Chapter 10

Stochastic well formed Petri nets

Serge Haddad , Patrice Moreaux[1]

1. Introduction

Introducing high level Petri nets allows us to cope with the complexity of systems during the design phase. However, the lack of structure of expressions and arc or transition guard functions make using this concision impractical, or even impossible, for verification purposes. To overcome this drawback, several subclasses of high level nets have been proposed, among them the well formed Petri Net [CHI 93a]. Its numerous verification possibilities made possible a lot of system studies. One of the reasons of this theoretical success is the explicit symmetry derived from the syntax of the domains and the color functions.

Since we have the same modeling needs for stochastic systems, research works have focused on:

- bringing to the fore technics taking advantage of possible symmetries of the stochastic process;
- how to express constraints on the stochastic high level model leading to symmetries in the process.

Markov chain aggregation technique [KEM 60] is perfectly suited to the first goal. This method aims at substituting *macro-states* to states of the Markov

[1]LSV, École normale supérieure de Cachan, Cachan, France, (haddad@lsv.ens-cachan.fr)
LISTIC, Polytech'Savoie, Université de Savoie BP 80439 74944 Annecy le Vieux Cedex, France, (patrice.moreaux@univ-savoie.fr),

chain where each visit to a macro-state corresponds to a subset of states. To get a Markov chain of macro-states, it is necessary that the visit of two given states of one macro-state determines in the same way the future of the macro-process. If these hypotheses are met, and if the process is ergodic, then the macro-process stationary probability of a macro-state is the sum of the initial process stationary probabilities of its states. Moreover, similar hypothesis on the past of the process imply equiprobabilities of states inside a macro-state. In this case, the macro-process is a sufficiently complete abstraction to get the stationary distribution of the process. We present the Markovian aggregation in the section two.

Therefore, several attempts to exploit the aggregation technic have been proposed for high level nets. The first approach [ZÉN 85] starts from the reduced reachability graph of a colored net by the Jensen method [HUB 84]. However, there is no guaranty that the macro-process is a Markov chain. In the second approach [LIN 88], the macro-process is necessarily a Markov chain but no algorithm is provided for building the macro-process. A last, more achieved approach [CHI 88] groups states according to a partition deduced from a supposed symmetry and then refines the partitions until aggregation conditions are satisfied. Unfortunately, this technic requires to build the whole reachability graph.

In the third section, we show how to add a stochastic semantics to well formed Petri nets such that aggregation conditions are met for symbolic markings. We detail a model of a multiprocessor system with stochastic well formed Petri nets. This model is representative of kinds of applications for which this technic provides solutions to the combinatorial exploding of the Markov chain.

In the last section, we present the main ideas about the proof of the validity of aggregation. Moreover, we show how to directly compute the parameters of the macro-process from the symbolic graph. The multiprocessor model illustrates the complexity savings provided by the symbolic graph.

Figure 1 presents the principle of the approach. In the best case, an a posteriori aggregation requires an explicit (net) unfolding or an implicit one when generating the state graph, and the underlying Markov chain. Beside, the direct solving of this chain is often impossible due to its size. In contrast, the symbolic graph allows for the a priori build of the aggregated chain. Moreover, the aggregates -the symbolic markings- may be interpreted for the modelling point of view and are usually sufficient to get significant performance indices. Al last, the steady state probability of each state may be computed from the aggregated solution and from the symbolic graph.

Other theoretical developments for stochastic well formed nets enlarge the applicability of this model. In the next chapter, the Markovian aggregation is combined with tensor decomposition which reduces accordingly the size of

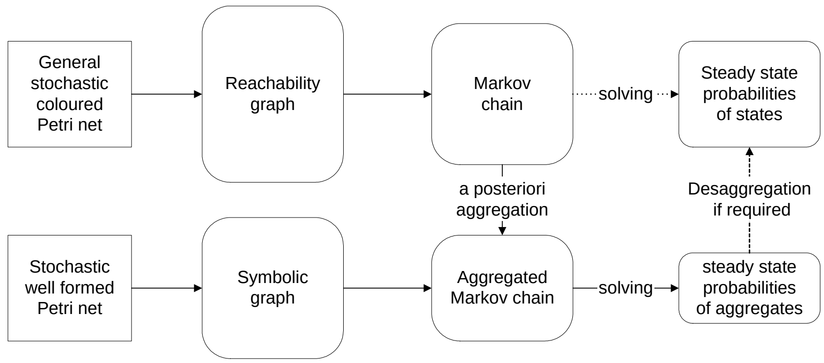

Figure 1: General stochastic colored net / well formed stochastic net

the built graphs. [FRA 93] proposes a bounding method for color domains made of static subclasses with same qualitative behavior but with different quantitative parameters. This method avoids partitioning the domain during the computation of the symbolic graph which reduces its size. In the same way, a stochastic simulation using symbolic markings is much faster. Indeed, the number of successors of a symbolic marking is significantly lower than the one of an ordinary marking [CHI 93b]. Most of these methods have been implemented in the GreatSPN tool [CHI 95].

2. Markovian aggregation

All aggregation technics substitute to a complex system, a easier system supposed to reflect the original system with regard to some criteria. In the performance evaluation area, this criterion is most often the fact that a synthetis of the stationary indices of the original system may be computed from the stationary indices of the reduced system. In the previous chapter, we saw how to approximate the stationary throughput of a stochastic marked graph (SMG) from the iterative evaluation of of reduced SMG. In this case, aggregation lies in the model which generates the stochastic process. Conversely, we can look for aggregation conditions of the stochastic process leading to exact results. Let us look at the figure 2. On the left, we have an excerpt of a discrete time Markov chain (DTMC). States are grouped in two subsets $E^k = \{e^k_1, e^k_2\}$ and $E^h = \{e^h_1, e^h_2, e^h_3\}$. We note that:

$$\Pr(X_{n+1} \in E^h \,|\, X_n = e^k_1) = \Pr(X_{n+1} \in E^h \,|\, X_n = e^k_2) = \frac{5}{6}$$

In other words, the probability for the process to reach E^h knowing it is in the subset of states E^k *does not depend of the specific visited state.*

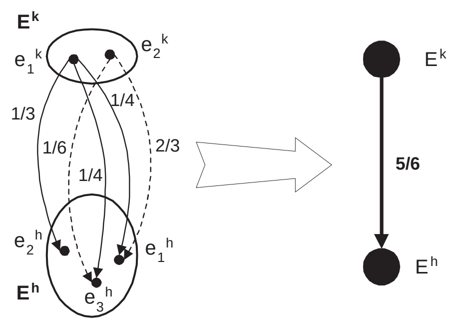

Figure 2: Markovian aggregation

Intuitively, if the partition satisfies this property for transitions between any pair of subsets, the aggregated process will be also a Markov chain. A formal expression of this fact is given in the following definition 1 and proposition 2.

Definition 1 *A discrete or continuous Markov chain* $\{Y_t\}_{t\in R^+}$ *may be aggregated with respect to the partition* $(E^k)_{k=1,\ldots,K}$*, iff the process* $\{Y_t^{(a)}\}_{t\in R^+}$ *with state space* $\widetilde{E} = \{E^k \mid k = 1,\ldots,K\}$*, defined by:*

$$\forall t \geq 0,\;\; Y_t^{(a)} = E^k \textit{ iff } Y_t \in E^k$$

is a Markov chain.

Proposition 2 (Strong aggregation condition [KEM 60]) *A chain may be aggregated whatever its initial distribution iff* $\forall h, k \in \{1,\ldots,K\}$, $\forall e, e' \in E^k$,

$$\sum_{e_h\in E^h} p_{e,e_h} = \sum_{e_h\in E^h} p_{e',e_h} \stackrel{def}{=} \widetilde{p}_{k,h} \quad \textit{(discrete time)} \qquad [1]$$

$$\sum_{e_h\in E^h} q_{e,e_h} = \sum_{e_h\in E^h} q_{e',e_h} \stackrel{def}{=} \widetilde{q}_{k,h} \quad \textit{(continuous time)}$$

The transition probabilities matrix (resp. the generator of the aggregated chain) is then $\widetilde{\mathbf{P}} = [\widetilde{p}_{k,h}]$ *for a discrete time Markov chain (resp.* $\widetilde{\mathbf{Q}} = [\widetilde{q}_{k,h}]$ *for a continuous time Markov chain).*

The definition itself of the aggregated process allows us to give the relationship between the stationary distributions.

Proposition 3 *Let $\{Y_t\}_{t\in R^+}$ be a chain satisfying the aggregation condition and $\{Y_t^{(a)}\}_{t\in R^+}$ its aggregated chain. If $\{Y_t\}$ is ergodic and has a stationary distribution $\boldsymbol{\pi}$, then $\{Y_t^{(a)}\}$ is ergodic and has a stationary distribution $\boldsymbol{\pi}^{(a)}$ satisfying:*

$$\boldsymbol{\pi}^{(a)}[E^k] = \sum_{e\in E^k} \boldsymbol{\pi}[e]$$

In the general case, we have $K \ll \sum_k |E^k|$ so that the computation complexity of the steady-state probabilities is significantly reduced for $\widetilde{\mathbf{P}}$ (resp. $\widetilde{\mathbf{Q}}$) with respect to $\mathbf{P}$ (resp. $\mathbf{Q}$).

It is unusual that the aggregated stationary distribution allows us to recognize the stationary distribution of the initial chain. However, this is the case if the process satisfies a condition about its *past* similar to the aggregation condition. We present the result in the framework of discrete time Markov chains (DTMC), but it also applies for continuous time Markov chains (CTMC).

Proposition 4 (Equiprobability of ordinary markings) *Let us given a discrete ergodic Markov chain satisfying the aggregation condition and let $\boldsymbol{\pi}^{(a)}$ be the stationary distribution of the aggregated chain. If $\forall h,k \in \{1,\ldots,K\}$, $\forall e,e' \in E^h$:*

$$\sum_{e_k\in E^k} p_{e_k,e} = \sum_{e_k\in E^k} p_{e_k,e'} \stackrel{def}{=} \widetilde{p}^{(\text{in})}_{k,h} \qquad [2]$$

then, the chain admits a stationary distribution $\boldsymbol{\pi}[e] = \frac{\boldsymbol{\pi}^{(a)}[E^h]}{|E^h|}$ where $|E^h|$ means the cardinality of E^h.

Proof
Let us recall that $\widetilde{p}_{k,h} = \sum_{e_h\in E^h} p_{e,e_h}$ for all $e \in E^k$.

Let us first prove that $|E^h|\widetilde{p}^{(\text{in})}_{k,h} = |E^k|\widetilde{p}_{k,h}$. The total flow from E^k to E^h is $F(k,h) = \sum_{e\in E^k}\sum_{e'\in E^h} p_{e,e'}$. Then, we have: $F(k,h) = \sum_{e\in E^k}\widetilde{p}_{k,h} = |E^k|\widetilde{p}_{k,h}$ and $F(k,h) = \sum_{e'\in E^h}\widetilde{p}^{(\text{in})}_{k,h} = |E^h|\widetilde{p}^{(\text{in})}_{k,h}$.

We have, for every $e \in E^h$,

$$\sum_{k=1}^{K}\sum_{e'\in E^k}\boldsymbol{\pi}[e']p_{e',e} = \sum_{k=1}^{K}\frac{\boldsymbol{\pi}^{(a)}[E^k]}{|E^k|}\sum_{e'\in E^k}p_{e',e} = \sum_{k=1}^{K}\frac{\boldsymbol{\pi}^{(a)}[E^k]}{|E^k|}\widetilde{p}^{(\text{in})}_{k,h} =$$

$$\frac{1}{|E^h|}\sum_{k=1}^{K}\boldsymbol{\pi}^{(a)}[E^k]\widetilde{p}_{k,h} = \frac{\boldsymbol{\pi}^{(a)}[E^h]}{|E^h|} = \boldsymbol{\pi}[e]$$

$\diamondsuit$

3. Presentation of stochastic well formed Petri nets

We refer the reader to chapter 7 for the definition of well formed (colored) Petri nets and we concentrate on introducing a stochastic semantics in the model.

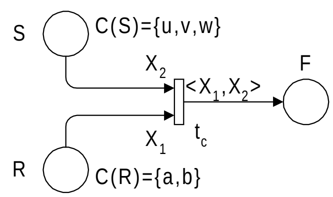

Figure 3: Transition and firing rates

Our approach is illustrated by the example in figure 3 showing a transition t of a well formed Petri net. This transition models a synchronization between requests (place R) of different kinds (a et b) and servers (place S) belonging to three categories (u, v et w).

3.1. *The stochastic process of a well formed Petri net*

Introducing a stochastic semantics for well formed Petri net aims to:

- be consistent with the stochastic semantics for Petri nets given in chapter 9;
- allow the user to specify it at the well formed net level;
- keep symmetries such that the symbolic reachability graph could be used for quantitative evaluation.

The easiest way is to define the stochastic semantics as the one of the unfolded (stochastic) Petri net. This is sufficient to ensure a coherent definition of a stochastic extension of well formed Petri nets. Among the various stochastic models of Petri nets, we choose the GSPN model which represents a right tradeoff between expressiveness and analysis potential.

Let us examine the three policies to be defined. Obviously, the choice policy will be the shortest sampled delay one if no immediate transition is enabled and a probabilistic selection conditioned on the weights of the enabled immediate transitions otherwise. A transition could, a priori, have immediate and exponential firing *instances*. However, the interpretation of the transition in the modeled system would be difficult and it is easy to replace it two transitions: an immediate one and an exponential one. So, we are led to define the kind of a transition rather than the kind of a firing instance of a transition.

The choice about the memory policy does not care here since we only use exponential and immediate distributions.

We remind the reader that GSPN allows us to specify dependencies of stochastic parameters with respect to the current marking. It is then easy to simulate an *infinite-server* policy with a *single-server* policy, knowing that the minimum of k exponential random variables with rate λ is an exponential random variable with rate $k.\lambda$. So, the problem is reduced to study what kinds of functional dependencies we would like to introduce in stochastic well formed Petri nets.

Let $\mathbf{w}[t](r,s,m)$ be the rate of the firing instance of t for the color couple (r,s) in a marking m. For an immediate transition, this expression is its weight among the probabilistic choices. Let us assume that we have the following values:

$$\mathbf{w}[t](a,u,m) = (m[S](a)+m[S](b)).\lambda \quad \mathbf{w}[t](b,u,m) = (m[S](a)+m[S](b)).\lambda$$
$$\mathbf{w}[t](a,v,m) = (m[S](a)+m[S](b)).\lambda \quad \mathbf{w}[t](b,v,m) = (m[S](a)+m[S](b)).\lambda$$
$$\mathbf{w}[t](a,w,m) = \lambda \ , \ \mathbf{w}[t](b,w,m) = \lambda$$

We first notice that each service (u,v,w) does not depend on the number of servers for a given color $(m[S](s))$; Only one instance of server is required to be served and two u servers (for instance) do not speed up the service. In contrast, servers u et v are sensitive to the number of requests since their rates are proportional to this number. (*infinite-server* policy) whereas w is a constant(*single-server* policy). At last, the request type does not impact how it is processed by the servers (from the quantitative point of view).

In brief, a and b have qualitative and quantitative equivalent behaviors. u,v,w have a qualitative equivalent behavior but w has a specific quantitative behavior. This means that $C(R)$ may comprise only on e static subclass while $C(S)$ must comprise two static subclasses $\{u,v\}$ et $\{w\}$. If the subclasses are defined before the stochastic semantics, then *functional dependencies have only to depend on static subclasses*. We will formalize this point in the next paragraph.

A last note about the example: as specified, the choice of the server for processing a request depends on the processing time known only at the end of the processing! A better modelling would be to introduce choice (immediate) transitions before the service transition.

3.2. *Definition of stochastic well formed Petri nets*

To formalize the restriction on functional dependencies, we introduce the notion of static subclass domain corresponding to a color domain and the image of a color in the associated domain. This will allow us to define firing rates only based on *projections* of the firing instance and on the current marking on the static subclasses.

Definition 5 *Let* $C(r) = \prod_{i=1}^{n} C_i^{e_i}$ *by the color domain of a node* r. *Let* $\widetilde{C}_i = \{C_{i,q} \mid \ 1 \leq q \leq s_i\}$ *be the set of static subclasses of* C_i. *The static subclass domain of* r *is*

$$\widetilde{C}(r) = \prod_{i=1}^{n} \prod_{j=1}^{e_i} \widetilde{C}_i = \prod_{i=1}^{n} \widetilde{C}_i^{e_i}$$

For $c \in C(r)$, $\widetilde{c} \in \widetilde{C}(r)$ *is the tuple of static subclasses to which each* c_i^j *belongs:* $\widetilde{c} = (C_{i,q_{i,j}})_n^{e_i}$ *with,* $\forall i, j,\ c_i^j \in C_{i,q_{i,j}}$

Hence, $\widetilde{C}(r)$ is the set of all possible static subclass tuples of a node. The next definition extends this transformation to markings.

Definition 6 (Static partition of a marking) *The static partition of a marking* m *is* $\widetilde{m} \in \prod_{p \in P} \mathrm{Bag}(\widetilde{C}(p))$ *with:*

$$\forall p \in P,\ \forall \widetilde{c} \in \widetilde{C}(p),\ \widetilde{m}[p](\widetilde{c}) = \sum_{c',\, \widetilde{c'} = \widetilde{c}} m[p](c')$$

$\widetilde{m}[p](\widetilde{c})$ *gives the number of tokens of* $m[p]$ *components of which are in the same static subclasses as* c.

We are now in position to give the formal definition of a stochastic well formed Petri net.

Definition 7 (Stochastic Well formed Petri net (SWN)) *A stochastic well formed Petri net (SWN) (SWN) is a pair*

$(\mathcal{S}, \mathbf{w})$ *where* $\mathcal{S} = (P, T, \mathbf{Pré}, \mathbf{Post}, \mathbf{Inh}, \mathbf{pri}, Cl, C, \Phi)$ *is a well formed Petri net and* $\mathbf{w}$ *a vector of functions defined on* T *such that:*

$$\mathbf{w}[t] :\ \widetilde{C}(t) \times \left(\prod_{p \in P} \mathrm{Bag}(\widetilde{C}(p)) \right) \longrightarrow \mathbb{R}^+$$

If $\mathbf{pri}[t] > 0$*, t is immediate and* $\mathbf{w}[t][\widetilde{c}, \widetilde{m}]$ *represents the weight of t. The firing probability of $t(c)$ in m is:*

$$\frac{\mathbf{w}[t][\widetilde{c}, \widetilde{m}]}{\sum_{(t',c')} \mathbf{w}[t'][\widetilde{c'}, \widetilde{m}]} \quad \textit{avec} \quad \mathbf{pri}[t'] = \mathbf{pri}[t] \ \ \textit{et} \ \ m[t'(c')\rangle$$

If $\mathbf{pri}[t] = 0$*, t is timed and* $\mathbf{w}[t][\widetilde{c}, \widetilde{m}]$ *represents the mean firing rate of any instance of $t(c)$ enabled in m.*

3.3. *Modelling a multiprocessor system*

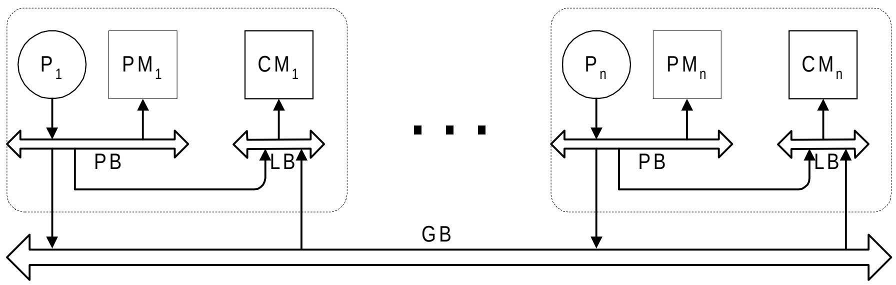

Figure 4: A multiprocessor system with private and common memories

In this section, we present a detailed example of a multiprocessor system modeled with stochastic well formed Petri net. Interests of this example are threefold:

- it provides an overview of the modelling process with stochastic well formed Petri nets;
- it allows for comparing sizes of the symbolic graph and the reachability graphs;
- it was used in a study to find a stochastic Petri net the reachability graph of which would be an aggregated version of the initial graph. This analysis is rather difficult and would probably not be generally applicable to more complex modelling; with the symbolic graph, the modeler gets the same reduction ratio effortlessly!

The multiprocessor architecture analysed in this example [Ajm 84, DUT 89, DUT 91] is shown in figure 4. Each processor p_i owns a local memory made up of two parts, a private memory (PM_i) and a common memory (CM_i). The private memory may only be acceded by its processor through its private bus (PB_i). The common memory may be reached by all processors of the system. The processor p_i reaches the common memory (CM_i) through its private bus and its local bus (LB_i). Other processors access this memory through the global bus (GB) and through the local bus.

Access conflicts arise when using either GB or local buses (and common memories). A processor is suspended when it tries to get an already used resource. We assume that external accesses to common memories have priority against local accesses and cause their preemption. We can describe the global behavior of the system as follows: processors alternate between duty periods involving only private memory accesses (so called *CPU burst*), and duty periods with common memories accesses. To simplify the exposition, we assume that the system is made up of n identical processors.

To build the SWN model, it is interesting to enumerate the possible states of a processor:

- *ACTIVE* the processor works with its private memory;
- *LOCAL* the processor works with its common memory;
- *DISTANT* the processor works with another common memory;
- *WAITING* the processor waits for the global bus;
- *BLOCKED* the processor waits for accessing its common memory.

The behavior of this system is described by the SWN (figure 5). There is only on color class, the class P of processors (sets p_i, PM_i, CM_i). X et Y denote processor variables.

Places represent the states of the processors. *Run* holds one token per processor in state *ACTIVE* (hence the S term, which corresponds to one token for each color). In the same way, $ExtMemAcc$ and *Queue* represent respectively states *DISTANT* and *WAIT*. The place *OwnMemAcc* represents either state *LOCAL*, or state *BLOCKED* depending on whether there is or not a token with same color in the place *Mem*. A probabilistic choice between private, local or external accesses is modeled by the immediate transitions in conflict *ReqPrivMem*, *BeginOwnAcc* et *ReqExtAcc*.

In this last case, the variable Y represents the choice of a common external memory ($\neq X$). At last, the place *ExtBus* represents the availability of the global bus.

Even if this net is correct and easily understood, it is not for sure the most compact one. For instance, the three conflicting immediate transitions

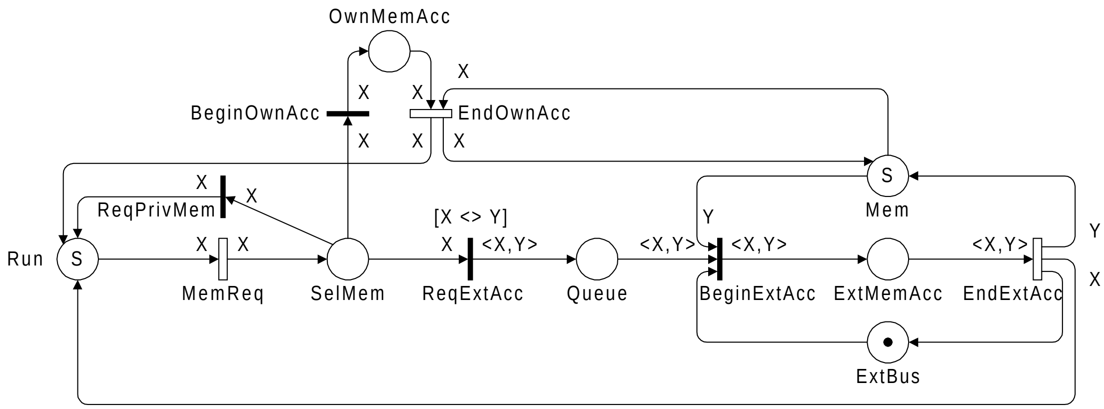

Figure 5: Initial stochastic well formed Petri net of the multiprocessor system

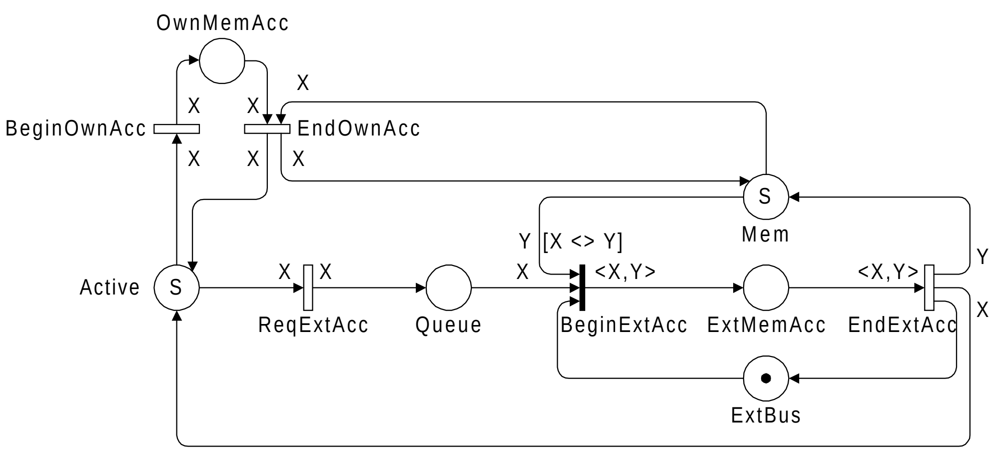

Figure 6: Simplified stochastic well formed Petri net of the multiprocessor system

may by *pre-agglomerated* with the exponential transition *MemReq* applying a reduction rule which preserves the qualitative and quantitative behavior of the net [HAD 89]. Moreover, the exponential transition resulting from the fusion of *MemReq* and *ReqPrivMem* may be discarded since its firing does not modify the state of the system. Finally, the external memory choice may be delayed until firing of the transition *BeginExtAcc*. This transformation is justified because the marking of the place *Mem* holds all the processors when the place *ExtBus* is marked. This domain *reduction* of the place *Queue* leads to a significant reduction of the state space.

Applying all the simplifications, we finally get the net of figure 6 with six places:

- *Active* holds tokens of processors in state *ACTIVE*;
- *Queue* holds tokens of processors in state *WAITING*;
- *ExtMemAcc* holds pairs (processor, common memory) representing current external accesses. With only one global bus, there is at most one token in this place.
- *OwnMemAcc* holds tokens of processors in state *LOCAL* or in state *BLOCKED*, the last ones being distinguished by no corresponding token in the place *Mem*.
- *Mem* holds common memories not used by an external processor.
- *ExtBus* with neutral domain, holds a token when *GB* is available.

Let us now have a look at the stochastic parameters. The transition *BeginExtAcc* is immediate since bus arbitration and bus release durations are negligible. Moreover, we assume that *CPU burst* an common memory accesses time are independent random variables with exponential distributions. The external memory choice is equi-distributed (probability $\frac{1}{n-1}$) as the resolution of conflicting global bus accesses. This leads to weight 1 for all instances of the immediate transition *BeginExtAcc*. Quantitative parameters of our model are: n, the number of processors, $\frac{1}{\lambda}$, the mean time of a *CPU burst* period (λ is the rate of the transition *EndOwnAcc*) and $\frac{1}{\mu}$ the mean common memory access time (μ is the rate of the transition *EndExtAcc*). An important auxiliary parameter $\rho = \frac{\lambda}{\mu}$ represents the load ratio of the system.

4. From the symbolic graph to Markovian aggregation

Since we deal with a semi-Markovian process and we want to apply the aggregation technic (valid for Markov chains) we proceed in two steps.

- We focus on the embedded chain at changing state times, as defined in chapter 9. We prove that this chain may be aggregated and that its aggregated version is isomorphic to the symbolic graph.
- Then we prove that sojourn times in a symbolic marking do not depend on the ordinary marking which concludes the proof.

For the computation of the parameters of the aggregated chain, we use formulae on cardinalities of the symbolic arcs and of the symbolic markings. Same formulae provide sojourn times.

4.1. *Verification of the aggregation condition*

We denote by **P** the state transition matrix of the embedded Markov chain of the semi-Markovian process associated to the SWN.

Theorem 8 (Aggregated Markov chain of a SWN) *Let C be the embed-ded*
Markov chain of a stochastic well formed Petri net. The symbolic markings partition of the state space satisfies the the aggregation condition.

Proof
By definition, $p_{m,m'} = \frac{\sum_{t(c),m[t(c)\rangle m'} \mathbf{w}[t](c,m)}{\sum_{t(c),m[t(c)\rangle} \mathbf{w}[t](c,m)}$ is the jump probability from m to m'.

Let $\widehat{m}$ and $\widehat{m}'$ be two symbolic markings. We have to show that:

$$\forall m_1, m_2 \in \widehat{m}, \quad \sum_{m' \in \widehat{m}'} p_{m_1,m'} = \sum_{m' \in \widehat{m}'} p_{m_2,m'}$$

Let us first show that, for an admissible permutation s and for two markings $m_1 \in \widehat{m}$, $m' \in \widehat{m}'$, $p_{m_1,m'} = p_{s.m_1,s.m'}$ ($s.m$ stands for the image of m by s).

We know that

- in a well formed Petri net, $m_1[t(c) > m'$ iff $s.m_1[t(s.c) > s.m'$;
- $\mathbf{w}[t](c, m_1)$ depends only on the static subclasses composing c and on the static subclasses to which dynamic subclasses of m_1 belong. Since static subclasses are invariant through admissible permutation,

$$\mathbf{w}[t](c, m_1) = \mathbf{w}[t](s.c, s.m_1)$$

Equities below are then valid since summation indexes are interrelated with s and because interrelated summation terms are equal.

$$p_{m_1,m'} = \frac{\sum_{t(c),m_1[t(c)\rangle m'} \mathbf{w}[t](c,m_1)}{\sum_{t(c),m_1[t(c)\rangle} \mathbf{w}[t](c,m_1)} =$$

$$\frac{\sum_{t(s.c),s.m_1[t(s.c)\rangle s.m'} \mathbf{w}[t](s.c,s.m_1)}{\sum_{t(s.c),s.m_1[t(s.c)\rangle} \mathbf{w}[t](s.c,s.m_1)} = p_{s.m_1,s.m'}$$

Let $m_1, m_2 \in \widehat{m}$. By definition, there is an admissible permutation s such that $m_2 = s.m_1$. Applying the previous result, we get:

$$\sum_{m' \in \widehat{m}'} p_{m_1,m'} = \sum_{s.m' \in \widehat{m}'} p_{s.m_1,s.m'} = \sum_{s.m' \in \widehat{m}'} p_{m_2,s.m'} = \sum_{m' \in \widehat{m}'} p_{m_2,m'}$$

The first equality derives from the previous result and we get the last equality permuting indexes of the sum with s^{-1}. Hence, we have got the aggregation condition. $\diamondsuit$

We supplement this result with the equi-probability of the ordinary markings of a symbolic marking in the stationary distribution. This allows us to compute, if needed, the steady-state probability of an ordinary marking from the probabilities of the aggregated chain.

Theorem 9 (Equi-probability inside a symbolic marking)
Let C be the embedded Markov chain of a stochastic well formed Petri net. If C is ergodic, then all markings of a tangible symbolic marking have same steady-state probability.

Proof
Let $m_1, m_2 \in \widehat{m}$. By definition, there is an admissible permutation such that $m_2 = s.m_1$. Hence, we have:

$$\sum_{m' \in \widehat{m}'} p_{m',m_1} = \sum_{s.m' \in \widehat{m}'} p_{s.m',s.m_1} = \sum_{s.m' \in \widehat{m}'} p_{s.m',m_2} = \sum_{m' \in \widehat{m}'} p_{m',m_2}$$

for the first equality comes from the intermediate result of the previous proof and we get the last equality permuting indexes of the sum with s^{-1}. We have got the sufficient condition of proposition 4, which concludes the proof. $\diamondsuit$

We have now to go back to the semi-Markovian process with the help of sojourn times.

Theorem 10 (Equality of sojourn times) *All markings of a tangible symbolic marking of a stochastic well formed Petri nethave same sojourn time.*

Proof
Let m_1 be an ordinary marking,

$$\frac{1}{\mathrm{sjtime}(m_1)} = \sum_{t(c),m_1[t(c)\rangle} \mathbf{w}[t](c, m_1) =$$

$$\sum_{t(s.c),s.m_1[t(s.c)\rangle} \mathbf{w}[t](s.c, s.m_1) = \frac{1}{\mathrm{sjtime}(s.m_1)}$$

As for the first theorem of this section, above equalities are valid since summation indexes are interrelated by s and since interrelated terms of the summation are equal. ♢

We denote by $\mathrm{sjtime}(\widehat{m}) = \mathrm{sjtime}(m)$ the sojourn time of any marking m of the tangible marking $\widehat{m}$.

4.2. *Computation of the parameters of the aggregated chain*

The SWN model is interesting because it also allows us to *compute the parameters* of the aggregated chain defined by the symbolic reachability graph, *from the definition of the net* and from this graph. Since we use the embedded chain method, it is sufficient to show how to compute the coefficient $\widehat{p}_{\widehat{m},\widehat{m}'}$ of the transition probabilities matrix of this chain and $\mathrm{sjtime}(\widehat{m})$, the sojourn time in an ordinary marking of the tangible marking $\widehat{m}$.

In the following, we denote by $\widehat{m}[t(\lambda, \mu)\rangle$ a symbolic firing and by $m[t(c)\rangle$ any of the ordinary firings corresponding to the symbolic firing.

By construction, all ordinary firings denoted by a symbolic arc drop on the same static subclasses in the sense of definition 5. Likewise, all ordinary markings of a symbolic marking drop on the same static partition in the sense of definition 6.

Hence, the stochastic parameter of the ordinary firing $\mathbf{w}[t][\widetilde{c}, \widetilde{m}]$ does not depend of the choice of the ordinary firing and derives directly from the symbolic marking and from the symbolic firing. We denote it by $\widehat{\mathbf{w}[t]}(\lambda, \mu, \widehat{m})$.

Expressions of the coefficients of the matrix of the embedded aggregated chain and the sojourn time are then given by the formulae:

$$\widehat{p}_{\widehat{m},\widehat{m}'} = \frac{\sum_{\langle t,\lambda,\mu\rangle, \widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle} \widehat{m}'} \widehat{\mathbf{w}[t]}(\lambda,\mu,\widehat{m}) |\widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}|}{\sum_{\langle t,\lambda,\mu\rangle, \widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}} \widehat{\mathbf{w}[t]}(\lambda,\mu,\widehat{m}) |\widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}|}$$

$$\mathrm{sjtime}(\widehat{m}) = \frac{1}{\sum_{\langle t,\lambda,\mu\rangle, \widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}} \widehat{\mathbf{w}[t]}(\lambda,\mu,\widehat{m}) |\widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}|}$$

where the second formula applies only to tangible symbolic markings and where $|\widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}|$ is the number of colored firings from a fixed marking of $\widehat{m}$, represented by the symbolic instanciation $\langle t,\lambda,\mu\rangle$. But, we show [DUT 91, CHI 93a]:

$$|\widehat{m} \xrightarrow{\langle t,\lambda,\mu\rangle}| = \prod_{i=1}^{h} \prod_{j=1}^{m_i} \frac{\mathrm{card}(Z_i^j)!}{(\mathrm{card}(Z_i^j) - \mu_i^j)!}$$

where h is the number of unordered classes, m_i is the number of dynamic subclasses of C_i in the representation and μ_i^j is the number of instanciations in Z_i^j.

Finally, if we want to get the steady-state probability of an ordinary marking m, we simply have to divide the probability of its symbolic marking with the cardinality of the latter which is:

$$\frac{1}{|S(\widehat{m})|} \left(\prod_{i=1}^{h} \prod_{q=1}^{s_i} \frac{|C_{i,q}|!}{\prod_{d(Z_i^j)=q} \mathrm{card}(Z_i^j)!} \right) \prod_{i=h+1}^{n} v(i)$$

with s_i the number of static subclasses of C_i, $v(i) = |C_i|$ if $m_i > 1$ and $s_i = 1$, and 1 otherwise and $S(\widehat{m})$ the admissible permutations of the *symbolic* marking $\widehat{m}$ that is the number of permutations defined on dynamic subclasses leaving invariant the symbolic marking (see [DUT 91] for more details).

4.3. *Performance indices of the multiprocessor system*

We apply the technic just described to our multiprocessor system. We choose two significant indices:

- $\overline{a}$ the mean ratio of actove processors with respect to the total number of processors, given by the formula

$$\overline{a} = \frac{1}{n} \sum_{\widehat{m}} \boldsymbol{\pi}^{(a)}[\widehat{m}]. \sum_{Z_1^j \in \widehat{m}(Active)} \mathrm{card}(Z_1^j)$$

n	ρ	|TSRS|	|TRS|	$\overline{a}$	$\overline{u}$
2	0.2	6	10	0.6752411	0.27009645
	0.5			0.4285714	0.42857145
	1.0			0.2608695	0.52173917
5	0.2	36	1652	0.6227463	0.62274684
	0.5			0.3444203	0.86105182
	1.0			0.1882871	0.94143486
10	0.2	146	1772494	0.475696106	0.95139024
	0.5			0.199762593	0.99881109
	1.0			0.099993149	0.99992986

Table 1: Performance results of the multiprocessor system

where $\widehat{m}$ is tangible and belongs to the symbolic graph.

- $\overline{u}$ the mean utilization of the global bus, given by the formula

$$\overline{u} = \sum_{\widehat{m}[ExtBus]=0} \boldsymbol{\pi}^{(a)}[\widehat{m}]$$

where $\widehat{m}$ is tangible and belongs to the symbolic graph.

Results, computed with the GreatSPN software [CHI 95], are presented in table 4.3. TSRS is the Tangible Symbolic Reachability Set and TRS is the Tangible Reachability Set of the net. We note that the increase of the symbolic graph size is almost linear with respect to the number of processors whereas the reachability graph reaches nearly 2 millions of states for 10 processors. Numerical results confirm that the global bus is very quickly the bottleneck of the system. Hence, it would be interesting to supplement these results by varying the number of global buses.

5. Conclusion

Markovian aggregation methods reduce the size of the Markov chain to be solved to get performance indices of discrete event systems. stochastic well formed Petri nets take advantage of Markovian aggregation for systems modeled with stochastic Petri nets and with behavioral symmetries. Taking these symmetries into account in the definition itself of colored nets, we have developed efficient resolution methods, that is to say without computing the non aggregated Markov chain. With the help of the symbolic reachability graph built form the description of the well formed net, we are able to define and analyze an aggregated Markov chainof the Markov chain of the colored stochastic Petri net. Moreover, the states of each aggregate, a set of colored markings, are equiprobable. The reduction ratio of the size of the studied Markov chain id obviously related to the symmetry level in the system, and may be very high as shown by a lot of examples. In the next chapter, we present the tensorial approach the goal of which is also to reduce the resolution complexity of the Markov chain, and we show how to combine these two methods.

References

[Ajm 84] M. Ajmone Marsan, G. Balbo and G. Conte. A class of generalized of stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on computer systems*, 2(2):93–122, May 1984.

[CHI 88] G. Chiola, G. Bruno and T. Demaria. Introducing a color formalism into generalized stochastic Petri nets. In *Proc.* 9^{th} *Europ. Workshop on Application and Theory of Petri Nets*, Venezia, Italy, June 1988.

[CHI 93a] G. Chiola, C. Dutheillet, G. Franceschinis and S. Haddad. Stochastic well-formed colored nets and symmetric modeling applications. *IEEE Transactions on Computers*, 42(11):1343–1360, November 1993.

[CHI 93b] G. Chiola and R. Gaeta. Efficient simulation of parallel architectures exploiting symmetric well-formed Petri net models. In C.E. Knadler and H. Vakilzadian, eds, *SCS Western Simulation Multiconference '93*, volume 25 of *Simulation Series*, pages 285–290, San Diego, California, January 1993. Society for Computer Simulation.

[CHI 95] G. Chiola, G. Franceschinis, R. Gaeta and M. Ribaudo. Great-SPN 1.7: GRaphical Editor and Analyser for Timed and Stochastic Petri nets. *Performance Evaluation*, 24(1,2):47–68, 1995.

[DUT 89] C. Dutheillet and S. Haddad. Aggregation of states in colored stochastic Petri nets: application to a multiprocessor architecture. In *Proc. of the Third International Workshop on Petri Nets and Performance Models*, pages 40–49, Kyoto, Japan, December 11–13 1989. IEEE Computer Society Press.

[DUT 91] C. Dutheillet. *Symétries dans les réseaux colorés. Définition, analyse et application à l'évaluation de performance.* Thèse, Université Paris VI, Paris, France, 28 janvier 1991.

[FRA 93] G. Franceschinis and R. R. Muntz. Computing bounds for the performance indices of quasi-lumpable stochastic well-formed nets. In *Proc. of the 5th International Workshop on Petri Nets and Performance Models*, pages 216–225, Toulouse, France, October 19–22 1993. IEEE Computer Society Press.

[HAD 89] S. Haddad. A reduction theory for coloured nets. In G. Rozenberg, ed, *Advances in Petri Nets 1989*, volume 424 of *LNCS*, pages 209–235. Springer–Verlag, 1989. reprinted in [JEN 91] pp. 399–425.

[HUB 84] P. Huber, A.M. Jensen, L.O. Jensen and K. Jensen. Towards reachability trees for high-level Petri nets. In G. Rozenberg, ed, *Advances in Petri Nets 1984*, volume 188 of *LNCS*, pages 215–233. Springer Verlag, 1984.

[JEN 91] K. Jensen and G. Rozenberg, eds. *High-Level Petri Nets. Theory and Application.* Springer–Verlag, 1991.

[KEM 60] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* V. Nostrand, Princeton, NJ, 1960.

[LIN 88] C. Lin and D. C. Marinescu. Stochastic high-level Petri nets and applications. *IEEE Transactions on Computers*, 37(7):815–825, July 1988. reprinted in [JEN 91] pp. 459–469.

[ZÉN 85] A. ZÉNIÉ. Colored stochastic Petri nets. In *Proc. of the International Workshop on Timed Petri Nets*, pages 262–271, Torino, Italy, July 1-3 1985. IEEE Computer Society Press.

Index

Haddad, S., 1
Moreaux, P., 1

aggregation, 3

stochastic well formed Petri net, 8