AGGREGATION OF STATES IN COLORED STOCHASTIC PETRI NETS : APPLICATION TO A MULTIPROCESSOR ARCHITECTURE

C. DUTHEILLET S. HADDAD

Laboratoire MASI- CNRS UA 818
Université Pierre & Marie Curie
4 place Jussieu
75252 Paris Cedex 05

ABSTRACT

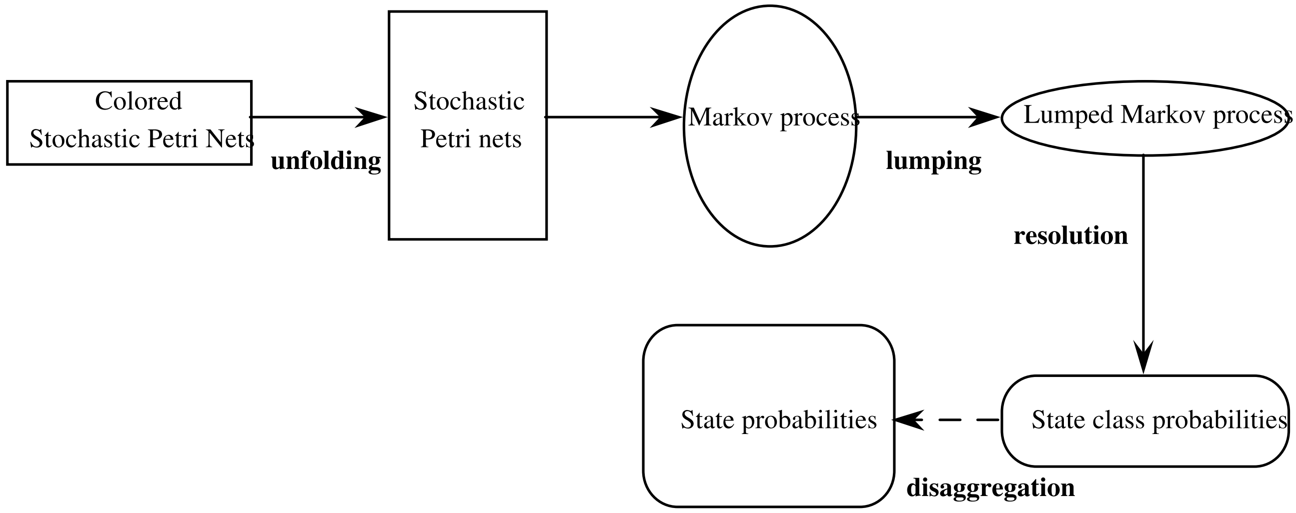
In this paper, we present a lumping method based on a particular class of colored Petri nets. We prove on an example that the method can be usefully applied to the performance evaluation of symmetric systems.

Topics : Generalized stochastic nets, Higher-level net models.

I - INTRODUCTION

Stochastic Petri nets [Nat 80, Mol 81, Zub 85] are more and more widely used for the performance evaluation of synchronized systems. Some extensions of the original models have been proposed, either including more general timing constraints, such as immediate firing [Ajm 84], deterministic delays [Ajm 87a], general timing [Dug 84], or extending the underlying model by including simultaneous transition firings [Haa 87], or unbounded places [Flo 86]. Other models have been presented, that are particularly well suited to the study of some specific problems, such as fault tolerance and degradable performance [Mey 85], resource sharing [Hol 85], or communication protocol performance [Raz 85].

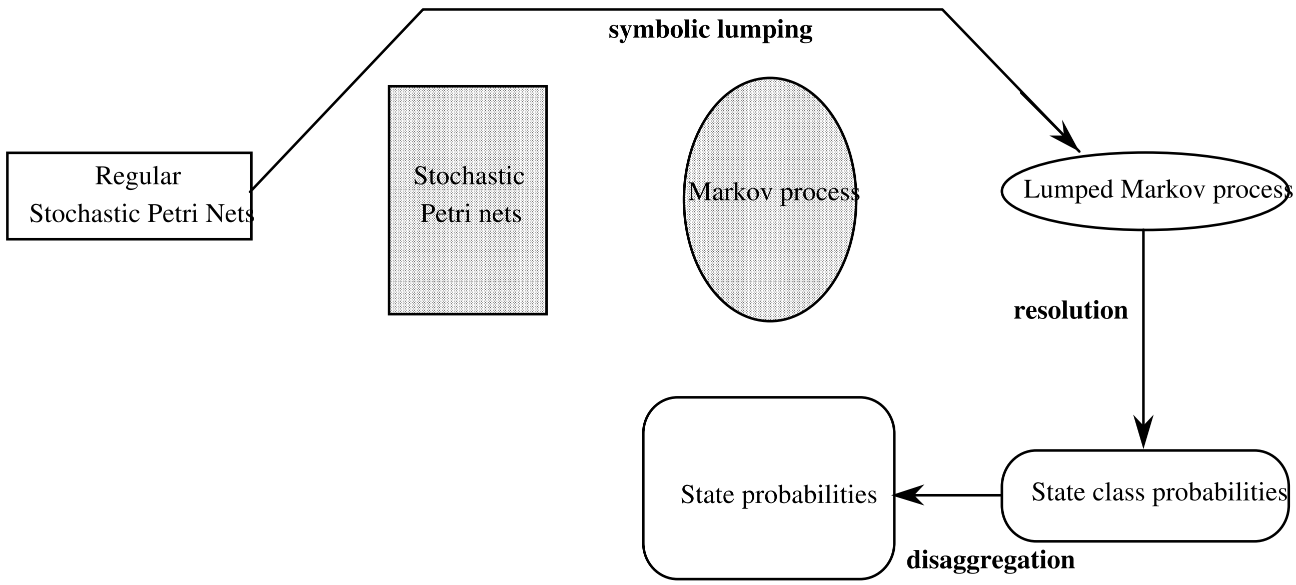
However, for any kind of model, obtaining performance results is not easy when complex systems are analyzed. Even when limiting to markovian models, the resolution is often quite difficult because of the size of the generated Markov process. Simulation techniques can be used, but they are very expensive and the accuracy of the result is not always guaranteed. Therefore, research has been done to provide more simple and exact analytic methods of resolution. Lumping techniques have been presented in [Kem 60]. They are based on an aggregation of states into classes that is performed once the Markov process has been constructed. The solution is then computed on the classes. The drawbacks of this method are twofold. On the one hand, the lumping phase generally depends on the ability of the analyst, as there is no algorithm that defines the partition of states into classes. On the other hand, it is not always possible to derive back the state probabilities from the class probabilities.



Because of their apparent symmetry properties, colored stochastic Petri nets seemed an interesting tool to be used as a basis for lumping. However, the aggregation of states proved to be difficult when working on general classes of colored nets [Chi 88]. There is no method avoiding construction of the whole reachability graph, and cumbersome equivalence relations that depend on the color functions of the net must be defined by the user, thus preventing a completely automatic analysis of the net. In fact, the representation power of colored nets is best exploited when the system to be modeled presents intrinsic symmetries so that functionalities of the system are not hidden in complex color transformation functions.

Comparable complexity problems due to the size of the reachability graph have been encountered in the study of untimed colored Petri nets. A successful approach has been presented in [Had 87] that took advantage of a completely symmetric model, the regular nets, to group the states into classes called symbolic markings.

The regular stochastic Petri nets have been introduced in [Dut 89]. They are very well suited to the modeling and analysis of symmetric systems. The originality of the model is that it allows a symbolic lumping of the states, which is in fact an a priori aggregation that can be performed without constructing the whole Markov process. The partition of states into classes is natural as it relies on the intrinsic symmetries of the model.



As the states within a class have the same probability, the disaggregation step is always possible, but the additional information it would bring to the user is not always significant. Actually, because of the symmetry, all the states within a class play the same part.

The improvements brought by the lumping are twofold. The gain exists not only for the size of the process to be built, but also for the resolution complexity, as the exact algorithms are $O(n^3)$ where n is the number of states of the process,a nd the complexity of the iterative methods depends on the number of arcs.

In the same way as we followed the approach in [Had 87] to extend stochastic Petri nets to a colored model, we follow the approach of [Ajm 84] to extend the timing of the model by introducing in it priorities and immediate transitions. Immediate transitions have higher priorities than timed transitions. This way, the semantics of the net is not modified by the addition of timing. The introduction of immediate transitions is particularly useful when one wants to model activities whose durations differ by orders of magnitude. The use of immediate transitions is convenient if by doing so, the number of states of the associated Markov chain model is reduced, thereby reducing the solution complexity. But the specification of the stochastic behavior of a model with immediate transitions requires the definition of random switches to solve the conflicts that may occur between immediate transitions [Ajm 84, Ajm 87b].

In this paper, we present the introduction of immediate transitions in regular stochastic Petri nets. To illustrate our approach , we have chosen to study a multiprocessor architecture that has already been completely analyzed by means of ordinary stochastic Petri nets [Ajm 84]. We show on this example how one can take advantage of regular stochastic Petri nets.

The paper is organized as follows. In the next section, we recall the definition of generalized colored stochastic Petri nets. Generalized regular stochastic Petri nets are introduced in Section 3. Section 4 contains the description of the multiprocessor architecture and the colored stochastic Petri net model. The lumping method is described in Section 5. At last in section 6, we discuss our results on the example.

II - GENERALIZED COLORED STOCHASTIC PETRI NETS

Generalized stochastic Petri nets (GSPN) [Ajm 84] are a tool well adapted to the modeling and the performance evaluation of distributed computer systems. In a GSPN, a firing delay is associated with each transition of priority zero. Firing delays are instances of random variables that have a negative exponential probability distribution. The probability that two timed transitions sample exactly the same delay time is zero, so that priority zero transitions are assumed to fire one at a time. The selection of the transition instance to fire among the set of the enabled ones follows a "race" policy [Ajm 85] (the transition that has drawn the least delay is the one that fires). Higher priority transitions are assumed to fire as soon as they are enabled in zero time (to be immediate). In case of conflict between immediate transitions, it is necessary to associate with transitions a random switch that allows the resolution of the conflict. The random switch probabilistically describes how an immediate transition is selected to fire. Thus, the stochastic behavior of the net is completely specified.

As the complexity of the system to be analyzed increases, the number of objects in the corresponding Petri net model makes it quite difficult to represent and, as a consequence, to read. Colored Petri nets [Jen 81] can be used to represent in a compact manner complex systems. Generalized colored stochastic Petri nets thus appear to be a natural extension of Generalized stochastic Petri nets.

A generalized colored stochastic Petri net [Chi 88] is defined as a 9-tuple :

$GCSPN = (P, T, C, W^+, W^-, W^h, \pi, M_0, Y)$, where

P is a finite set of places,

T is a finite set of transitions, such that $P \cap T = \emptyset$, $P \cup T \neq \emptyset$,

$C : P \cup T \rightarrow \boldsymbol{C}$, $\boldsymbol{C} \cap P = \emptyset$, $\boldsymbol{C} \cap T = \emptyset$ is a function mapping $P \cup T$ on a non-empty finite set of colors,

W^-, W^+, W^h : $P \times T \rightarrow F$ are the input, output and inhibition functions respectively,

where F is the set of functions $[C(P) \times C(T) \rightarrow N]$,

$\pi : T \rightarrow N$ is the priority function, mapping transitions into natural numbers,

$M_0 : [P \rightarrow [C(P) \rightarrow N]]$ is the initial marking

$Y : T \rightarrow [C(T) \times M \rightarrow R^+]$ is a possibly marking-dependent weight function associating a positive real number with each transition.

C(p) (resp. C(t)) is the color domain of p (resp. t).

As a consequence of the definition of the priority function, the priority level of a transition does not depend on its color instance.

If t is a timed transition, the value of Y(t)(c, M) is the firing delay associated to the color c instance of transition t in marking M, whereas if t is immediate, Y(t)(c, M) will give the probability that the color c instance of transition t is chosen to fire among all the transitions enabled in marking M.

Definition of the firing rule : a transition t is enabled for a color c in a marking M iff the following conditions are fulfilled :

(1) $\forall p \in P, \forall c' \in C(p), W^-(p, t)(c', c) \leq M(p, c')$,

(1') $\forall p \in P, \forall c' \in C(p)$, either $W^h(p, t)(c', c) = 0$, or $M(p, c') < W^h(p, t)(c', c)$.

which is the expression of the firing rule in a net without priorities

(2) $\forall t'$ with $\pi(t') > \pi(t), \forall c'' \in C(t'), \exists p \in P, \exists c' \in C(p)$,

$W^-(p, t')(c', c'') > M(p, c')$ or $0 < W^h(p, t)(c', c) \leq M(p, c')$, which means that no higher priority transition is enabled.

The introduction of immediate transitions in the model naturally partitions the states in two classes. The states in which an immediate transition is enabled, called vanishing markings, have a probability zero as the process spends no time in them. The states in which no immediate transition is enabled, called tangible states, have a positive probability. It is thus important that the aggregation of states derived from the colored net preserve this partition. With a general definition of colored nets, it is not always the case [Chi 88]. However, we will show that by imposing symmetries on the marking, the color functions and the timing, we can ensure that this property will be verified. A class of colored nets with symmetries is presented in the next section.

III - GENERALIZED REGULAR STOCHASTIC PETRI NETS

In [Dut 89], a definition of regular Petri nets was given that implied that all places and transitions had the same color domain. In this section, we give the definition of general regular nets in which this assumption is no longer true. We introduce priorities in our model, so that the possibility of immediate firing will not modify the semantics of the untimed model. We also present the transformation rule that maps general regular nets on nets in which all places and transitions have the same color domain. This mapping results in a simplified analysis of the model. Finally we present the definition of Generalized regular stochastic Petri nets in which a possibly null firing delay is associated to the transitions.

3.1 General regular nets

A general regular net, which will be simply called a regular net, is a net in which the color domain of a place or a transition is built on any Cartesian product of basic object domains.

3.1.1 Color domains

A color domain can be a set of undistinguished resources if the product is null, a set of objects if there is a single element in the product, and a set of associations between objects if the product is made of several elements.

Construction of the color domains :
Let $C = \{C_1, \ldots, C_n\}$ a set of basic object domains, with $C_i \cap C_j = \emptyset$. Let J an ordered subset of the set $I = \{1, \ldots, n\}$, $J = \{i_1, \ldots, i_k\}$ (for instance, $J = \{1, 3, 2\}$ and $J' = \{2, 1, 3\}$ are different ordered subsets).
We denote $C_J = \prod_{i \in J} C_i$, a cartesian product of object domains. If $J = \emptyset$, then $\prod_{i \in J} C_i = \langle \varepsilon \rangle$, where

ε is the neutral color.
An element $(c_{i_1}, \ldots, c_{i_k})$ of C_J will be denoted $\prod_{i \in J} c_i$.

Definition : color domain of a node.
Let $r \in P \cup T$. The color domain $C(r)$ of r is defined by $C(r) = C_{J(r)}$, where $J(r)$ is an ordered subset of I.

3.1.2 Color functions

Let J and J' two ordered subsets of $I = \{1, \ldots, n\}$. In this paragraph, we define the possible color functions between a place p and a transition t, with $C(p) = C_J$ and $C(t) = C_{J'}$. The set of such functions is denoted $F_{J,J'}$. We can notice that $F_{J,J'}$ is a subset of the set $\{f : C_J \times C_{J'} \rightarrow N\}$.
We proceed by induction on the size of J.

case 1 : $J = \emptyset$
Those functions are equivalent to valuated arcs in an ordinary net.
$F_{\emptyset, J'} = \{\widehat{b} / b \in N\}$, where $\widehat{b}(\varepsilon, c) = b$.
In the rest of the paper, this will be simply denoted b.

case 2 : J = {i}

We first define two basic color functions. The first function corresponds to a synchronization of all the objects in a class if it labels an input arc. It represents a diffusion to all the objects in a class if it labels an output arc.

$\forall$ J', $\forall$ a $\geq$ 0,

$a.S_i : C_i \times C_{J'} \rightarrow N$

$$a.S_i(c'_i, \prod_{j \in J'} c_j) = a.$$

The second function allows one to select one object whose behavior will be independent of that of the other objects of the class when firing the transition.

$\forall$ J' such that $i \in J'$, $\forall$ b $\geq$ 0,

$b.X_i : C_i \times C_{J'} \rightarrow N$

$$b.X_i(c'_i, \prod_{j \in J'} c_j) = Id(c'_i, c_i) = (\text{If } c_i = c'_i \text{ then b else } 0).$$

Now we can define the sets

$$F_{i,J} = \begin{cases} \{a_i . S_i, \quad a_i \geq 0\} & \text{if } i \notin J \\ \{a_i . S_i + b_i . X_i, \quad a_i \geq 0,\ a_i + b_i \geq 0\} & \text{if } i \in J \end{cases}$$

case 3 : J $\neq$ Ø

On each component of the product, the function behaves like an elementary one.

$$F_{J,J'} = \left\{ \prod_{i \in J} \langle a_i . S_i + b_i . X_i \rangle, \text{ with } a_i .S_i + b_i .X_i \in F_{i,J'} \right\}.$$

where $\prod_{i \in J} \langle a_i . S_i + b_i . X_i \rangle$ is a notation for $\langle a_{i_1} . S_{i_1} + b_{i_1} . X_{i_1}, \ldots, a_{i_k} . S_{i_k} + b_{i_k} . X_{i_k} \rangle$.

$$\left(\prod_{i \in J} \langle a_i . S_i + b_i . X_i \rangle \right) (c'_J, c_{J'}) = \prod_{i \in J} (\langle a_i . S_i + b_i . X_i \rangle (c'_i, c_{J'}))$$

Notice that X_i does not appear if $i \notin J'$. This comes from the fact that the firing of a transition cannot distinguish an object of C_i if this transition does not include C_i in its color domain. As a consequence, $\forall f \in F_{J,Ø}$, $\forall c, c' \in C_J$, $f(c, \varepsilon) = f(c', \varepsilon)$.

3.1.4 Regular nets

A regular net is a colored net whose color domains and functions are those defined in the former section.

Definition : A regular net $RN = \langle P, T, C, J, W^-, W^+, \pi, M_0 \rangle$ is defined by :

P, the finite set of places,

T, the finite set of transitions, $P \cap T = \emptyset$, $P \cup T \neq \emptyset$,

C the set of object classes : $C = \{C_1, \ldots, C_n\}$, with $C_i \cap C_j = \emptyset$

(we will denote $I = \{1, \ldots, n\}$ the ordered set of indexes),

$J : P \cup T \rightarrow P(I)$, where P(I) denotes the set of ordered parts of I

($C(s) = \prod_{j \in J(s)} C_j$ denotes the color domain of s. If $J = \emptyset$, $\prod_{j \in J} C_j = \{\varepsilon\}$),

$W^-(p, t)$, $W^+(p, t) \in F_{J(p), J(t)}$ the input and output functions,

$\pi : T \rightarrow N$ the priority function.

$M_0(p) \in F_{J(p),\emptyset}$ is the initial marking of the place p.

Notation : $M_0(p)(c, \varepsilon)$ will be simply denoted $M_0(p, c)$. As a consequence of the symmetry property on $F_{J,\emptyset}$, we have $\forall p \in P, \forall c, c' \in C(p), M_0(p, c) = M_0(p, c')$.

Definition of the firing rule : a transition t is enabled for a color c in a marking m iff the two following conditions are fulfilled.

(1) $\forall p \in P, \forall c' \in C(p), W^-(p, t)(c', c) \leq M(p, c')$

which is the expression of the firing rule in a net without priorities

(2) $\forall t'$ with $\pi(t') > \pi(t)$, $\forall c'' \in C(t')$, $\exists p \in P$, $\exists c' \in C(p)$,

$W^-(p, t')(c', c'') > M(p, c')$, which means that no higher priority transition is enabled.

3.2 Symmetry properties of regular nets

The introduction of regular nets has been motivated by the fact that many symmetries existed in most of the system studied. In this section we present some symmetry properties of this class of colored nets.

Definition : Let $\zeta = \{s = \langle s_1, \ldots, s_n \rangle / s_i$ is a permutation on $C_i\}$, a subgroup of the permutations on $C_1 \times \ldots \times C_n$.

($s = \langle s_1, \ldots, s_n \rangle$ will be also denoted $\prod_{i=1}^{n} s_i$).

$\forall s \in \zeta, \forall J' \subseteq I, \forall c = \prod_{i \in J'} c_i, \forall J \subseteq J', s^J(c) = \prod_{i \in J} s_i(c_i)$.

When no confusion on J is possible, $s^J(c)$ can be simply denoted s(c).

Definition : let M a marking, $s \in \zeta$ a permutation. Then s.M is a marking defined by :

$$\forall p \in P, \forall c \in C(p), \; s.M(p, c) = M(p, s^{J(p)}(c)).$$

Proposition : s.M defines an operation of the group ζ on the set of markings, i.e.,

$\forall$ s, s' $\in \zeta$, $\forall$ M a marking, (s $\circ$ s').M = s.(s'.M)

id.M = M.

Definition : let (t, c) a colored transition, s $\in \zeta$ a permutation. Then s.(t, c) is a colored transition defined by :

$$\forall\, t \in T,\ \forall\, c \in C(t),\ s.(t, c) = (t, s^{J(t)}(c))$$

Proposition : s.(t, c) defines an operation of the group ζ on $\{(t, c) / t \in T, c \in C(t)\}$.

We now give the expression of the basic theorem in general regular nets with priorities. This theorem expresses the fundamental symmetry property of regular nets.

Theorem : $\forall$ M, M' two markings, $\forall$ t $\in$ T, $\forall$ c $\in$ C(t), $\forall$ s $\in \zeta$,

$$M[(t, c) > M' \,/\, s.M[s.(t, c) > s.M'.$$

The firing of a transition is preserved by the operation of a permutation on the marking and the color of the transition.

We introduce another operation of ζ useful for the definition of our stochastic model.

Definition : let Ω a set of colored transitions, s $\in \zeta$ a permutation. Then s.Ω is a set of colored transitions defined by : $s.\Omega = \{s.(t, c) / (t, c) \in \Omega\}$.

Proposition : s.Ω defines an operation of the group ζ on the powerset of colored transitions.

3.3 Normalization of a regular net

The normalization of a net maps a net in which the color domain of a place or a transition is built on any Cartesian product of basic object domains on a net in which the color domains of all places and transitions are the Cartesian product of all basic object domains. The normalization does not modify the semantics of the net, and it results in a simplification of the analysis of the model.

<u>Definition</u> : normalization of a function. Let $f \in F_{J,J'}$ defined by

$$f = \prod_{i \in J} \langle a_i . S_i + b_i . X_i \rangle , \text{ with } a_i . S_i + b_i . X_i \in F_{i,J'}$$

The normalized color function $\hat{f}$ is defined by

$$\hat{f} \in F_{I,I} , \ \hat{f} = \prod_{i \in I} \langle \hat{a}_i . S_i + \hat{b}_i . X_i \rangle .$$

with if $i \in J$ then $\begin{cases} \hat{a}_i = a_i \\ \hat{b}_i = b_i \end{cases}$

else $\begin{cases} \hat{a}_i = 1 \\ \hat{b}_i = 0 \end{cases}$

<u>Definition</u> : Let R a regular net. The normalized regular net $\tilde{R}$ associated to R is defined by :
$\tilde{P} = P$,
$\tilde{T} = T$,
$\tilde{C} = C$,
$\forall$ s, $\tilde{J}(s) = I$,
$\tilde{W}^+(p, t) = \widehat{W}^+(p, t)$, $\tilde{W}^-(p, t) = \widehat{W}^-(p, t)$,
$\tilde{\pi} = \pi$,
$\tilde{M}_0(p) = \widehat{M}_0(p)$.

<u>Definition</u> : restricted color.
Let J an ordered subset of I, and $c = \prod_{i \in J} c_i$ an element of C_J . Then, for any ordered $J' \subset J$, the restriction of c to $C_{J'}$ is defined by $c^{J'} = \prod_{i \in J'} c_i$.

<u>Remark</u> : Let M a marking in R, $\widehat{M}$ a marking in $\tilde{R}$.
$\forall p \in P, \forall c \in C, \widehat{M}(p, c) = M(p, c^{J(p)})$.

<u>Property</u> : Semantics preservation. $\forall (c_1, \ldots, c_k) \in (C_I)^k$,

(1) in $\tilde{R}$, $\widehat{M}_0 [(t_1, c_1), \ldots, (t_k, c_k) > M \Rightarrow$ in R, $M_0 [(t_1, c_1^{J(t_1)}), \ldots, (t_k, c_k^{J(t_k)}) > M'$, with $\widehat{M'} = M$.

(2) in R, $M_0 [(t_1, c_1^{J(t_1)}), \ldots, (t_k, c_k^{J(t_k)}) > M$
$\Rightarrow$ in $\tilde{R}$, $\forall c'_i$ such that $c'^{J(t_i)}_i = c_i^{J(t_i)}$, $\widehat{M}_0 [(t_1, c'_1), \ldots, (t_k, c'_k) > \widehat{M}$.

Corollary :
Let G the reachability graph of R, $G = <V, A>$, $\tilde{G}$ the reachability graph of $\tilde{R}$, $\tilde{G} = <\tilde{V}, \tilde{A}>$.
The elements of the set of vertices V are the markings, and A is the set of arcs.

(1) $\exists$ a unique bijection $\varphi : V \rightarrow \tilde{V}$, such that $\varphi(M) = \widehat{M}$.

(2) For each arc $M[(t, c) > M' \in A$, we have
$\forall c' \in C_I$ such that $c'^{J(t)} = c$, $\widehat{M}[(t, c') > \widehat{M}' \in \tilde{A}$.

(3) $\forall c \in C(t)$, if $\widehat{M}[(t, c) > \widehat{M}' \in \tilde{A}$, then $M[(t, c^{J(t)}) > M' \in A$.

3.4 Generalized Regular Stochastic Petri Nets

Generalized Regular Stochastic Petri Nets are a timed extension of Regular Petri nets in which transitions are either immediate, or have an exponentially distributed firing delay. To preserve the symmetry of the underlying model, we impose that all the color instances of a transition have the same firing delay. This is not a very restrictive condition as all the objects of a class have similar behaviors and it is thus natural to consider that they require the same service.

Notation : we denote $T_n = \{t \in T, \pi(t) = n\}$, the set of all priority n transitions.
$\overline{M}(p) = \sum_{c \in C(p)} M(p, c)$ is the total number of tokens in p, whatever the color.

Preliminary explanations : Let M a reachable marking of a regular net R. If M is not a dead marking, then there exists a single i such that the set of the transitions enabled in the marking M is included in $\{(t, c) / t \in T_i, c \in C(t)\}$. If i is equal to zero the stochastic behavior of the net is entirely determined by the firing rates of the enabled transitions. If i is positive, the specification of the stochastic behavior requires the definition of a switch table. In this table, a probability to fire is associated to any enabled transition, depending on the other enabled transitions.

Thus the stochastic behavior is completely specified by :

- for any timed transition t, a firing rate that possibly depends on the marking M (denoted by $\lambda(t)(\overline{M})$).
- A switch table for some subsets Ω of immediate transitions (denoted ST_Ω). These switch tables will be grouped according to the priority level (denoted ST_i). In fact, we do not need to specify all the subsets of $\{(t, c) / t \in T_i, c \in C(t)\}$ but only the sets of transitions enabled in the same marking (denoted $D_F(ST_i)$). The union of all these tables will be denoted ST.

Definition : A GRSPN is defined by (RN, λ, ST), where

RN is a regular stochastic Petri net,

$\lambda : T_0 \to [N^p \to R^+]$, such that $\lambda(t)(\overline{M})$ is the possibly marking dependent weight associated to the transition t of priority 0.

$ST = \bigcup_{i=1}^{max} ST_i$ is the set of switching tables, with max being the maximum priority level in the model,

Let $P^*(\{(t, c) \mid t \in T_i, c \in C(t)\})$, the set of non-empty parts of $\{(t, c) \mid t \in T_i, c \in C(t)\}$.

Then $D_F(ST_i) \subset P^*(\{(t, c) \mid t \in T_i, c \in C(t)\})$ is the domain of ST_i , where

$ST_i = \bigcup_{\Omega \in D_F(ST_i)} ST_\Omega$ is the switching table at the priority level i,

$ST_\Omega : \Omega \to [N^p \to R^+]$.

Conditions on the model :

(1) $\forall M \in G$, either $T_0 \in \{(t, c) \mid M[(t, c) > \}$, or $\exists$ i, $\{(t, c) \mid M[(t, c) > \} \in D_F(ST_i)$.

Either all the transitions enabled by a reachable marking M are timed, or a random switch must be specified for the set of conflicting transitions in M.

(2) $\forall$ i, $\forall \Omega \in D_F(ST_i)$, $\forall M \in N^p$, $\sum_{(t, c) \in \Omega} ST_\Omega(t, c)(M) > 0$.

Among the transitions that are in conflict in a marking, one at least has a positive probability to fire.

(3) $(\Omega \in D_F(ST_i) \Rightarrow s.\Omega \in D_F(ST_i)$ and $(ST_{s.\Omega}(s.(t, c)) = ST_\Omega(t, c))$.

Given a subset specified by a switch table, the operation of a permutation on the colored transitions leads to a new subset that must be also specified. Moreover the functions of the original table are preserved by the operation of the permutation on the colored transitions.

Implicit conditions (included in the definition) :

(4) we impose that all the color instances of a transition have the same firing rate.

(5) all the functions in a switch table, and the firing rates depend only on $\overline{M}$ and not on M.

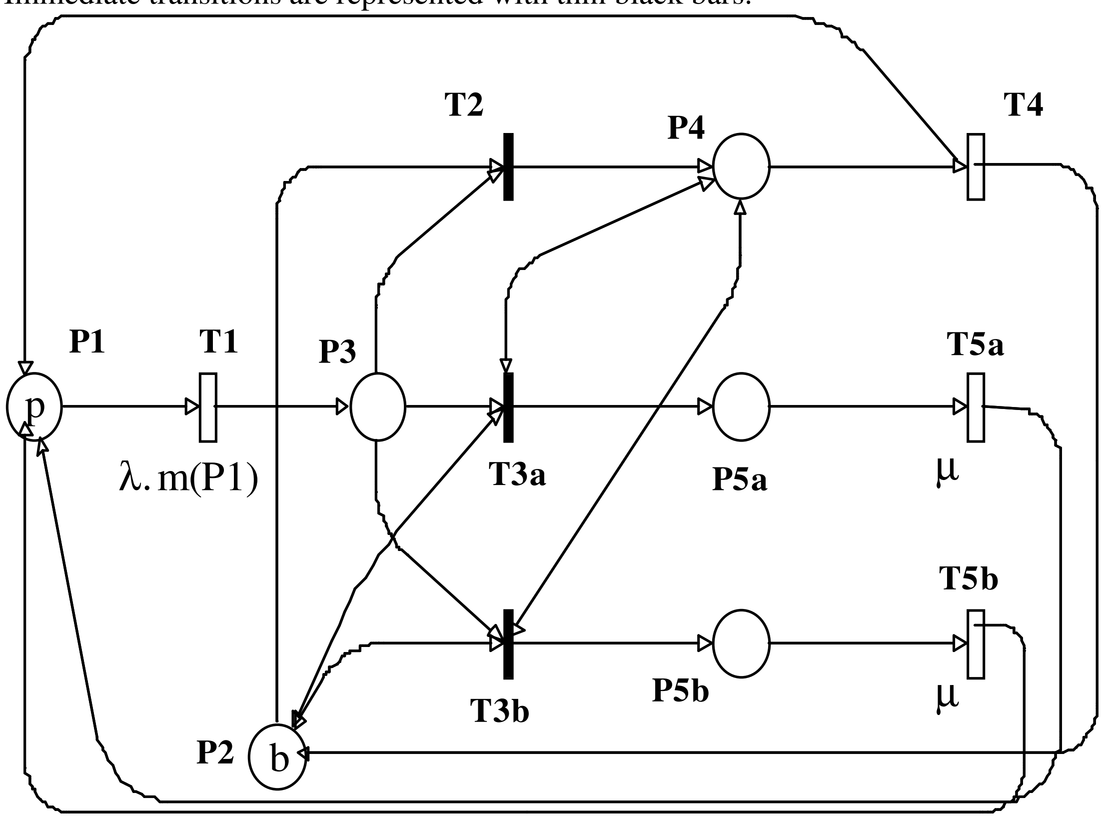
The next section describes a multiprocessor architecture we will analyze with our lumping technique. A stochastic Petri net model is presented, together with a colored stochastic Petri net of the same system.

IV - A MULTIPROCESSOR ARCHITECTURE

Our example is derived from the multiprocessor architecture presented in [Ajm 84].

In that paper, it has been shown that generalized stochastic Petri nets were well suited to the performance evaluation of a multiprocessor architecture. The same multiprocessor architecture was represented by several more and more simple models. The model finally obtained was rather concise, yet structurally depending on the number of buses. However, the simplification of the model required quite a good skill both in the comprehension of the system and the handling of Petri net models. Here we present a regular Petri net model of the same architecture. The introduction of color functions allows us to obtain a more intuitive model, which structure does not depend on the parameters of the system (number of buses, processors, memories). Description of the models :

A set of p processing units (place P1) cooperate by exchanging messages through a set of m common memories that can be reached through a network of b buses (place P2). A processor executes in its private memory for an exponentially distributed random time with average $1/\lambda$ before issuing (firing of T1) an access request directed to one of the common memories in the system. A processor that wants to issue a request will have to wait if no bus is available (place P3). The tokens in P4 represent the processors using one of the common memories, and the tokens in P5 the processors that have issued a request to a busy memory. The durations of accesses to common memories are independent exponentially distributed, random variables with average $1/\mu$. We first recall the Petri net model in [Ajm 84], in case of p processors, 3 buses and at least two memories, then we give the regular net model. Immediate transitions are represented with thin black bars.



A token in P3 represents a processor that has issued a request to a memory, and the number of tokens in P4 gives the number of busy memories. If P4 is empty, only T2 is enabled. If there is at least one token in P4 then T2, T3a, T3b are in conflict, thus requiring the definition of a random switch. A processor issues a request to a free memory with probability (1 - M(P4)/m). If there is exactly one token in P4, either P5a or P5b will represent the queue for the busy memory. So if neither is marked, there is an equal probability of firing T3a or T3b, whereas if one is marked, only the corresponding transition can be fired. If there are two marks in P4, T3a and T3b have the same firing probability, meaning that a processor has an equal probability of accessing either of the busy memories. Notice that as a processor needs a bus to issue his request, there can be no more than two busy memories at the time the request is issued.
We give the definition of the switch using GreatSPN formalism (#P means the marking of place P, "ever" gives the value of the rate when none of the former conditions is true).

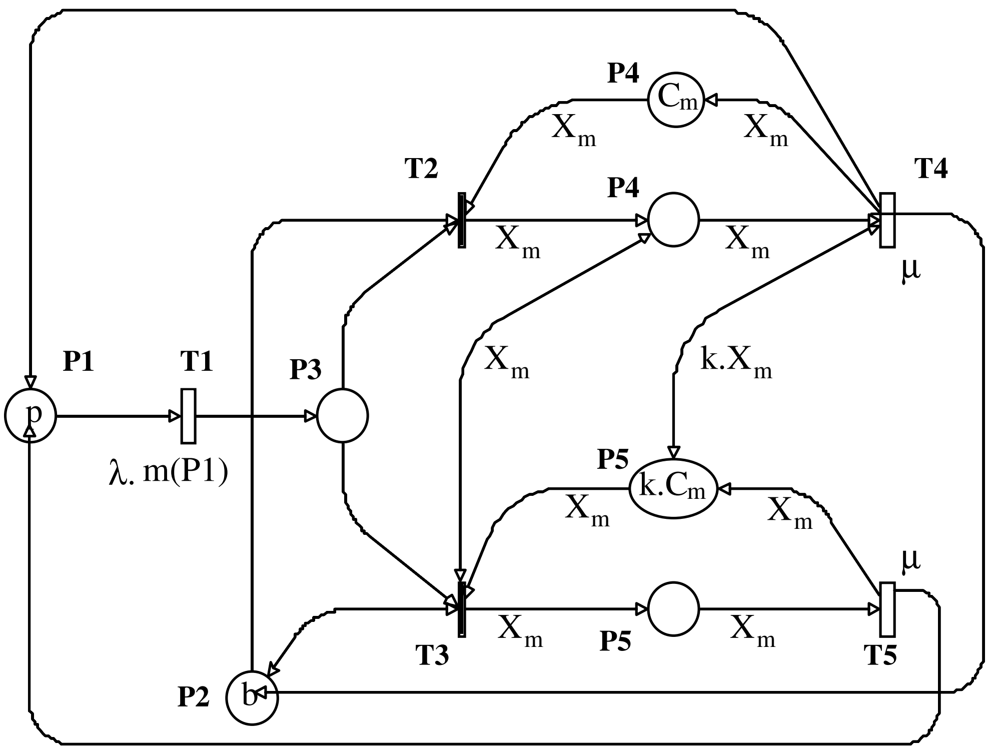
switch : T2 : ever 1 - #P4/m ;
T3a : when (#P4 = 1 & #P5b > 0) : 0 ; when (#P4 = 1 & #P5a > 0) : #P4/m ; ever #P4/2m ;
T3b : when (#P4 = 1 & #P5a > 0) : 0 ; when (#P4 = 1 & #P5b > 0) : #P4/m ; ever #P4 /2m ;

The marking of P4 represents the number of busy memories. So if no processor is awaiting a memory, T4 is fired with a rate μ*M(P4). If there are as many busy memories as awaited memories, then T4 is not enabled and the end of an access will be represented by the firing of T5a or T5b. In fact, the firing of T5 means that the processor awaiting in P5 immediately takes the memory that has just been liberated and replaces the token in P4 meaning that the memory is busy again.

rate of T4 : when (#P4 > 2 & #P5a > 0 & #P5b > 0) : μ * (#P4 - 2) ;
when (#P4 • 2 & #P5a > 0 & #P5b > 0) : 0 ;
when (#P5a > 0 & #P5b = 0) or (#P5a = 0 & #P5b > 0) : μ * (#P4 - 1) ; ever μ * #P4 ;

The construction of such a model is not natural and requires a great ability in the understanding of the system and the handling of Petri nets. Only a skilled user could produce such a model. Moreover, the modification of the number of buses changes the structure of the model, and requires to redefine the switch and the rate of T4.

We now give a more intuitive colored Petri net model of the same system :



C_m is the set of shared memories, with $| C_m | = m$. The value of k is any integer verifying $k \geq p - 1$ (k is the capacity of the queue for one memory). The contention for memories is represented using color functions. X_m is the identity function applied to the color set C_m. It is used to identify the different memories. $\overline{P4}$ models the free memories, and less than k color m_i tokens in $\overline{P5}$ means that at least one processor is waiting for m_i.

Switching table definition :

For any marking of the net in which a processor tries to access a memory, either no bus is available and neither T2 nor T3 is enabled, or a processor can access any memory with the same probability. The set of conflicting transitions in an ordinary marking M is given by

$\Omega_M = \{(T2,\ c) \mid M(p4,\ c) = 0\} \cup \{(T3,\ c) \mid M(p4,\ c) > 0\}$.

We can notice that for one color c, the choice between T2 and T3 is deterministic, as (T2, c) and (T3, c) are not simultaneously fireable.

For any marking M, we have $| \Omega_M | = m$. All the memories are accessed with the same probability. Therefore, for all reachable M

$\forall\, x \in \Omega_M,\ ST_{\Omega_M}(x) = \frac{1}{m}$.

The switch can thus be defined by a unique formula, whatever the marking. This is a specific property of the example, not a general property of GRSPN's. Moreover, the definition of the switch remains true for any value of the parameters of the model.

V - STATE AGGREGATION

When trying to solve large Markov chains, several techniques can be used. Some approximate techniques, such as simulation, are always possible, but they are generally very expensive, and the user may have to face some accuracy problems. As far as an exact resolution is concerned, the most frequently used technique is an aggregation of states that preserves the markovian property of the original process. A strong lumping condition has been defined in [Kem 60] under which an aggregated process was markovian. Unfortunately, the condition can be applied only once the partition of states into classes has been performed. No algorithm exists that allows one to regroup states in a way that ensures that the lumping condition will be verified. The definition of a sound partition relies on the skill of the analyst who generally uses the intrinsic symmetries of the model.

Because of their symmetries, the regular nets allow one to define a priori the partition of states into classes called symbolic markings, thus avoiding the construction of the whole reachability graph. A lumped process can be directly constructed, from which class probabilities can be computed. Moreover, all the states within a class have the same probability.

In this section, the definition of a symbolic marking, and the construction of the lumped Markov process are presented. We also explain how the original probabilities can be calculated.

5.1 Symbolic markings

For the sake of simplicity, we will only consider in this section nets in which all places and transitions have the same color domain. As we have already shown, this is not a restriction. We define here the way of building a symbolic marking.

Definition : Let $\zeta = \{s = <s_1, \ldots, s_n>, s_i \text{ is a permutation on } C_i\}$, and $reg(M) = \{s.M, s \in \zeta\}$. The sets reg(M) define a partition of the set of markings that induces an equivalence relation **R** :

$$M \, \mathbf{R} \, M' \, / \, reg(M) = reg(M').$$

The equivalence classes of the relation **R** are called symbolic markings, denoted by M.

Optimal representation of a symbolic marking :

The basic principle for representing a symbolic marking consists in grouping in a subclass all the objects of a class that have the same marking i.e, the marking is left unchanged when permuting two objects of the subclass. The identity of the objects in a subclass is then forgotten and only the number of objects is taken into account for each subclass. In this goal, each object class C_i is partitioned in a number of subclasses, $C_i = \{C_{i,1}, \ldots, C_{i,si}\}$, such that all the objects in a subclass have the same marking. The cardinalities of each subclass, which values are in N^*, verify
$\sum_{j=1,si} |C_{i,j}| = |C_i|$. The marking of each place is then similar to an ordinary marking where the subclasses are considered as objects. Moreover the grouping is maximal, i.e., the objects belonging to two different subclasses do not have the same marking.

The representation of symbolic markings as defined above is unique within a permutation $<p_1,\ldots,p_n>$ of the set of subclasses. However, it is possible to define and to calculate a canonical representation for each symbolic marking by an adequate ordering [Had 87].

Notice that the decomposition of subclasses is local to a symbolic marking. Thus the subclass $C_{i,j}$ appearing in a symbolic marking M and the same subclass $C_{i,j}$ appearing in a symbolic marking M' do not have related meanings.

Notation In M, we will denote the partition of a class C_i in $\{C_{i,1}, \ldots, C_{i,si}\}$by :
$C_i = \{C_{i,1}, \ldots, C_{i,si}\}$ and when confusion may arise $|C_{i,j}|_M$ will denote the cardinality $|C_{i,j}|$ in M.

5.2 Symbolic firing rule

In order to build a symbolic graph, we first define a symbolic firing rule on the symbolic markings which must be sound i.e., an ordinary marking enables a colored transition if and only if its symbolic marking enables an equivalent symbolic firing and the ordinary marking obtained by the firing belongs to the symbolic marking obtained by the symbolic firing. The primary effect of the symbolic firing will be to split each instantiated subclass in two subclasses, one with the object instantiated in the underlying firing and the other with the objects remaining in the subclass. Thus we formally define this splitting.

Definition 1 : Let M a symbolic marking. Then $M[C_{1,u1}, \ldots, C_{n,un}]$ is a marking defined by :
- If $|C_{i,ui}| > 1$ then
  $C_i = \{C_{i,1}, \ldots, C_{i,si}, C_{i,si+1}\}$ with
  $|C_{i,si+1}| = |C_{i,ui}|_M - 1$, $|C_{i,ui}| = 1$, $|C_{i,j}| = |C_{i,j}|_M$ for any $j \neq ui$ and $\neq si+1$
  else the partition of C_i is unchanged
- The marking of the old subclasses is unchanged and the marking of the new subclass $C_{i,si+1}$ is the same as the one of $C_{i,ui}$.

Notice that in $M[C_{1,u1}, \ldots, C_{n,un}]$ the grouping is not always maximal and that even if the grouping is maximal the representation is not always canonical. But it does not matter since this symbolic marking is just an intermediate marking and it will not appear in the symbolic graph.

The instantiation of a transition in a symbolic firing will be made by choosing a subclass per class instead of an object per class in an ordinary firing. Thus we must define the value of the colored functions for subclasses. This definition is the same as the one for the objects. In the case where an instantiated subclass has more than one object the symbolic firing should be enabled for the object instantiated in the underlying firing and for the other objects of the subclass which are not instantiated. Thus the definition should be different but since we apply our definitions on split markings, this case never appears.

Definition 2 : Let M a symbolic marking. Then :
$\langle a_i.S_i + b_iX_i \rangle$ is a function from $C_i \times \prod_{j=1,n} C_j \rightarrow N$ and
$\langle a_i.S_i + b_iX_i \rangle (C_{i,vi} , (C_{1,u1}, \ldots, C_{n,un}))$ = If $u_i \neq v_i$ then a_i else (a_i+b_i)

We extend this definition to a definition for the general class of functions $F_{I,I}$.

Definition 3 Let M a symbolic marking. Then :
$\prod_{j=1,n} \langle a_i.S_i + b_iX_i \rangle$ is a function from $\prod_{i=1,n} C_i \times \prod_{i=1,n} C_i \rightarrow N$ and
$\prod_{j=1,n} \langle a_i.S_i + b_iX_i \rangle ((C_{1,v1}, \ldots, C_{n,vn}) , (C_{1,u1}, \ldots, C_{n,un})) =$
$\prod_{j=1,n} (\langle a_i.S_i + b_iX_i \rangle (C_{i,vi} , (C_{1,u1}, \ldots, C_{n,un})))$

Let M_j a symbolic marking, t a transition, and $(C_{1,u1}, \ldots, C_{n,un})$ a tuple of subclasses such that
$C_{i,ui} \in C_i$. $C_{i,ui}$ is the distinguished subclass of C_i for the firing of t.

Definition 4 : t is enabled from M for $(C_{1,u1}, \ldots, C_{n,un})$ iff :

(1) $\forall\, p \in P$,
$M[C_{1,u1}, \ldots, C_{n,un}]\,(p, C_{1,v1}, \ldots, C_{n,vn}) \geq W^{-}(p, t)\,((C_{1,v1}, \ldots, C_{n,vn}), (C_{1,u1}, \ldots, C_{n,un}))$

(2) $\forall\, t'$ with $\pi(t') > \pi(t)$, $\forall\, (C'_{1,u1}, \ldots, C'_{n,un})$, $\exists\, p \in P$, $\exists\, (C_{1,v1}, \ldots, C_{n,vn})$,
$M[C'_{1,u1}, \ldots, C'_{n,un}]\,(p, C_{1,v1}, \ldots, C_{n,vn}) < W^{-}(p, t)\,((C_{1,v1}, \ldots, C_{n,vn}), (C'_{1,u1}, \ldots, C'_{n,un}))$

The symbolic marking M' obtained by the firing $t(C_{1,u1}, \ldots, C_{n,un})$ is calculated with the three following steps :

Step 1 : We apply the incidence functions on $M[C_{1,u1}, \ldots, C_{n,un}]$ giving a new symbolic marking M_1

$\forall\, p \in P$,

$$M_1(p, C_{1,v1}, \ldots, C_{n,vn}) = M[C_{1,u1}, \ldots, C_{n,un}]\,(p, C_{1,v1}, \ldots, C_{n,vn}) + W(p, t)\,((C_{1,v1}, \ldots, C_{n,vn}), (C_{1,u1}, \ldots, C_{n,un}))$$

Step 2 : as the grouping of states may not be maximal in M_1, it consists in grouping all the subclasses that have same markings giving a new marking M_2 . In fact, only the split subclasses may be equivalent to previously existing ones.

Step 3 : calculation for M_2 of the canonical representative marking M'.

5.3 Graph of symbolic markings :

5.3.1 Construction :

The algorithm for constructing the graph of reachable symbolic markings (GSM) is different from the construction of the ordinary reachability graph (RG) only by the firing rule and the labels of the arcs that are made of the transitions and the tuples of firing subclasses, whereas in RG we have the transition and the tuple of firing objects. Notice that the initial symbolic marking only contains the initial ordinary marking because of the symmetry of the initial marking.

5.3.2 Some properties of the graph of symbolic markings :

Many properties have been proved on the graph of symbolic markings, such as quasi-liveness, and the possibility of finding a home state. However, we will only present some properties that are useful for the proof of the algorithm that computes the symbolic marking probabilities. All the proofs of the propositions given here are in [Had 87] and will not be repeated.

Notations : Let $C_j = \{C_{j,1}, \ldots, C_{j,k}\}$ the partition of the class C_j in the symbolic marking M. Let M be an ordinary marking of M. We will denote by $M.reg(c_j) = C_{j,q}$ the subclass to which c_j belongs in the marking M. Notice that $M(p, C_{1,u1}, \ldots, C_{n,un}) = M(p, (c_1, \ldots, c_n))$, for $M \in M$ and $M.reg(c_j) = C_{j,uj}$.

Proposition 1 : The enabling of a transition is equivalent for an ordinary marking and for the associated symbolic marking :

$\forall M \in M, M[t(c_1, \ldots, c_n)> \,/\, reg(M)[t(M.reg(c_1), \ldots, M.reg(c_n))>$.

Proposition 2 : The firing property comprises two steps :

$\forall M, M' , M[t(c_1, \ldots, c_n)>M' \Rightarrow reg(M)[t(M.reg(c_1), \ldots, M.reg(c_n))>reg(M')$.

The reciprocal property is not true for any couple (M, M'). Yet we have :

$\forall$ M, M' two symbolic markings, $\forall M' \in reg^{-1}(M')$,

$M[t(C_{1,u1}, \ldots, C_{n,un})>M' \,/$

$\exists M \in reg^{-1}(M), \forall (c_1, \ldots, c_n)$ such that $M.reg(c_k) = C_{k,uk}$, $M[t(c_1, \ldots, c_n)>M'$.

Proposition 3 : The reachability property is equivalent for the ordinary and the symbolic markings :

$M \in RG \,/\, reg(M) \in GSM$.

Because of the homogeneous timing of the different color instances of a transition, the partition of states into symbolic markings avoid the construction of non-uniform classes [Chi 88] that comprise both vanishing and tangible markings. If one state in a class enables an immediate transition, then it is the same for all the states in the class.

5.4 Calculation of the solution

Once the symbolic graph has been computed, we apply a similar method as the one in [Ajm84]. This method consists in solving an embedded Markov chain in which the transition instants correspond to a change of state. The solution of the original process is then obtained by a simple transformation. We first present the resolution algorithm on the symbolic reachability graph, then we give a sketch of the proof.

The first step of the algorithm consists in the computation of the transition probability matrix A^*, in which the entry (I, J) is the probability of going from the symbolic marking I to the symbolic marking J, disregarding the notion of time.

$$A^*_{(I,\ J)} = \frac{\prod_{a \in \boldsymbol{A}_{(I,\ J)}} \lambda_{t(a)}(\bar{I}) \,.\, |D_1^a| \ldots |D_n^a|}{\prod_{a \in \boldsymbol{A}_I} \lambda_{t(a)}(\bar{I}) \,.\, |D_1^a| \ldots |D_n^a|} \qquad \text{I tangible,}$$

where $\boldsymbol{A}_{(I,\ J)}$ is the set of symbolic arcs leading from I to J, $\boldsymbol{A}_I$ is the set of symbolic arcs out of I (including the loops on I), and $D^a{}_i$ is the subclass of C_i instantiating the transition that labels a.

$$A^*_{(I,\ J)} = \frac{\prod_{a \in \boldsymbol{A}_{(i.k)J}} ST_{\Omega(i.\ k)}(t(a))(\overline{i.\ k})}{\prod_{a \in \boldsymbol{A}_{(i.k)}} ST_{\Omega(i.\ k)}(t(a))(\overline{i.\ k})} \qquad \text{I vanishing,}$$

$\boldsymbol{A}_{(i.\ k)\ J}$ is the set of arcs leading from one marking (i.k) of I to any marking in J. Because of the symmetries, the value of the sum does not depend on the choice of (i. k).

$$A^*_{(I,\ I)} = - \prod_{I \neq J} A^*_{(I,\ J)} \,.$$

The solution of the embedded Markov chain is obtained by solving the linear system Y^* . A^* = 0, Y^* being a probability vector. The state probabilites of the original process are obtained from Y* by a simple transformation [Ajm 84]. The vanishing states have a probability zero, whereas the probability of the tangible state I is obtained by :

$$P^*_I = Y^*_I \,.\, \frac{\dfrac{1}{\prod_{a \in \boldsymbol{A}_I} \lambda_{t(a)}(\bar{I}) \,.\, |D_1^a| \ldots |D_n^a|}}{\sum_{J \in GSM} \dfrac{Y^*_J}{\prod_{a \in \boldsymbol{A}_J} \lambda_{t(a)}(\bar{J}) \,.\, |D_1^a| \ldots |D_n^a|}}$$

The algorithm can be proved by showing that the solutions of the embedded Markov chains are the same in the ordinary and the aggregated case, and that the transformation coefficients are equal. The first step consists in showing that all the markings within a symbolic marking have the same probability. The linear system can thus be reduced, and the solution is still a probability vector. The reduced system is exactly the one obtained with the algorithm we have presented. The equality of the transformation coefficients is derived from the correspondence between the arcs in the ordinary and the symbolic reachability graph (Proposition 2 of Section 5.3.2).

5.5 Disaggregation of states

One of the major drawbacks of markovian lumping is that once the probability of a class of states has been calculated, it is not easy to derive the probability of one particular state within the class. However, the symmetry of our model allowed us to prove that all the states within a class have the same probability. As the cardinality of a class of states can be easily computed, the knowledge of the probability of either a state or its class are equivalent.
Yet because of the meaningful way of lumping the states, the disaggregation step is not always necessary. As the states within a class are symmetric, it can be more significant to the analyst to know for instance that one memory is being accessed by a processor than to know that the memory M is being accessed by the processor P. The improvement brought by our method depends on the cardinalities of state classes and is all the more important as the cardinalities of the color sets are great (factorial dependence).

VI - EXAMPLE

A complete performance evaluation of that system has already been presented in [Ajm 84]. We will not give again all the results, but we will show that the same results can be obtained with a colored model by generating only the symbolic reachability graph.

<u>Results</u> :

We just focus on one symbolic marking to show the correspondance with the ordinary markings.

	marking	probability
symbolic (P1, P2, P3, P4, $\overline{\text{P4}}$, P5, $\overline{\text{P5}}$)	(1, 1, 0, C2 + C3, C1, 2.C2, k.(C1+C3) + (k-2).C2) |C1| = 2, |C2| = |C3| = 1.	1.0641e-02
ordinary (P1, P2, P3, P4, P5a, P5b)	(1, 1, 0, 2, 2, 0)	5.3204e-03
	(1, 1, 0, 2, 0, 2)	5.3204e-03

The probabilities have been obtained for a load factor $\rho = \lambda/\mu = 0.3$.
This marking represents a state in which two memories are busy, two processors are waiting for the same busy memory, and the fifth processor executes in its private memory. The number of corresponding markings in the unfolded net would be 12.

The analysis of the model has been performed using GreatSPN software [Chi 87]. We compare the number of markings obtained for our model, and for the model in [Ajm 84]. For less than 3 buses, the number of markings in both models is exactly the same. This is due to the fact that a processor trying to access a memory needs a bus. So at the moment when it can ask for a memory, one at most is busy, and the coloration has no influence on the model. We present the results for a model with 3 buses, 4 memories and 5 processors. In fact, the number of symbolic markings does not depend on the number of memories, as far as $m \geq b$. An increase in the number of memories would modify only the cardinalities of the symbolic markings.

markings	tangible	vanishing	total
symbolic	19	10	29
ordinary	31	15	46

For the ordinary net, the number of markings actually constructed is 50 tangible and 32 vanishing markings. This is due to the timing definition, and the fact that the timing is checked only once the reachability graph has been constructed. As a consequence, some markings are constructed, that in fact are not reachable. Those markings are called inconsistent.

When the number of buses increases, the definition of the marking dependent rate associated with T4 in the ordinary net becomes quite complex, as it depends on the markings of three waiting places. It is almost impossible to still use that model. With our GRSPN model, the number of tangible markings for 4 buses, 8 processors and at least 4 memories would be 67, whereas the GSPN model would lead to 450 tangible markings, some of them inconsistent.

The probabilities obtained with the symbolic graph and the ordinary graph are exactly the same. The model in [Ajm 84] was already a great improvement in the complexity of the analysis. Besides an easier modeling, the GRSPN model provides a smaller reachability graph. We therefore believe that GRSPN's can be a useful extension of GSPN's.

VII - CONCLUSION

In this paper, we showed that a convenient lumping of states can be an efficient analysis method provided that the system to be studied has symmetry properties. Such systems can be modeled using a particular class of colored stochastic Petri nets for which the lumping of states can be done a priori without constructing the whole Markov process. Moreover, the method is exact and we showed by applying it to the performance evaluation of a multiprocessor architecture that the improvement in the complexity is all the more important as the cardinalities of the color classes are great.
A software tool is being developed to automatically construct the graph of symbolic markings. We should thus be able to obtain more complete results in the performance analysis of symmetric systems.

We now intend to analyze wider classes of colored nets. In this aim, our research directions will be twofold. We will investigate other classes of nets for which we believe that symbolic markings can be defined. However some classes of colored nets with particular symmetry types cannot be analyzed this way. Therefore, we will try to extend to the stochastic analysis of a model the reachability tree defined in [Hub 84] for colored nets.

References

[Ajm 84] M. Ajmone Marsan, G. Conte, and G. Balbo, "A class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems", in ACM Transactions on Computer Systems, vol. 2, no. 2, pp 93-122 (May 1984).

[Ajm 85] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani,"On Petri Nets with Stochastic Timing", in proc. International Workshop on Timed Petri Nets, pp 80-87, IEEE-CS Press, Torino, Italy (July 1985).

[Ajm 87a] M. Ajmone Marsan, G. Chiola, "On Petri Nets with Deterministic and Exponentially Distributed Firing Times", in Advances in Petri Nets '87, pp 132-145, G. Rozenberg ed., LNCS 266, Springer Verlag (1987).

[Ajm 87b] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte, "Generalized Stochastic Petri Nets Revisited : Random Switches and Priorities", in proc. International Workshop on Petri Nets and Performance Models, pp 44-53, IEEE-CS Press, Madison, WI, USA (August 1987).

[Chi 87] G. Chiola, "A Graphical Petri Net Tool for Performance Analysis", 3rd International Workshop on Modeling Techniques and Performance Evaluation, AFCET, Paris (March 1987).

[Chi 88] G. Chiola, G. Bruno, and T. Demaria, "Introducing a Color Formalism into Generalized Stochastic Petri Nets", in proc. 9th European Workshop on Application and Theory of Petri Nets, pp 202-215, Venezia, Italy (June 1988).

[Dug 84] J. B. Dugan, K. S. Trivedi, R. M. Geist, V. F. Nicola, "Extended Sochastic Petri Nets : Applications and Analysis", in Performance'84, 10th Int. Symposium on Computer Performance, pp 507-519, E. Gelenbe ed., Elsevier Science Publishers B. V., North Holland, Paris, France (December 1984).

[Dut 89] C. Dutheillet, S. Haddad, "Regular Stochastic Petri Nets", 10th International Conference on Application and Theory of Petri Nets, Bonn, West Germany (June 1989).

[Flo 86] G. Florin, S. Natkin, "One-Place Unbounded Stochastic Petri Nets : Ergodic Criteria and Steady-State Solution", in Journal of Systems and Software, vol. 1, no. 2, pp 103-115, Elsevier Science Publishing Co. (1986).

[Haa 87] P. J. Haas, G. S. Shedler, "Stochastic Petri Nets With Simultaneous Transition Firings", in proc. International Workshop on Petri Nets and Performance Models, pp 24-32, IEEE-CS Press, Madison, WI, USA (August 1987).

[Had 87] S. Haddad, "Une Catégorie Régulière de Réseaux de Petri de Haut Niveau : Définition, Propriétés et Réductions", Thèse de Doctorat, RR87/197, Laboratoire MASI, Université Paris VI, Paris, France (October 1987).

[Hol 85] M. A. Holliday, M. K. Vernon, "A generalized Timed Petri Net Model for Performance Analysis", in proc. International Workshop on Timed Petri Nets, pp 181-190, IEEE-CS Press, Torino, Italy (July 1985).

[Hub 84] P. Huber, A.M. Jensen, L.O. Jepsen, and K. Jensen, "Towards Reachability Trees for High Level Petri Nets", 5th European Workshop on Application and Theory of Petri Nets, Aarhus, Denmark (1984).

[Jen 81] K. Jensen, "Coloured Petri Nets and the Invariant Method", Theoretical Computer Science no. 14, pp 317-336 (1981).

[Kem 60] J. G. Kemeny and J. L. Snell, "Finite Markov Chains", Van Nostrand, Princeton, NJ (1960).

[Mey 85] J. F. Meyer, A. Movaghar, W. H. Sanders, "Stochastic Activity Networks :Structure, Behavior, and Application", in proc. International Workshop on Timed Petri Nets, pp 106-115, IEEE-CS Press, Torino, Italy (July 1985).

[Mol 81] M. K. Molloy, "On the Integration of Delay and Throughput Measures in Distributed Processing Models", Ph.D. Dissertation, University of California, Los Angeles, CA, USA (September 1981).

[Raz 84] R. R. Razouk, "The Derivation of Performance Expressions for Communication Protocols from Timed Petri Net Models", Computer Communication Review, vol. 14, no. 2, pp 210-217 (1984).

[Zub 85] W. M. Zuberek, "Performance Evaluation Using Extended Timed Petri Nets", in proc. International Workshop on Timed Petri Nets, pp 272-278, IEEE-CS Press, Torino, Italy (July 1985).