Functional and Performance Analysis of Cooperating Sequential Processes *

E. Teruel, M. Silva, J.M. Colom, and J. Campos

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, E-50015 Zaragoza (Spain)

Summary. This paper presents some results concerning the structural analysis of sequential proceses cooperating via message passing through a set of buffers. Both functional — boundedness, deadlock-freeness, liveness, existence of home states — and performance — marking ergodicity, computation of visit ratios and insensitive throughput bounds — properties are considered.

1. Introduction

The design of distributed systems is usually a complex task, compelling the use of formal methods. A major trend in the modelling of concurrent and distributed systems is the use of a single formalism during the entire design and analysis process [9, 16]. Such formalism should provide:

- Basic modelling features like: simple primitives for the modelling of *sequence*, *choice*, and *synchronisation*; hierarchical and modular modelling methodologies; the possibility of parametrisation of models...
- A well founded logical theory providing the definition of functional properties like deadlock-freeness or the absence of (buffer) overflows, and validation algorithms for them.
- A natural representation of time and the possibility of qualitative and quantitative analysis of performance properties.

At present, two modelling paradigms satisfying these requirements are that based on *programming language constructs*, like CCS [11] or CSP [10], and that based on *graphical constructs*, like Petri nets [12], both extended with the corresponding time representations. In this paper we consider the latter and, in particular, we concentrate on systems obtained by the application of a simple modular design principle: several *sequential processes* execute concurrently and *cooperate using asynchronous communication by message passing through a set of buffers*. The restrictions imposed to the connectivity of buffers aim at preventing *competition*. The possible information contained in messages can be disregarded, paying attention to the control flow only. In other words, messages can be considered as *authorisations*. Application domains where this class of systems appear are computer networks, information

* This work was partially supported by the European ESPRIT BRA Project 7269 QMIPS and the Spanish PRONTIC 242/94.

systems, operating systems, real-time systems, nonsequential programming languages, and discrete part manufacturing systems, among others.

Several works exist concerning functional and performance analysis of systems of sequential processes communicating through buffers modelled with Petri nets. Various aspects of modelling and functional analysis can be found in [15, 17]. A first approach to efficient (with polynomial time complexity on the net size) performance analysis was presented in [6], stressing both functional and performance aspects. In this paper we bridge qualitative and quantitative aspects of *Deterministic Systems of Sequential Processes*, with the goal of obtaining benefits in both the validation of functional properties and the evaluation of performance indices of such net systems. The present paper is an updated version of [5]; some conjectures from [5] were proven in [14], from which we incorporated also some denotations.

The paper is organised as follows: in Section 2. the class of Deterministic Systems of Sequential Processes is defined. Section 3. deals with functional analysis, and Section 4. considers performance properties.

2. Deterministic Systems of Sequential Processes

The class that we consider in this paper is an extension of that introduced in [17], although we keep the same name. In that work, sequential processes are modelled with *safe State Machines* while the communication among them is described by their connection through particular places called *buffers.* Their buffers are private in the sense that each of them has only one input and only one output sequential process. Our extension allows that several sequential processes deposit messages (tokens) in a buffer.

2.1 Basic Definitions and Notations of Petri Nets

Let us recall some definitions and notations about Petri nets (see [12] for a more comprehensive presentation).

A *P/T net* is a 4–tuple $\mathcal{N} = (P, T, Pre, Post)$, where P and T are disjoint sets of *places* and *transitions* ($|P| = n$, $|T| = m$), and Pre and $Post$ are the *incidence functions* representing the input and output arcs: $Pre, Post\colon P \times T \to \mathbb{N} = \{0, 1, 2, \ldots\}$, which are usually represented in matrix form. The *incidence matrix* of the net is defined as $C = Post - Pre$. The incidence function of a given arc is called *weight* or *multiplicity.* When all weights are 0 or 1 the net is called *oridinary.* A net can be seen as a bipartite directed graph in which places and transitions are the two kinds of nodes. The conventional dot-notation is used for *pre-* and *post-sets* of (a set of) nodes. *Flows* are integer annullers of C. *Semiflows* are semipositive flows. Right and left annullers are called T- and P-(semi)flows respectively. Flows are important because they induce certain invariant relations which are useful for reasoning on the behaviour. Several structural properties are based on them. For instance, a net is *consistent* when it has a positive T-semiflow, and *conservative* when it has a positive P-semiflow.

A function $M\colon P \to \mathbb{N}$, usually represented in vector form, is called *marking*. A *P/T system*, or *marked Petri net*, $(\mathcal{N}, M_0)$, is a P/T net $\mathcal{N}$ with an *initial marking* M_0. A transition $t \in T$ is *enabled* at marking M iff $\forall p \in P$: $M(p) \geq Pre(p,t)$. A transition t enabled at M can *fire* yielding a new marking M' (*reached* marking) defined by $M'(p) = M(p) - Pre(p,t) + Post(p,t)$ (it is denoted by $M \xrightarrow{t} M'$). A sequence of transitions $\sigma = t_1 t_2 \ldots t_n$ is a *firing sequence* in $(\mathcal{N}, M_0)$ iff there exists a sequence of markings such that $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \ldots \xrightarrow{t_n} M_n$. In this case, marking M_n is said to be *reachable* from M_0 by firing σ, and this is denoted by $M_0 \xrightarrow{\sigma} M_n$. The *reachability set* $R(\mathcal{N}, M_0)$ is the set of all markings reachable from the initial marking. The function $\vec{\sigma}\colon T \to \mathbb{N}$, usually represented in vector form, is the *firing count vector* of σ. If $M_0 \xrightarrow{\sigma} M$, then $M = M_0 + C \cdot \vec{\sigma}$, which is referred to as the *linear state equation* of the net. A marking M is said to be *potentially reachable* iff $\exists \vec{\sigma} \in \mathbb{N}^m$ such that $M = M_0 + C \cdot \vec{\sigma} \geq 0$. Denoting by $PR(\mathcal{N}, M_0)$ the set of all potentially reachable markings, $PR(\mathcal{N}, M_0) \supseteq R(\mathcal{N}, M_0)$.

A place $p \in P$ is said to be *k–bounded* iff $\forall M \in R(\mathcal{N}, M_0)$, $M(p) \leq k$. A P/T system is said to be (marking) k–bounded iff every place is k–bounded, and bounded iff there exists some k for which it is k–bounded. A P/T system is *live* when every transition can ultimately occur from every reachable marking, and it is *deadlock-free* when at least one transition is enabled at every reachable marking. M is a *home state* in $(\mathcal{N}, M_0)$ iff it is reachable from every reachable marking, and $(\mathcal{N}, M_0)$ is *reversible* iff M_0 is a home state. Boundedness is necessary whenever the system is to be implemented, while liveness is often required, specially in reactive systems. A net $\mathcal{N}$ is *structurally bounded* when $(\mathcal{N}, M_0)$ is bounded for *every* M_0, and it is *structurally live* when *there exists* an M_0 such that $(\mathcal{N}, M_0)$ is live. Consequently, if a net $\mathcal{N}$ is structurally bounded and structurally live there exists some marking M_0 such that $(\mathcal{N}, M_0)$ is bounded and live, and we say it is *well-formed*.

Let $\mathcal{E}$ be the set of Equal Conflict sets of a net $\mathcal{N}$, i.e., the quotient set of the Equal Conflict relation, which is “being the same transition or having the same non-null pre-incidence function”. A general necessary condition for well-formedness of polynomial time complexity is:

Theorem 2.1 ([21]). *Let $\mathcal{N}$ be a* P/T *net.*
If $\mathcal{N}$ is well-formed, then it is conservative, consistent, and $\mathrm{rank}(C) < |\mathcal{E}|$.

2.2 Deterministic Systems of Sequential Processes, and Other Subclasses

State Machines are ordinary Petri nets such that every transition has only one input and only one output place ($\forall t \in T$: $|{}^\bullet t| = |t^\bullet| = 1$). State Machines allow the modelling of sequences, decisions (or conflicts), and re-entrancy (when they are marked with more than one token) but not synchronisation. Some trivial properties of State Machines are:

– The rank of their incidence matrix equals their number of places minus one.
– They are conservative (thus, structurally bounded).
– Structural liveness is equivalent to strong connectedness, which is equivalent to consistency.
– Liveness is equivalent to strong connecteness and being marked.
– Provided liveness, k–boundedness is equivalent to containing k tokens.

Regarding the timing, it is assumed that every cycle in a State Machine contains at least one timed transition. Topologically speaking, strongly connected State Machines are the Petri net counterpart of classical closed monoclass queueing networks. In *closed* networks, no customer leaves the system or arrives from the outside, hence the population is preserved. The corresponding property in Petri nets terminology is *conservativeness*, which leads to global token conservation laws for any initial marking.

Deterministic Systems of Sequential Processes are used for the modelling and analysis of distributed systems composed by sequential processes communicating through output-private buffers. Each sequential process (SP) is modelled by a safe (1–bounded) State Machine. The communication among them is described by *buffers* (places) which contain *products/messages* (tokens), which 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 SP (see Figure 2.1, where shaded places are the buffers).

Definition 2.1. *A P/T system* $(\mathcal{N}, M_0)$ *is a* Deterministic System of Sequential Processes (DSSP) *iff* $P = P_1 \cup \ldots \cup P_q \cup B$ *and* $T = T_1 \cup \ldots \cup T_q$ *are such that:*

1. $\forall i, j \in \{1, \ldots, q\}, i \neq j$: $P_i \cap P_j = \emptyset$, $T_i \cap T_j = \emptyset$, $P_i \cap B = \emptyset$.
2. $\forall i \in \{1, \ldots, q\}$: $(\mathcal{SP}_i, M_{0i}) = (P_i, T_i, Pre_i, Post_i, M_{0i})$ *is a safe strongly connected State Machine (where* Pre_i,$Post_i$, *and* M_{0i} *are the restrictions of* Pre, $Post$, *and* M_0 *to* P_i *and* T_i*).*
3. $\forall b \in B$:
   a) $|{}^\bullet b| \geq 1$ *and* $|b^\bullet| \geq 1$.
   b) $\exists i \in \{1, \ldots, q\}$, *such that* $b^\bullet \subseteq T_i$.
   c) $\forall p \in P_1 \cup \ldots \cup P_q$: $t, t' \in p^\bullet \Rightarrow Pre(b, t) = Pre(b, t')$.

The first two items of the previous definition state that a DSSP is composed by a set of SP's ($\mathcal{SP}_i, i = 1, \ldots, q$) and a set of buffers ($B$). By item *3.a*, buffers are neither source nor sink places. The output-private condition is expressed by condition *3.b*. This, together with *3.c* which forbids that a buffer affects the resolution of conflicts in an SP, prevents competition. This

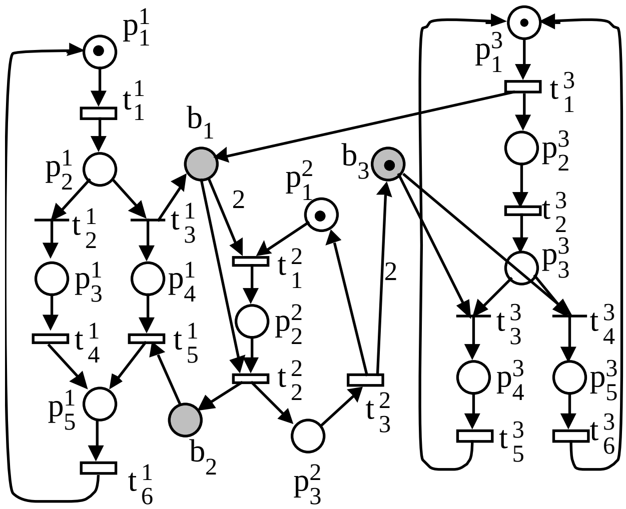

Fig. 2.1. A bounded and live Deterministic System of Sequential Processes.

definition generalises the class of DSSP's defined in [17], where buffers are required to have not only a single output SP (output-private) but also a single input one (input-private). From a queueing network perspective, DSSP's are a mild generalisation of *Fork-Join Queuing Networks with Blocking* where servers are complex (safe State Machines with a rich connectivity to buffers).

Another interesting subclass of P/T nets are *Equal Conflict* nets [20]:

Definition 2.2. *A P/T net $\mathcal{N}$ is an* Equal Conflict (EC) *net iff* $\forall t, t' \in T\colon {}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \forall p \in P\colon Pre(p,t) = Pre(p,t')$.

EC nets generalise the ordinary subclass of Extended Free Choice nets. Many nice results from the Free Choice theory have been recently extended to EC net systems:

- [20] The potential reachability graph of a live EC system is *directed*. Thus, live EC systems do not have *killing spurious solutions* (spurious solutions that do not enable any transition), and live and bounded EC systems have home states. Since a bounded strongly connected EC system is live iff it is deadlock-free, liveness of a bounded system can be determined by checking absence of integer solution to some linear equation system [19].
- [21] The possibility of marking boundedly and lively a given EC net can be determined in polynomial time: $\mathcal{N}$ is well-formed iff it is consistent, conservative, and rank$(C) = |\mathcal{E}| - 1$. Moreover, if $\mathcal{N}$ is well-formed, it can be decomposed in a meaningful way, similarly to the decomposition of a Free Choice net into State Machines and/or Marked Graphs. Liveness of the whole system can be compositionally characterised in terms of the liveness of the analogues to State Machine components.

Observe that, in a DSSP, all the private conflicts of the SP's are Equal Conflict sets, by the assumption that buffers do not disturb choices. All the remaining transitions are elementary Equal Conflict sets. Neither DSSP's are a subclass of EC nets nor the converse. Nevertheless, if it happens that all buffers of a DSSP have exactly one output Equal Conflict set ($\forall b \in B : b^\bullet \in \mathcal{E}$), that is, if they are *strictly* output-private, then obviously they are EC, and they are called *Equal Conflict DSSP (DSSP/EC)*. Naturally, DSSP/EC's inherit all the nice properties of EC net systems. In fact, by means of several transformations preserving, among other properties, boundedness, liveness and the existence of home states (see [5]), the results that are valid for the DSSP/EC subclass can be extended to many non EC nets, although *not* all DSSP's can be transformed into EC. As an example, we can transform the net of Figure 2.1 into a DSSP/EC by pre-fusion of t_1^2 and t_2^2 into t_{12}^2 [3]. The new transition has b_1 with weight 3 and p_1^2 as inputs and b_2 and p_3^2 as outputs. Place p_2^2 has been removed.

2.3 Time Representation

One of the advantages of Petri net models for the design and analysis of concurrent and distributed systems is that they can be naturally extended by time attributes in order to evaluate the system performance. We consider net systems with timed transitions. Marking and time independent Coxian random variables associated to the firing of transitions define their *service time*. The mean values of these variables are denoted s_i for each transition t_i of the net.

For the modelling of conflicts we use *immediate transitions* with the addition of (marking and time independent) *routing rates* [1]. In other words, for the subset of immediate transitions $\{t_1, \ldots, t_k\} \subset T$ being in conflict at each reachable marking, we assume that the constants $r_1, \ldots, r_k \in \mathbb{Q}^+$ are explicitly defined in the system interpretation in such a way that when $t_1, \ldots, t_k$ are simultaneously enabled, transition t_i, $i = 1, \ldots, k$, fires with relative rate $r_i / (\sum_{j=1}^{k} r_j)$. Consequently, routing is completely decoupled from duration of activities. The only restriction that this decoupling imposes to the system is that *preemption* cannot be modelled with two timed transitions (in conflict) competing for the tokens. (In other words, a *race policy* cannot be modelled. Our constraint is equivalent to the use of a *preselection policy* for the resolution of conflicts among timed transitions.)

Assuming the above described time interpretation, the timed model has almost surely the *fair progress* property, that is, no transition can be permanently enabled without firing. Additionally, it has the *local fairness* property, that is, all output transitions of a shared place simultaneously enabled at infinitely many markings will fire infinitely often. (In other words, all possible outcomes of any conflict have a non-null probability of firing.)

The *visit ratio* of transition t_i with respect to t_j, $v_i^{(j)}$, is the average number of times t_i is visited (fired) for each visit to (firing of) the reference transition t_j. The computation of visit ratios is interesting for the performance analysis of formal models. For example, it is well-known that the steady-state probability of a state in a product-form queueing network with single-server semantics [8] depends on the *average service demands* of customers from station i, defined as:

$$D_i^{(j)} \stackrel{def}{=} v_i^{(j)} \cdot s_i \qquad i = 1, \ldots, m \tag{2.1}$$

The computation of average service demands is also very important in the performance analysis of stochastic Petri net models. In Section 4., applying the theory presented in [7], we use these values to compute upper and lower bounds for the *throughput* of transitions, i.e. the average number of service completions (firings) per time unit, in a bounded and live DSSP.

3. Functional Analysis of DSSP's

One of the benefits of using restricted subclasses of nets is the availability of results that facilitate the analysis. In the case of DSSP's, we have a polynomial time characterisation of well-formedness, and a simple algebraic sufficient condition for liveness, based on the equivalence of liveness and deadlock-freeness, which is also proven to be necessary for DSSP/EC.

3.1 The Coarse net of a DSSP

In order to concentrate on the *interconnection level* of the net, we want to obviate the details regarding the inner structure of the SP's, while keeping relevant information concerning the effect on the buffers. A first approach could be substituting a (coarse) transition for each SP, keeping the interconnection with buffers. This is the *coarse structure* proposed in [15], which is not enough for analytical purposes (see [14]). To keep more information on the effect of buffer interconnections, we give symbolic relative rates to conflicting transitions (e.g., r_2^1, r_3^1 to t_2^1, t_3^1 in the example net of Figure 2.1), which are assumed to be *normalised* (e.g., $r_2^1 + r_3^1 = 1$), and take them into account when computing the coarse model. The idea is to replace each SP by a transition whose effect on the buffers is the same as that of the original SP in the long term when respecting the given rates. This is done using the unique minimal T-semiflow of each SP respecting those rates (see [14]). We call the obtained coarse model *coarse net*, even if it is not a net in a strict sense, because arc weights may not be natural numbers. We even speak of strong connectedness, conservativeness, or consistency: we would say, for instance, that the "net" in Figure 3.1, which is the coarse net of the DSSP of Figure 2.1, is strongly connected, conservative ($Y \cdot C = 0$ for $Y = [1\ 1\ 1]$), and consistent ($C \cdot X = 0$ for $X = [1\ r_3^1\ 2r_3^1]$).

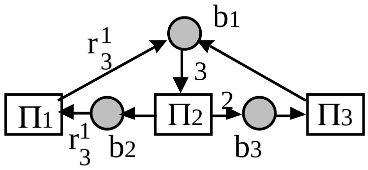

Fig. 3.1. The coarse net of the DSSP of Fig. 2.1.

With the aid of the structure theory of Choice-free nets [18], and using the fact that the coarse net is obtained by linear combinations of the transitions of the DSSP, the following properties of the coarse net are proven in [14]:

Proposition 3.1 ([14]). *Let $\mathcal{N}$ be the net of a DSSP.*
If $\mathcal{N}$ is conservative and consistent, then:

1. *The coarse net of $\mathcal{N}$ is strongly connected, conservative and Choice-free. If $\mathcal{N}$ is well-formed, then its coarse net is also consistent.*
2. $\mathrm{rank}(C) < |\mathcal{E}|$.

Observe that, in the example, the coarse net is a conservative and consistent Choice-free net *for all* (positive) rates. It is also interesting to observe that the complete T-semiflow structure of $\mathcal{N}$ is somehow represented in the coarse net thanks to the parametric weighting.

3.2 Well-Formedness and Liveness

The first important fact concerning liveness analysis of DSSP's is that liveness and deadlock-freeness are equivalent, assuming boundedness and strong connectedness.

Theorem 3.1 ([5]). *Let $(\mathcal{N}, M_0)$ be a bounded strongly connected DSSP. $(\mathcal{N}, M_0)$ is live iff it is deadlock-free.*

Combining Proposition 3.1.2 with Theorem 2.1, and also making extensive use of the T-semiflow structure of DSSP's and of Theorem 3.1, the following characterisation of well-formedness is obtained in [14]:

Theorem 3.2 ([14]). *Let $\mathcal{N}$ be the net of a DSSP.*
$\mathcal{N}$ is well-formed iff it is conservative, consistent, and $\mathrm{rank}(C) = |\mathcal{E}| - 1$.
Moreover, there exists M_0 such that $(\mathcal{N}, M_0)$ is a bounded and live DSSP.

If $\mathcal{N}$ is not well-formed, for which we have given a polynomial time characterisation, then for every bounded initial marking the system deadlocks. In performance terms, the system presents null throughput for any initial configuration of resources/customers due to a problem that is rooted on the net structure, so this should be checked prior to any other — more complex — analysis.

In case $\mathcal{N}$ is well-formed, the problem is determining whether the initial marking makes the system live or not. To achieve this, Theorem 3.1 can be used, so only deadlock-freeness needs to be proven, instead of liveness. In [19] a general sufficient condition for deadlock-freeness in terms of the solvability over the naturals of a set of linear equation systems is given. The basic idea is to ask for absence of potentially reachable deadlock markings. In the particular case of DSSP's, among other subclasses, such algebraic condition can be expressed as a single linear equation system (use the rules presented in [19] plus the particular transformation shown in Figure 3.2, that preserves deadlock-freeness — actually it preserves the language modulo a projection — thanks to the output-private and the State Machines' safeness hypothesis).

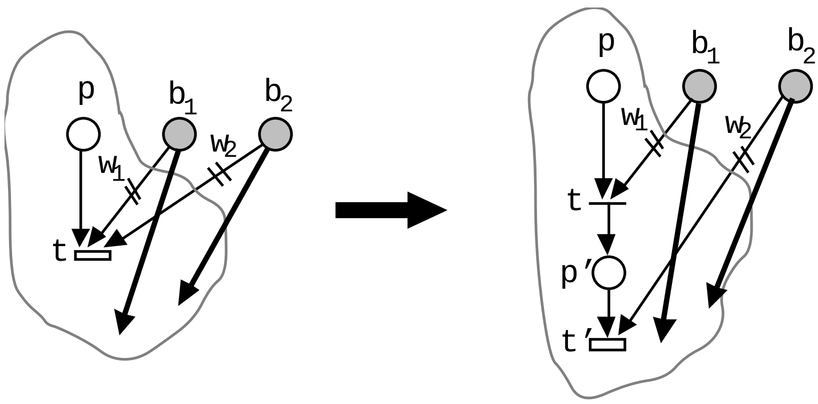

Fig. 3.2. A transformation preserving deadlock-freeness. (Place p must *not* be a choice place.)

As an example, the sufficient condition for liveness of the net of Figure 2.1 — which is well-formed — is absence of *natural* solutions to the linear equation system below. The first line defines the set of potentially reachable markings (i.e., it is the state equation). The following three lines express disabledness of "private" transitions of the three State Machines (i.e. transitions whose only input is a State Machine place), and the last four express disabledness of the transitions having some buffer as input ($SB(b)$ denotes the *structural (marking) bound* of buffer b, that is defined as $\max\{M(b) \mid M \in PR(\mathcal{N}, M_0)\}$; the structural bound of State Machine places is obviously one).

$$
\begin{array}{l}
M = M_0 + C \cdot \vec{\sigma} \\
M(p_1^1) = M(p_2^1) = M(p_3^1) = M(p_5^1) = 0 \\
M(p_3^2) = 0 \\
M(p_1^3) = M(p_2^3) = M(p_4^3) = M(p_5^3) = M(p_6^3) = 0 \\
SB(b_1) \cdot M(p_1^2) + M(b_1) \leq SB(b_1) + 1 \\
SB(b_1) \cdot M(p_2^2) + M(b_1) \leq SB(b_1) \\
SB(b_2) \cdot M(p_4^1) + M(b_2) \leq SB(b_2) \\
SB(b_3) \cdot M(p_3^3) + M(b_3) \leq SB(b_3)
\end{array}
$$

This general sufficient condition is also necessary in the case of EC systems [20]. (We conjecture that it is so also for DSSP's; safeness of the SP's is necessary here, because there are examples of DSSP nets with a 2–bounded SP having killing spurious solutions.) In the example net, if we pre-fuse t_1^2 and t_2^2, as described above, to get a DSSP/EC, we have to replace the two conditions involving b_1 by:

$$SB(b_1) \cdot M(p_1^2) + M(b_1) \leq SB(b_1) + 2$$

and now absence of natural solutions is equivalent to liveness.

4. Performance Analysis of DSSP's

4.1 Home States and Ergodicity

It is well-known that under (possibly marking-dependent) exponentially distributed random variables associated to the firing of transitions and Bernouilli trials for the succesive resolutions of each conflict, the underlying *Continuous Time Markov Chain* (CTMC) is isomorphous to the reachability graph of the untimed net model [2]. Thus, the existence of home states leads to ergodicity of the marking process for bounded net systems with exponential firing times.

Theorem 4.1 ([5]). *Let $(\mathcal{N}, M_0)$ be a live and bounded DSSP/EC with Coxian random variables associated to the firing of transitions and Bernouilli trials for the succesive resolutions of each conflict.*

The underlying CTMC of $(\mathcal{N}, M_0)$ is ergodic.

Although the property is stated for DSSP/EC, again the result can be extended to many non EC nets by the corresponding net transformations preserving the existence of home states. We conjecture that DSSP's have home states too, that is, that the ergodicity result holds for the entire DSSP class.

4.2 Computation of Visit Ratios

The computation of the average service demands of tokens from transitions, Equation (2.1), is useful for the performance analysis of timed systems. Assuming that the average service times of transitions, s_i, are known, then it is necessary to compute the vector of visit ratios to transitions, $\vec{v}^{(j)}$.

If $\mathcal{N}$ is consistent, the visit ratio vector normalised for t_j, $\vec{v}^{(j)}$, should be a T-semiflow of $\mathcal{N}$. Otherwise stated (observe that $v_i^{(j)} \geq 0$ by definition):

$$C \cdot \vec{v}^{(j)} = 0 \tag{4.1}$$

The conflicts in the SP's of a DSSP are free, because the buffers do not condition the conflict resolution. Structurally speaking, these conflicts correspond to Equal Conflicts at the net level. On the sequel, for easy presentation, Equal Conflict sets will be supposed to have at most two transitions; otherwise serialise choices into binary ones. Let t_a and t_b be in Equal Conflict relation. The corresponding visit ratios should verify the following equation:

$$r_b \cdot v_a^{(j)} - r_a \cdot v_b^{(j)} = 0 \tag{4.2}$$

An equation like (4.2) holds for every (binary) Equal Conflict. Rewritten in vector form: $r_{ab} \cdot \vec{v}^{(j)} = 0$, which for the set of all Equal Conflicts leads to:

$$R \cdot \vec{v}^{(j)} = 0 \tag{4.3}$$

where R is a matrix with $m - |\mathcal{E}|$ (number of binary Equal Conflicts) rows and m columns. In other words, $m - |\mathcal{E}|$ is the number of independent linear relations fixed by the routing rates at (binary) Equal Conflicts, so $\text{rank}(R) = m - |\mathcal{E}|$. Using this together with Theorem 3.2, it follows that:

Theorem 4.2 ([5]). *Let $\mathcal{N}$ be a well-formed DSSP net.*
The system of equations:

$$\begin{pmatrix} C \\ R \end{pmatrix} \cdot \vec{v}^{(j)} = 0,\ v_j^{(j)} = 1 \tag{4.4}$$

has only one solution (i.e., the vector of visit ratios depends neither on the marking, provided it allows infinite behaviours, nor on the service times).

4.3 Performance Bounds

This section is devoted to present some *insensitive* (i.e., holding for any probability distribution function for the firing times) performance bounds. Basically, throughput upper bounds are computed by finding the slowest isolated subnet among those generated by P-semiflows of the net, and are presented in the next theorem.

Theorem 4.3 ([7]). *For a DSSP, $(\mathcal{N}, M_0)$, a lower bound for the mean interfiring time $\Gamma^{(j)}$ of transition t_j (or its inverse an upper bound for the throughput σ_j^*) can be computed by solving the following linear programming problem:*

$$\begin{array}{rll} \Gamma^{(j)} \geq & \text{maximise} & Y \cdot Pre \cdot \vec{D}^{(j)} \\ & \text{subject to} & Y \cdot C = 0 \\ & & Y \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \qquad \text{(LPP1)}$$

where $\vec{D}^{(j)}$ is the vector of average service demands for transitions.

We remark that the computation of the above bound has polynomial time complexity on the net size. This is because the computation of vector $\vec{D}^{(j)}$ is polynomial and because linear programming problems can also be solved in polynomial time.

If the solution of (LPP1) is unbounded and since it is a lower bound for the mean interfiring time of transition t_j, the non-liveness can be assured (infinite interfiring time). If the visit ratios of all transitions are non-null, the unboundedness of the problem (LPP1) implies that a total deadlock is reached by the net. This result has the following interpretation: if (LPP1) is unbounded then there exists an unmarked P-semiflow, and the system is non-live.

Concerning throughput lower bounds, provided the net system is live, they can be derived by summing up the service time of all transitions, weighted by the visit ratios. This computation implies a complete sequentialisation of all the activities represented in the model.

Theorem 4.4 ([7]). *For a bounded and live DSSP, $(\mathcal{N}, M_0)$, an upper bound for the mean interfiring time $\Gamma^{(j)}$ of transition t_j (or its inverse a lower bound for the throughput) is:*

$$\Gamma^{(j)} \leq \sum_{i=1}^{m} s_i \; v_i^{(j)} = \sum_{i=1}^{m} D_i^{(j)} \qquad (4.5)$$

where s_i, $v_i^{(j)}$, and $D_i^{(j)}$ are the mean service time, visit ratio, and average service demand, respectively, for transition t_i, $i = 1, \ldots, m$.

We remark that the above bound, provided the system is bounded and live, can also be computed in polynomial time, since the vector of visit ratios can be computed with such complexity.

Bounds for other performance indices can be computed using classical formulas in Queuing Networks theory such as Little's formula.

The number of tokens in a place defines the lenght of the represented queue (including the customers in service!). Thus it may be important to known bounds on average marking of places.

As an example, in [4] it has been shown that the following are lower and upper bounds for the average marking, $\overline{M}$:

$$\overline{M}^{lb} = Pre \cdot \mathcal{S} \cdot \vec{\sigma^*}^{lb} \tag{4.6}$$

$$\overline{M}^{ub}(p) \quad = \quad \max \; \{ \; M(p) \mid B \cdot M = B \cdot M_0 \; , \; M \geq \overline{M}^{lb} \; \} \tag{LPP2}$$

where $\mathcal{S} = \text{diag}\ (s_i)$, $\vec{\sigma^*}^{lb}$ is the vector of throughput lower bounds, and the rows of B are the basis of left annullers of the incidence matrix of the net.

As an interesting remark, the reader may check that a structural absolute bound for the marking of a place is given for conservative nets (i.e., $\exists Y > 0$, $Y \cdot C = 0$) by the following expression:

$$SB(p) \quad = \quad \max \; \{ \; M(p) \mid B \cdot M = B \cdot M_0 \; , \; M \geq 0 \; \} \tag{LPP3}$$

The constraint in (LPP3) being weaker than that in (LPP2) ($M \geq \overline{M}^{lb}$ is transformed into $M \geq 0$), it is obvious that $\overline{M}^{ub} \leq SB(p)$.

In [13], additional results on the approximate throughput computation for DSSP's have been derived, based on iterative response time approximation algorithms.

References

1. M. Ajmone-Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93-122, 1984.
2. M. Ajmone-Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems.* MIT Press, 1986.
3. G. Berthelot. Checking properties of nets using transformations. In G. Rozenberg (ed.) *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 19–40. Springer Verlag, 1986.
4. J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed Free Choice synchronized monoclass queuing networks. *IEEE Trans. Automatic Control*, 36(12):1368–1382, 1991.
5. J. Campos, J. M. Colom, M. Silva, and E. Teruel. Functional and performance analysis of cooperating sequential processes. In O. J. Boxma and G. M. Koole (ed.) *Performance Evaluation of Parallel and Distributed Systems. Solution Methods*, volume 106 of *CWI Tracts*, pages 233–251. CWI, 1994.
6. J. Campos and M. Silva. Steady-state performance evaluation of totally open systems of Markovian sequential processes. In M. Cosnard and C. Girault (ed.) *Decentralized Systems*, 427-438. North-Holland, 1990.

7. J. Campos and M. Silva. Structural Techniques and Performance Bounds of Stochastic Petri Net Models. In G. Rozenberg (ed.) *Advances in Petri Nets 1992*, volume 609 of *LNCS*, 352-391. Springer-Verlag, 1992.
8. E. Gelenbe and G. Pujolle. *Introduction to Queuing Networks.* Wiley, 1987.
9. U. Herzog. Performance evaluation and formal description. In *Procs. of the IEEE Conference CompEuro 91*, 750-756, Bologna, Italy, May 1991.
10. C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall, Englewood Cliffs, NJ, 1985.
11. R. Milner. *A Calculus of Communicating Systems.* Prentice Hall, London, 1989.
12. T. Murata. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541-580, 1989.
13. C.J. Perez, J. Campos, M. Silva: On approximate throughput computation of DSSP. Submitted paper. March, 1995.
14. L. Recalde, E. Teruel, M. Silva: On Well-formedness Analysis: The Case of DSSP. To appear in J. Desel (ed.) *Proc. of Int. Workshop on Structures in Concurrency Theory*. Springer-Verlag, 1995.
15. W. Reisig. Deterministic buffer synchronization of sequential processes. *Acta Informatica*, 18, 117-134, 1982
16. M. Silva. Interleaving functional and performance structural analysis of net models. In M. Ajmone Marsan (ed.) *Application and Theory of Petri Nets 1993*, volume 691 of *LNCS*, 17-23. Springer-Verlag, 1993.
17. Y. Souissi and N. Beldiceanu. Deterministic Systems of Sequential Processes: Theory and tools. In F.H. Vogt (ed.) *Concurrency 88*, volume 335 of *LNCS*, 380-400. Springer-Verlag, 1988.
18. E. Teruel. *Structure Theory of Weighted Place/Transition Net Systems: The Equal Conflict Hiatus.* PhD thesis, DIEI. Univ. Zaragoza, June 1994.
19. E. Teruel, J.M. Colom and M. Silva. Linear analysis of deadlock-freeness of Petri net models. In J.W. Nieuwenhuis et al. (ed.) *Second European Control Conference*, volume 2, 513-518. North-Holland, 1993.
20. E. Teruel and M. Silva. Liveness and home states in Equal Conflict systems. In M. Ajmone Marsan (ed.) *Application and Theory of Petri Nets 1993*, volume 691 of *LNCS*, 415-432. Springer-Verlag, 1993.
21. E. Teruel and M. Silva. Well-formedness of Equal Conflict systems. In R. Valette (ed.) *Application and Theory of Petri Nets 1994*, volume 815 of *Lecture Notes in Computer Science*, pages 491–510. Springer Verlag, 1994.