, for notational convenience we use $\mathcal{ES}$ instead of $\mathcal{S}$, and we therefore prove that

$$\forall\, \mathbf{m}, \mathbf{m}' \in \mathrm{RS}(\mathcal{ES}): \quad \mathbf{R}[\mathbf{m},\mathbf{m}'] = \mu \implies \mathbf{G}[\mathbf{m},\mathbf{m}'] = \mu$$

Case $\mathbf{z} = \mathbf{z}'$: if the high level view is the same, then the change of state from $\mathbf{m}$ to $\mathbf{m}'$ can only be due to internal transitions (i.e., not belonging to TI), thus belonging to a single automata $\mathcal{SM}_i$. But, by definition of $\bigoplus$, $\mathbf{G}(\mathbf{z},\mathbf{z})$ expresses the independent composition of the stochastic processes represented by the $\mathbf{R}_i(\mathbf{z},\mathbf{z})$, which is exactly the behaviour of $\mathcal{LS}_i$ system due to transitions local to $\mathcal{SM}_i$ (transitions belonging to $T_i \setminus$ TI).
Case $\mathbf{z} \neq \mathbf{z}'$: if the high level view is different, the change of state from $\mathbf{m}$ to $\mathbf{m}'$ can only be due to transitions in TI. Let t be such a transition and assume, for the time being, that there is only one, so that $\mu = w(t)$. If t is a transition in the interface, then its enabling depends on the marking of the buffer places, and on the places of a *single* automata.

If t is enabled in $\mathbf{m}$, then t is enabled in *each* $(\mathbf{l}_i, \mathbf{b}, \mathbf{H})$ state of the K $\mathcal{LS}_i$ systems, and in each $\mathcal{LS}_i$ system produces a change of state $\mathbf{l}_i, \mathbf{b}, \mathbf{H} \xrightarrow{t} \mathbf{l}'_i, \mathbf{b}', \mathbf{H}$. By definition of $\mathbf{K}_i(t)(\mathbf{z},\mathbf{z}')$ we have that $\mathbf{K}_i(t)(\mathbf{z},\mathbf{z}')[\mathbf{m}_i, \mathbf{b}, \mathbf{H}] = 1$, $\forall i$, and by definition of $\bigotimes$ there is a 1 in the corresponding entry of $\bigotimes_{i=1}^{K} \mathbf{K}_i(t)(\mathbf{z},\mathbf{z}')$, and therefore a value of $w(t)$ in the $\mathbf{G}(\mathbf{z},\mathbf{z}')$ matrix.

In the expression of $\mathbf{G}(\mathbf{z},\mathbf{z}')$ the situation in which more than one transition realizes the same change of state is accounted for by the summation over all transitions in $\mathrm{TI}_{\mathbf{z},\mathbf{z}'}$.

We now prove the second part of the theorem, that can be rewritten, in terms of $\mathcal{ES}$, as:

$$\forall \mathbf{m} \in \mathrm{RS}(\mathcal{ES}), \forall \mathbf{m}' \in \mathrm{PS}(\mathcal{S}) \setminus \mathrm{RS}(\mathcal{ES}) : \mathbf{G}[\mathbf{m}, \mathbf{m}'] = 0$$

As in the previous case, from $\mathbf{m}$ and $\mathbf{m}'$ the high level states $\mathbf{z}$ and $\mathbf{z}'$ are uniquely determined, and we consider for $\mathbf{m}$ and $\mathbf{m}'$ the same decomposition in terms of state machines, buffers, and implicit places.

We prove this part by contradiction, assuming that $\mathbf{G}[\mathbf{m}, \mathbf{m}'] \neq 0$, which can be rewritten as: $\mathbf{G}[(\mathbf{l}_1, \ldots, \mathbf{l}_K, \mathbf{b}, \mathbf{H}), (\mathbf{l}'_1, \ldots, \mathbf{l}'_K, \mathbf{b}', \mathbf{H}')] \neq 0$.
Case $\mathbf{z} = \mathbf{z}'$: by definition of $\bigoplus$, $\mathbf{G}[\mathbf{m}, \mathbf{m}'] \neq 0$ implies that there exists exactly one index i such that $\mathbf{R}_i(\mathbf{z}, \mathbf{z})[(\mathbf{l}_i, \mathbf{b}, \mathbf{H}), (\mathbf{l}'_i, \mathbf{b}, \mathbf{H})] \neq 0$, but since this change of state can be due only to the $\mathbf{l}_i$ portion of the state (since the high level view $\mathbf{z}$ does not change), then $\mathbf{R}_i(\mathbf{z}, \mathbf{z})[(\mathbf{l}_i, \mathbf{b}, \mathbf{H}), (\mathbf{l}'_i, \mathbf{b}, \mathbf{H})] \neq 0$. Therefore there must be a transition $t \in T_i \setminus \mathrm{TI}$ which is enabled in state $(\mathbf{l}_i, \mathbf{b}, \mathbf{H})$ of $\mathcal{LS}_i$, but then t is enabled also in state $\mathbf{m} = (\mathbf{l}_1, \ldots, \mathbf{l}_K, \mathbf{b}, \mathbf{H})$, and its firing produces $\mathbf{m}' = (\mathbf{l}_1, \ldots, \mathbf{l}'_i, \ldots, \mathbf{l}_K, \mathbf{b}, \mathbf{H})$, thus making $\mathbf{m}'$ is reachable in $\mathcal{ES}$, which contradicts the hypothesis.
Case $\mathbf{z} \neq \mathbf{z}'$: by definition of $\bigotimes$, $\mathbf{G}[\mathbf{m}, \mathbf{m}'] \neq 0$ implies that there exists a $t \in \mathrm{TI}$ such that, for all indices i, $\mathbf{K}_i(t)[(\mathbf{l}_i, \mathbf{b}, \mathbf{H}), (\mathbf{l}'_i, \mathbf{b}', \mathbf{H}')] = 1$, but then t is enabled in state $\mathbf{m}$ of $\mathcal{ES}$, and its firing will exactly produce the state $(\mathbf{l}'_1, \ldots, \mathbf{l}'_K, \mathbf{b}', \mathbf{H}')$, thus making $\mathbf{m}'$ is reachable in $\mathcal{ES}$, and therefore in $\mathcal{S}$, which contradicts the hypothesis. $\diamondsuit$

As a consequence the steady state distribution of a St-DSSP system can be computed using the $\mathbf{G}$ matrices given in equations (2). Indeed, as is the *Superposed GSPN* case [11], if we apply an iterative solution method for $\boldsymbol{\pi} \cdot \mathbf{G} = \mathbf{0}$, and if the initial probability vector assigns a non-null probability only to reachable states, for example by assigning a value of 1 to the initial marking, then by the second item of the above theorem we never assign a non-null probability to a non reachable state.

Coming back to the example, we can order states in $\mathrm{RS}(\mathcal{S})$ according to their projection over $\mathcal{BS}$, so that $\mathbf{R}$ can be written in block structured form as:

$$\mathbf{R} = \left(\begin{array}{c|c|c|c}
\mathbf{R}(\mathbf{z}_1,\mathbf{z}_1) & \mathbf{R}(\mathbf{z}_1,\mathbf{z}_2) & \mathbf{R}(\mathbf{z}_1,\mathbf{z}_3) & \mathbf{R}(\mathbf{z}_1,\mathbf{z}_4) \\ \hline
\mathbf{R}(\mathbf{z}_2,\mathbf{z}_1) & \mathbf{R}(\mathbf{z}_2,\mathbf{z}_2) & \mathbf{R}(\mathbf{z}_2,\mathbf{z}_3) & \mathbf{R}(\mathbf{z}_2,\mathbf{z}_4) \\ \hline
\mathbf{R}(\mathbf{z}_3,\mathbf{z}_1) & \mathbf{R}(\mathbf{z}_3,\mathbf{z}_2) & \mathbf{R}(\mathbf{z}_3,\mathbf{z}_3) & \mathbf{R}(\mathbf{z}_3,\mathbf{z}_4) \\ \hline
\mathbf{R}(\mathbf{z}_4,\mathbf{z}_1) & \mathbf{R}(\mathbf{z}_4,\mathbf{z}_2) & \mathbf{R}(\mathbf{z}_4,\mathbf{z}_3) & \mathbf{R}(\mathbf{z}_4,\mathbf{z}_4)
\end{array}\right)$$

By definition of St-DSSP only interface transitions contribute to $\mathbf{R}(\mathbf{z}_i, \mathbf{z}_j)$, when $i \neq j$, while only internal transitions contribute to $\mathbf{R}(\mathbf{z}_i, \mathbf{z}_i)$. Let us consider in more detail blocks $\mathbf{R}(\mathbf{z}_1, \mathbf{z}_1)$ and $\mathbf{R}(\mathbf{z}_1, \mathbf{z}_2)$: we explicitly write in the matrix the state identifier for rows and columns as pairs of states of $\mathcal{LS}_1$ and $\mathcal{LS}_2$. For example the second row corresponds to state $\mathbf{v}_2$ of $\mathrm{RS}(\mathcal{S})$, which is obtained as composition of state $\mathbf{x}_1$ of $\mathcal{LS}_1$ and state $\mathbf{y}_2$ of $\mathcal{LS}_2$.
$\mathbf{R}(\mathbf{z}_1, \mathbf{z}_1) =$

	$\mathbf{x}_1, \mathbf{y}_1$	$\mathbf{x}_1, \mathbf{y}_2$	$\mathbf{x}_1, \mathbf{y}_3$	$\mathbf{x}_2, \mathbf{y}_1$	$\mathbf{x}_2, \mathbf{y}_2$	$\mathbf{x}_2, \mathbf{y}_3$
$\mathbf{x}_1, \mathbf{y}_1$		$w(T_{13})$	$w(T_{12})$	$w(Ta_2)$		
$\mathbf{x}_1, \mathbf{y}_2$					$w(Ta_2)$	
$\mathbf{x}_1, \mathbf{y}_3$						$w(Ta_2)$
$\mathbf{x}_2, \mathbf{y}_1$					$w(T_{13})$	$w(T_{12})$
$\mathbf{x}_2, \mathbf{y}_2$						
$\mathbf{x}_2, \mathbf{y}_3$						

$\mathbf{R}(\mathbf{z}_1, \mathbf{z}_2) =$

	$\mathbf{x}_3, \mathbf{y}_4$	$\mathbf{x}_3, \mathbf{y}_5$	$\mathbf{x}_3, \mathbf{y}_6$	$\mathbf{x}_3, \mathbf{y}_7$	$\mathbf{x}_4, \mathbf{y}_4$	$\mathbf{x}_4, \mathbf{y}_5$	$\mathbf{x}_4, \mathbf{y}_6$	$\mathbf{x}_4, \mathbf{y}_7$
$\mathbf{x}_1, \mathbf{y}_1$								
$\mathbf{x}_1, \mathbf{y}_2$	$w(I_6)$							
$\mathbf{x}_1, \mathbf{y}_3$		$w(I_3)$						
$\mathbf{x}_2, \mathbf{y}_1$								
$\mathbf{x}_2, \mathbf{y}_2$					$w(I_6)$			
$\mathbf{x}_2, \mathbf{y}_3$						$w(I_3)$		

The only non null element of $\mathbf{R}_1(\mathbf{z}_1, \mathbf{z}_1)$ is $\mathbf{R}_1(\mathbf{z}_1, \mathbf{z}_1)[\mathbf{x}_1, \mathbf{x}_2] = w(Ta_2)$, while the only non null elements of $\mathbf{R}_2(\mathbf{z}_1, \mathbf{z}_1)$ are: $\mathbf{R}_2(\mathbf{z}_1, \mathbf{z}_1)[\mathbf{y}_1, \mathbf{y}_2] = w(T_{13})$ and $\mathbf{R}_2(\mathbf{z}_1, \mathbf{z}_1)[\mathbf{y}_1, \mathbf{y}_3] = w(T_{12})$.

The change of state from $\mathbf{z}_1$ to $\mathbf{z}_2$ can be due only to transition I_3 and I_6, we therefore only build matrices $\mathbf{K}_1(I_3)(\mathbf{z}_1, \mathbf{z}_2)$, $\mathbf{K}_2(I_3)(\mathbf{z}_1, \mathbf{z}_2)$, $\mathbf{K}_1(I_6)(\mathbf{z}_1, \mathbf{z}_2)$, and $\mathbf{K}_2(I_6)(\mathbf{z}_1, \mathbf{z}_2)$. $\mathbf{K}_1(I_3)(\mathbf{z}_1, \mathbf{z}_2)$ and $\mathbf{K}_1(I_6)(\mathbf{z}_1, \mathbf{z}_2)$ are indentity matrices. $\mathbf{K}_2(I_3)(\mathbf{z}_1, \mathbf{z}_2)[\mathbf{y}_3, \mathbf{y}_5] = 1$ and the other elements of $\mathbf{K}_2(I_3)(\mathbf{z}_1, \mathbf{z}_2)$ are null. $\mathbf{K}_2(I_6)(\mathbf{z}_1, \mathbf{z}_2)[\mathbf{y}_2, \mathbf{y}_4] = 1$ and the other elements of $\mathbf{K}_2(I_6)(\mathbf{z}_1, \mathbf{z}_2)$ are null.

According to equations (2) we get:

$$\begin{aligned}
\mathbf{G}(\mathbf{z}_1, \mathbf{z}_1) &= \mathbf{R}_1(\mathbf{z}_1, \mathbf{z}_1) \oplus \mathbf{R}_2(\mathbf{z}_1, \mathbf{z}_1) \\
\mathbf{G}(\mathbf{z}_1, \mathbf{z}_2) &= w(I_3)(\mathbf{K}_1(I_3)(\mathbf{z}_1, \mathbf{z}_2) \otimes \mathbf{K}_2(I_3)(\mathbf{z}_1, \mathbf{z}_2)) + \\
&\quad w(I_6)(\mathbf{K}_1(I_6)(\mathbf{z}_1, \mathbf{z}_2) \otimes \mathbf{K}_2(I_6)(\mathbf{z}_1, \mathbf{z}_2))
\end{aligned}$$

Since $\mathrm{PS}(\mathcal{S}) = \mathrm{RS}(\mathcal{S})$, then $\mathbf{G}(\mathbf{z}_1, \mathbf{z}_1) = \mathbf{R}(\mathbf{z}_1, \mathbf{z}_1)$ and $\mathbf{G}(\mathbf{z}_1, \mathbf{z}_2) = \mathbf{R}(\mathbf{z}_1, \mathbf{z}_2)$.

5 Conclusion

Exact performance evaluation of a modular class of PN's has been addressed. The considered modular class, DSSP, allows an efficient investigation of qualitative properties that are interesting before going towards the performance analysis. Among them, it can be pointed out that live and bounded DSSP are well characterized [17, 19], leading additionally to the existence of home state [18] that for the given stochastic interpretation produces ergodic models.

The quantitative analysis is based on a tensor algebra asynchronous approach, following [3, 4]. Some ideas on net decomposition that were used in [5, 15] for approximate performance evaluation are incorporated here for the exact analysis. The use of aggregated views of the original system, like the basic skeleton, allows a compositional way of computing the exact steady-state distribution.

Indeed the rule presented here is not the only possible: the characteristics that a reduction rule has to fullfil is that the states of the corresponding high level system must induce a *partition* on the reachability sets of the low level systems. This is straighforward in all those cases in which the low level system and the aggregated view can co-exist in the same net (as it is indeed the case in our reduction). For example the extreme case in which $\mathcal{BS}$ is equal to $\mathcal{S}$ (no aggregation, the $\mathcal{BS}$ contains exactly the same behaviour of $\mathcal{S}$) in general leads to dead $\mathcal{ES}$; on the other extreme the "most abstract" rule, that aggregates all local subnets into a common place with appropriate initial marking, may lead to a $\mathcal{BS}$ in which all combinations of interface transitions are legal firing sequences, so that there is no point in using the $\mathcal{BS}$ at all.

What is the complexity of the proposed approach with respect to the straightforward solution of the St-DSSP system? There are clearly limit cases: for example if all transitions are interface transitions (the system is tightly coupled), then $\mathcal{S} = \mathcal{BS} = \mathcal{LS}_i$, and it makes no sense to apply this method.

The computational cost to solve a St-DSSP is the sum of the cost to build the RG, the cost to compute the associated CTMC, and the cost of solving the characteristic equation $\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$. The proposed method has instead a cost that is due to: the construction of the $K + 1$ auxiliary models, the construction of the RG_i of each auxiliary model, the construction of the $\mathbf{R}_i(\mathbf{z}, \mathbf{z}')$ and $\mathbf{K}_i(t)(\mathbf{z}, \mathbf{z}')$ matrices (that may include a re-ordering of the states in the reachability sets), the cost of solving the characteristic equation $\boldsymbol{\pi} \cdot \mathbf{G} = \mathbf{0}$, when $\mathbf{G}$ is expressed as in equation (2). It is clear that the advantages/disadvantages of the method depend on the relative size of the reachability graphs of $\mathcal{S}$, $\mathcal{BS}$, and $\mathcal{LS}_i$.

The storage cost of the classical solution method is due to the storage of vector $\boldsymbol{\pi}$ of size $|\mathrm{RS}(\mathcal{S})|$, and of matrix $\mathbf{Q}$. Usually $\mathbf{Q}$ is stored in sparse form, so that, disregrading the diagonal, its occupation is of the same order as the number of arcs in the $\mathrm{RG}(\mathcal{S})$. The storage cost of the proposed approach is instead that of a vector $\boldsymbol{\pi}$ of size $|\mathrm{PS}(\mathcal{S})|$, and of a number of matrices, all stored in sparse form: the total number of non-null elements, disregarding the diagonal, is of the same order as the sum of the number of arcs in the K reachability graphs $\mathrm{RG}_i(\mathcal{LS}_i)$. It is actually possible to use a vector $\boldsymbol{\pi}$ of size $|\mathrm{RS}(\mathcal{S})|$, by paying an overhead in computational times, following the technique proposed in [13].

The difference between the number of arcs in $\mathrm{RG}(\mathcal{S})$ and the sum of the number of arcs in the K $\mathrm{RG}_i(\mathcal{LS}_i)$ is what makes the method applicable in cases in which a direct solution is not possible, due to the lack of memory to store $\mathbf{Q}$.

Despite the fact the *asynchronous* and *synchronous* approaches for a tensor algebra solution of GSPN's have been presented in the literature as different methods, in a way they are not. Indeed in the DSSP case we can build a Superposed GSPN system (synchronous approach) by simply superposing the K $\mathcal{LS}_i$ systems: we get a potential state space $\mathrm{PS}(\mathcal{S})$ that is bigger than the one obtained with the asynchronous approach, but it is still possible to iteratively compute the correct steady-state solution, if a non-null probability is assigned only to a (some of the) reachable state(s).

In a way, we can consider the technique based on the basic skeleton as a method to use high level information to cut from $\mathrm{PS}(\mathcal{S})$ states that are not reachable, and consequentely, from the $\mathbf{G}$ matrix, rows and columns that correspond to non-reachable states. The price we have to pay is a more complex expression for the supermatrix of the infinitesimal generator, and therefore a more complex storage scheme for the matrices.

Acknowledgements

We would like to thank Laura Recalde for the many clarifying discussions, and the anonymous referees for the helpful comments, specifically for the last comment in Section 3.

References

[1] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. Generalized stochastic Petri nets revisited: Random switches and priorities. In *Proc. of the Intern. Workshop on Petri Nets And Performance Mod-*

els, pages 44–53, Madison, WI, USA, August 1987. IEEE-CS Press.

[2] P. Buchholz. A hierarchical view of GCSPN's and its impact on qualitative and quantitative analysis. *Journal of Parallel and Distributed Computing*, 15(3):207–224, July 1992.

[3] P. Buchholz. A class of hierarchical queueing networks and their analysis. *Queueing Systems*, 15:59–80, 1994.

[4] P. Buchholz and P. Kemper. Numerical analyisis of stochastic marked graphs. In *Proc.* 6^{th} *Intern. Workshop on Petri Nets and Performance Models*, pages 32–41, Durham, NC, USA, October 1995. IEEE-CS Press.

[5] J. Campos, J. M. Colom, H. Jungnitz, and M. Silva. Approximate throughput computation of stochastic marked graphs. *IEEE Transactions on Software Engineering*, 20(7):526–535, July 1994.

[6] G. Chiola, S. Donatelli, and G. Franceschinis. GSPNs versus SPNs: What is the actual role of immediate transitions? In *Proc. of the* 4^{th} *Intern. Workshop on Petri Nets and Performance Models*, pages 20–31, Melbourne, Australia, December 1991. IEEE-CS Press.

[7] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets for symmetric modelling applications. *IEEE Transactions on Computers*, 42(11), November 1993.

[8] J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 113–145. Springer-Verlag, Berlin, 1991.

[9] M. Davio. Kronecker products and shuffle algebra. *IEEE Transactions on Computers*, 30(2):116–125, 1981.

[10] S. Donatelli. Superposed stochastic automata: A class of stochastic Petri nets with parallel solution and distributed state space. *Performance Evaluation*, 18:21–36, 1993.

[11] S. Donatelli. Superposed generalized stochastic Petri nets: Definition and efficient solution. In R. Valette, editor, *Proc. of the* 15^{th} *Intern. Conference on Applications and Theory of Petri Nets*, volume 815 of *Lecture Notes in Computer Science*, pages 258–277. Springer-Verlag, Berlin Heidelberg, 1994.

[12] S. Haddad and P. Moreaux. Asynchronous Composition of High Level Petri nets: a Quantitative Approach. In J. Billington and W. Reisig editors, *Proc. of the* 17^{th} *Intern. Conference on Applications and Theory of Petri Nets*, volume 1091 of *Lecture Notes in Computer Science*, pages 192–211. Springer-Verlag, Berlin Heidelberg, 1996.

[13] P. Kemper. Numerical analyisis of superposed GSPN. In *Proc.* 6^{th} *Intern. Workshop on Petri Nets and Performance Models*, pages 52–61, Durham, NC, USA, October 1995. IEEE-CS Press.

[14] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transactions on Computers*, 31(9):913–917, September 1982.

[15] C. J. Pérez-Jiménez, J. Campos, and M. Silva. State machine reduction for the approximate performance evaluation of manufacturing systems modelled with cooperating sequential processes. In *Proc. of the 1996 IEEE Intern. Conf. on Robotics and Automation*, pages 1159–1165, Minneapolis, Minnesota, USA, April 1996.

[16] B. Plateau. PEPS: A package for solving complex Markov models of parallel systems. In R. Puigjaner and D. Poiter, editors, *Modeling techniques and tools for computer performance evaluation*, pages 291–306. Plenum Press, New York and London, 1990.

[17] L. Recalde, E. Teruel, and M. Silva. On well-formedness analysis: The case of Deterministic Systems of Sequential Processes. In J. Desel, editor, *Structures in Concurrency Theory, Berlin 1995*, pages 279–293. Springer, 1995.

[18] L. Recalde, E. Teruel, and M. Silva. Modelling and analysis of sequential processes that cooperate through buffers. Research Report RR-96, Dpto. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 1996.

[19] L. Recalde, E. Teruel, and M. Silva. Structure theory for a class of modular and hierarchical cooperating systems. Research Report RR-96, Dpto. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 1996.

[20] M. Silva. Introducing Petri nets. In F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F.B. Vernadat, editors, *Practice of Petri Nets in Manufacturing*, chapter 1. Chapman & Hall, London, 1993.

[21] E. Teruel, M. Silva, J. M. Colom, and J. Campos. Functional and performance analysis of cooperating sequential processes. In G. Cohen and J.P. Quadrat, editors, *Analysis and Optimization of Systems: Discrete Event Systems*, volume 199 of *Lecture Notes in Control and Information Sciences*, pages 169–175. Springer-Verlag, London, 1994.