Well-defined stochastic Petri nets

Gianfranco Ciardo[*]

Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795, USA
ciardo@cs.wm.edu

Robert Zijal[†]

Institut für Technische Informatik
Technische Universität Berlin
Franklinstr. 28/29, 10587 Berlin, Germany
bob@cs.tu-berlin.de

Abstract

Formalisms based on stochastic Petri Nets (SPNs) can employ structural analysis to ensure that the underlying stochastic process is fully determined. The focus is on the detection of conflicts and confusions at the net level, but this might require to overspecify a given SPN model. The problem becomes even more critical when reward processes of interest derived from the basic underlying process are considered. Typical examples are state-dependent impulse reward measures. We propose a definition of well-defined SPNs, which takes into account whether the basic underlying stochastic process or the derived reward processes are determined. A state-space-based algorithm to determine whether a given SPN is well-defined is provided.

Keywords: Stochastic Petri net, reward process, conflict, confusion.

1 Introduction

The Petri net (PN) formalism is an excellent tool for the description of the logical behavior of discrete-state systems exhibiting concurrence, synchronization, and conflict.

In the performance and reliability arena, the need to explicitly model the timing and probabilistic behavior of a system further led to several definitions of related, but not exactly equivalent, stochastic Petri net (SPN) formalisms [14, 4, 5, 11, 12].

Most of the work has focussed on SPNs having an underlying continuous-time Markov chain (CTMC). This is achieved by associating an exponentially or PH-type [13] distributed *firing time* to each transition.

The continuous support interval $[0, +\infty)$ of these distributions implies that the modeler does not have to worry about contemporary events in the model: they have null probability. However, this property implies that it is actually difficult to model systems where causal connections between untimed events does indeed arise.

This problem was solved in the GSPNs and related formalisms [4, 2, 3, 6, 11] by introducing the *immediate transitions*, which fire immediately upon becoming enabled. This allows for contemporary firings, and requires to specify additional information in the model, to "break the ties". More recently, a class of SPNs having an underlying discrete-time Markov chain (DTMC) has been introduced [10], which extends earlier definitions [15]. For these SPN models, contemporary firings become the rule more than the exception, since all transitions fire only at integral times. Also in this case, we must ensure that the underlying stochastic process of interest is fully determined. We then say that such a SPN is *well-defined*.

The initial approach to break the ties in GSPNs was to define (global) *weights* for the immediate transitions, and select the one to fire among those enabled with a probability proportional to its weight. This results in a model whose stochastic behavior is completely defined. The modeler, however, is forced to assign weights to each immediate transition, hence he must implicitly know which sets of immediate transitions might become enabled in some reachable marking, a formidable task in a large, complex model. The issue is particularly important if there is a possibility of *confusion* among immediate transitions.

In [6], a different approach is chosen. Weights are only meaningful locally, within an *extended conflict set* (ECS). Immediate transitions in different ECSs can be fired independently, or concurrently, because any sequentialization of their firing has the same effect. A structural analysis of the GSPN is performed to define the ECSs, ensuring that no confusion or conflict exists among transitions in different ECSs. This class of GSPNs is said to be "defined at the net level".

Both approaches to ensure that the SPN clearly defines a stochastic process have advantages and disadvantages. Global weights put the burden on the modeler, while local weights are easier to specify and less prone to change the model behavior in unexpected and subtle ways which can result in undetected errors. However, the structural tests of [6] are based on necessary, not sufficient, conditions, hence they can generate false alarms. When this happens, the modeler is forced to either specify weights which are not going to be used in the analysis, or ignore the structural analysis warnings, taking responsibility for its consequences. In this second case, the burden on the modeler is analogous to that required to ensure that global weights result in the intended behavior.

We propose a mixed approach, where weights are

---

[*] Gianfranco Ciardo was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480

[†] Robert Zijal was supported by a doctoral fellowship from the German National Research Council under grant Ho 1257/1-2.

meaningful locally, but the determination of whether the SPN is well-defined is performed during the generation of the underlying stochastic process, not before. Our approach is more general and does not cause "false alarms", but it is more computationally expensive. Indeed, it can be used in conjunction with the approach in [6], since it applies to the subsequent phase of the analysis. The work most closely related to ours is that of Sanders [17] on well-specified Stochastic Activity Networks (SANs). Our contribution can be seen as an extension of that work, because it applies also to reward structures where impulse rewards are state-dependent. It should also be noted that Sanders recently generalized further his definition of well-specified SANs [16]. The approach presented here has several similarities with this later work. Our SPN definition allows marking-dependent arc cardinalities, which have no correspondence in SANs; on the other hand, the definition in [16] associates reward impulses to entire vanishing paths (steps, in his terminology), rather than just individual firings, hence it is potentially even more general in this respect.

Sections 2 and 3 define the PN and SPN formalisms we adopt. Section 4 describes stochastic conflicts and confusions, and how they can be solved, resulting in the definition of well-defined SPNs of Section 5. Finally, Section 6 offers an algorithm to ensure that a SPN is indeed well-defined.

2 The PN formalism

We recall the (extended) PN formalism used in [10]. See also [9] for more details on PNs with marking-dependent arc multiplicities. A PN is a tuple $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}\right)$ where:

- P is a finite set of *places*, which can contain tokens. A marking $\mu \in \mathbb{N}^{|P|}$ defines the number of tokens in each place $p \in P$, indicated by μ_p (when relevant, a marking should be considered a column vector).
- T is a finite set of *transitions.* $P \cap T = \emptyset$.
- $\forall p \in P, \forall t \in T, \forall \mu \in \mathbb{N}^{|P|}$, $D^-_{p,t}(\mu) \in \mathbb{N}$, $D^+_{p,t}(\mu) \in \mathbb{N}$, and $D^\circ_{p,t}(\mu) \in \mathbb{N}$ are the multiplicities of the *input arc* from p to t, the *output arc* from t to p, and the *inhibitor arc* from p to t, when the marking is μ, respectively.
- $\succ \; \subseteq T \times T$ is an acyclic (*pre-selection*) *priority relation.*
- $\forall t \in T, \forall \mu \in \mathbb{N}^{|P|}, g_t(\mu) \in \{0, 1\}$ is the *guard* for t in marking μ.
- $\mu^{[0]} \in \mathbb{N}^{|P|}$ is the *initial marking.*

Places and transitions are drawn as circles and rectangles, respectively. The number of tokens in a place is written inside the place itself (default is zero). Input and output arcs have an arrowhead on their destination, inhibitor arcs have a small circle. The multiplicity is written on the arc (default is 1); a missing arc indicates that the multiplicity is 0. The default value for guards is 1.

Let $\mathcal{E}(\mu)$ be the set of transitions *enabled* in marking μ. A transition $t \in T$ is enabled in marking μ if, and only if, its guard evaluates to 1, its input and inhibitor arc conditions are satisfied, and no other transition with pre-selection priority over t is enabled: $(g_t(\mu) = 1) \wedge \left(\forall p \in P, D^-_{p,t}(\mu) \leq \mu_p \wedge \left(D^\circ_{p,t}(\mu) > \mu_p \vee D^\circ_{p,t}(\mu) = 0\right)\right) \wedge \left(\forall u \in \mathcal{E}(\mu), u \not\succ t\right)$.

A transition $t \in \mathcal{E}(\mu)$ can fire, causing a change to marking $\mathcal{M}(t, \mu)$, obtained from μ by subtracting the *input bag* $D^-_{\bullet,t}(\mu)$ and adding the *output bag* $D^+_{\bullet,t}(\mu)$ to it: $\mathcal{M}(t, \mu) = \mu - D^-_{\bullet,t}(\mu) + D^+_{\bullet,t}(\mu) = \mu + D_{\bullet,t}(\mu)$, where $D = D^+ - D^-$ is the incidence matrix. $\mathcal{M}$ can be extended to its reflexive and transitive closure by considering the marking reached from μ after firing a sequence of transitions. The *reachability set* is then given by $\mathcal{S} = \{\mu : \exists \sigma \in T^* \wedge \mu = \mathcal{M}(\sigma, \mu^{[0]})\}$, where T^* indicates the set of transition sequences. The *reachability graph* is $(\mathcal{S}, \mathcal{A})$, where $\mathcal{A}$ contains an arc $\mu \xrightarrow{t} \mu'$ iff $\mu \in \mathcal{S}$, $t \in T$, and $\mu' = \mathcal{M}(t, \mu)$.

3 The SPN formalism

A SPN is obtained by associating random durations to the firing time of the PN transitions. If no restriction is placed on the distributions, the state s of a SPN has a discrete structural component, the marking μ, and a continuous timing component, the *remaining firing times* (RFTs) τ: $s = (\mu, \tau) \in \mathbb{N}^{|P|} \times (\mathbb{R}^0)^{|T|}$, where we define $\mathbb{R}^0 = [0, +\infty)$ and $\mathbb{R}^+ = (0, +\infty)$. For each transition t, τ_t is the time that must elapse while t is enabled, before it can fire. If we do not reset τ_t when t becomes disabled, we can model an "age memory" behavior [1]. Otherwise, if we resample it when t becomes enabled again, we can model an "enabling memory" behavior [1]. Any mixture of these behaviors, and other more complex ones are included in our definition.

Using the terminology of [1], we assume a "race policy": only the enabled transitions with the minimum RFT can fire in a given state. In addition to the distribution of the firing times, stochastic information might be required to fully define the behavior of the SPN, since multiple enabled transitions might have the same minimum RFT in a state. The weights are used to determine the probability that a given one fires first. Formally, a SPN is a tuple $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}, F, \tau^{[0]}, \mathrel{-\!\!\!\succ}, C, w\right)$ where:

- $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}\right)$ define a PN.
- $\forall \mu \in \mathcal{S}, \forall t \in \mathcal{E}(\mu), \forall u \in T, \forall \tau_u \in \mathbb{R}^0, F_{t,u}(\mu, \tau_u, \cdot)$ is the cumulative distribution function (CDF) for the new RFT of u when t fires in marking μ, and the RFT of u just before the firing of t was τ_u.
- $\forall t \in T, \tau^{[0]}_t \in \mathbb{R}^0$ is the initial RFT of t.
- $\mathrel{-\!\!\!\succ} \; \subset T \times T$ is an acyclic (post-selection) priority relation.
- C is a partition of T into weight classes. Let C_t be the weight class containing transition $t \in T$.

- $\forall \mu \in \mathcal{S}, \forall t \in \mathcal{E}(\mu), \forall S \subseteq C_t \cap \mathcal{E}(\mu), w_{t|S}(\mu) \in \mathbb{R}^+$ is the firing weight for t in marking μ when S is the set of *candidates* (see the following) to fire in the same weight class as t.

We now formalize the dynamic evolution of a SPN. At time θ_1, the SPN is in state $s(\theta_1) = (\mu(\theta_1), \tau(\theta_1))$. If $\mathcal{E}(\mu(\theta_1))$ is empty, the state is absorbing. Otherwise, let $\tau^* = \min_{t \in \mathcal{E}(\mu(\theta_1))}\{\tau_t(\theta_1)\}$ be the minimum RFT among the enabled transitions and $\theta_2 = \theta_1 + \tau^*$. Then, at any time $\theta' \in [\theta_1, \theta_2)$, the state is $s(\theta') = (\mu(\theta_1), \tau(\theta'))$, where

$$\tau_t(\theta') = \begin{cases} \tau_t(\theta_1) & \text{if } t \notin \mathcal{E}(\mu(\theta_1)) \\ \tau_t(\theta_1) - (\theta' - \theta_1) & \text{if } t \in \mathcal{E}(\mu(\theta_1)) \end{cases}$$

This means that the RFT of each transition is decremented by the amount of time elapsed while being enabled. At time θ_2, at least one RFT reaches zero, and the corresponding transitions attempt to fire. The notation for the "state at time θ" needs then to be refined. If one or more firings occur at time θ:

- $\lim_{\theta' \to \theta^+} s(\theta') = s(\theta^+)$ and $\lim_{\theta' \to \theta^-} s(\theta') = s(\theta^-)$ normally differ.

- We define the state at time θ as the new state reached after any firings occurring at time θ: $s(\theta) \overset{\text{def}}{=} s(\theta^+)$.

- Assuming n firings occur, we need to denote the $n-1$ intermediate states visited during these firings: $s(\theta, i) = (\mu(\theta, i), \tau(\theta, i)), i \in \{1, 2, \ldots, n-1\}$ is the i-th state entered at time θ. Also, let $s(\theta, 0) = (\mu(\theta, 0), \tau(\theta, 0)) \overset{\text{def}}{=} s(\theta^-)$ and $s(\theta, n) = (\mu(\theta, n), \tau(\theta, n)) \overset{\text{def}}{=} s(\theta)$.

Continuing our example, if only one transition $t \in \mathcal{E}(\mu(\theta_1))$ has the minimum RFT τ^*, it fires immediately at time θ_2, and the new state of the SPN is $s(\theta_2) = (\mu(\theta_2), \tau(\theta_2)) = \mathcal{M}(t, s(\theta_1))$, where $\mathcal{M}$ has been extended from markings to states: $\mu(\theta_2) = \mathcal{M}(t, \mu(\theta_1))$, and $\forall u \in T, \tau_u(\theta_2)$ is a random deviate from the distribution $F_{t,u}(\mu(\theta_2, 0), \tau_u(\theta_2, 0), \cdot)$, that is, from $F_{t,u}(\mu(\theta_1), \tau_u(\theta_1) - \tau^*, \cdot)$, if $u \in \mathcal{E}(\mu(\theta_1))$, and from $F_{t,u}(\mu(\theta_1), \tau_u(\theta_1), \cdot)$, otherwise.

If multiple transitions reach a zero RFT at the same time, we choose one to fire among them, say t, according to the post-selection priority and the weights. In the new marking, some transitions might still have a zero RFT, so they can fire immediately, depending on the effect of the firing of t on the marking and on the RFTs, and so on, until we reach a state where no enabled transition has a zero RFT. Formally, a transition $t \in T$ is said to be a *candidate* in state $s(\theta) = (\mu(\theta), \tau(\theta))$ iff $t \in \mathcal{E}(\mu(\theta)) \wedge \tau_t(\theta) = 0 \wedge (\forall u \in T, u \not\succ t \vee u$ is not a candidate in $s(\theta))$. That is, a transition t is a candidate in $s(\theta)$ if it is enabled, its RFT is zero, and no other candidate has post-selection priority over it.

Let $\mathcal{C}(s)$ be the set of candidates in state $s = (\mu, \tau)$. Then, transition $t \in \mathcal{C}(s)$ is chosen to fire among the transitions in its weight class C_t w.p.

$$\hat{w}_{t|\mathcal{C}(s) \cap C_t}(\mu) = \frac{w_{t|\mathcal{C}(s) \cap C_t}(\mu)}{\displaystyle\sum_{u \in \mathcal{C}(s) \cap C_t} w_{u|\mathcal{C}(s) \cap C_t}(\mu)}$$

Unlike [11], no firing probability is defined when t_1 and t_2 are candidates belonging to different classes.

We adopt the GSPN terminology and call a state s *tangible* if $\mathcal{C}(s) = \emptyset$, *vanishing* otherwise.

4 Stochastic conflict and confusion

Fig. 1 shows three SPNs and their underlying basic process. For all three, $s(0) = (1100, cc\bullet)$, where $\bullet$ indicates a "no-value" RFT for t_3, meaning that its firing time must be resampled as soon as it becomes enabled. Consider the first SPN. At time c^-, the state is $s(c, 0) = (1100, 00\bullet)$. Since $\mathcal{C}(s(c, 0)) = \{t_1, t_2\}$, the state is vanishing. Then, either t_1 or t_2 fires next, with probability α or $1 - \alpha$, and the next state $s(c, 1)$ is $(0110, \bullet 0 \bullet)$ or $(1001, 0 \bullet 0)$, respectively (note that α is defined by the SPN iff $C_{t_1} = C_{t_2}$). If $s(c, 1) = (0110, \bullet 0 \bullet)$, the new state is again vanishing, and t_2 fires immediately, leading to state $s(c, 2) = (0011, \bullet\bullet 0)$, also vanishing. Finally, t_3 fires, leading to the tangible (absorbing) state $s(c) = s(c, 3) = (0010, \bullet\bullet\bullet)$. If $s(c, 1) = (1001, 0 \bullet 0)$, the new state has two candidates, t_1 and t_3, with probability β and $1 - \beta$, respectively, and so on, until reaching state $(0010, \bullet\bullet\bullet)$. Any path from $s(0) = (1100, cc\bullet)$ results in $s(c) = (0010, \bullet\bullet\bullet)$ with probability one.

In the second SPN, the firing of t_3 also moves a token from p_1 to p_3, thus disabling t_1. The choices labeled with probabilities α vs. $1 - \alpha$ and β vs. $1 - \beta$ correspond to reaching tangible states $s(c) = (0011, \bullet\bullet\bullet)$ or $s(c) = (0010, \bullet\bullet\bullet)$. These two events happen with probability $\alpha + (1 - \alpha)\beta$ and $(1 - \alpha)(1 - \beta)$, respectively. The first choice probabilistically resolves a confusion, while the second resolves the conflict created by the confusion. It is interesting to note that, from a stochastic point of view, there is no difference between confusion and conflict. Both express the fact that different states can be reached according to the choice of sequentialization for transitions attempting to fire at the same time, and both must be resolved by assigning appropriate probabilities.

Finally, in the third SPN, the firing of t_3 moves all tokens (either zero or one) from p_1 to p_3. We still have a confusion between t_1 and t_2, but the resulting conflict between t_1 and t_3 is a new type of asymmetric conflict due to marking-dependent arc cardinalities. The firing of t_3 disables t_1, but not vice versa. Here, as in the first case, the tangible state at time c is $s(c) = (0010, \bullet\bullet\bullet)$ with probability one. However, sequences with different firing counts lead to this state.

If we are only interested in the tangible-to-tangible transitions, α and β are relevant only for the second SPN. If we are also interested in the number (but not the order) of firings, then α and β become relevant for the third SPN as well.

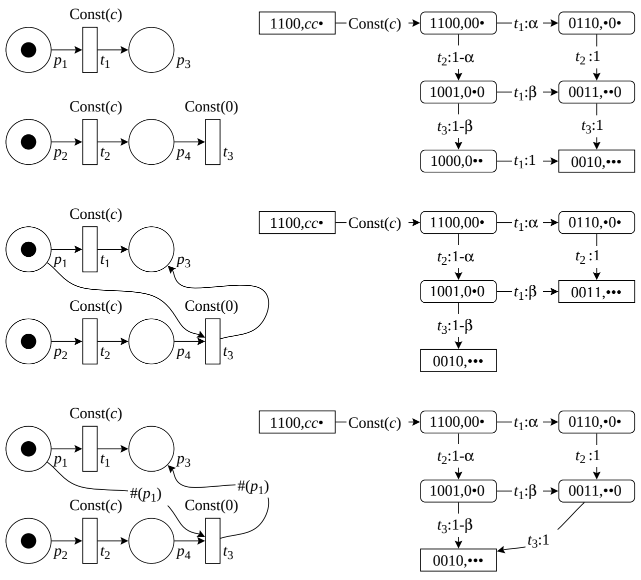

Figure 1: Stochastic conflicts and confusions.

5 Well-defined SPNs

We now introduce the concept of well-defined SPNs, and provide a state-space-based algorithm to determine when a SPN satisfies it.

First, we need to define the "stochastic process underlying a SPN". Often, this is taken to mean the marking at time θ, $\{\mu(\theta) : \theta \in \mathbb{R}^0\}$. However, this takes into account only information regarding the sojourn of tokens into places, not the firing of transitions. A more detailed process is needed:

Definition 5.1 The underlying stochastic process for a SPN, or *basic process*, is $\{(t^{[n]}, \theta^{[n]}, \mu^{[n]}) : n \in \mathbb{N}\}$, where, for $n > 0$, $t^{[n]} \in T$ is the n-th transition to fire, in marking $\mu^{[n-1]}$, $\theta^{[n]}$ is the time at which it fires, and $\mu^{[n]}$ is the new marking reached with this firing ($t^{[0]} = NULL$, $\theta^{[0]} = 0$, and $\mu(\theta) = \mu^{[\hat{n}]}$, where $\hat{n} = \max\{n : \theta^{[n]} \leq \theta\}$ is the number of firings up to time θ).

In our notation, $\mu^{[0]}$ and $\mu(0)$ represent different concepts. The former is the initial marking, before any firing. The latter is the marking at time 0^+, that is, after any firing occurring at time 0. The two can differ if a transition has a zero RFT initially.

Definition 5.2 A SPN is *well-defined* if its basic process is completely defined, that is, if $\forall n \in \mathbb{N}, \forall t \in T, \forall \theta \geq 0, \forall \mu \in \mathbb{N}^{|P|}$, $\Pr\{t^{[n]} = t, \theta^{[n]} \leq \theta, \mu^{[n]} = \mu\}$ is completely determined by the elements of the SPN.

Hence, any stochastic conflicts and confusions must be resolved by post-selection priorities or weights. If any is unresolved, the SPN is not well-defined.

In practice, we are normally interested in stochastic reward processes derived from the basic process [11].

Definition 5.3 $\{Y(\theta) \in \mathbb{R} : \theta \in \mathbb{R}^0\}$ is a *reward process* derived from the basic process through the reward structure (ρ, r) if: $Y(\theta) = \int_0^\theta \rho(\mu(u))du + \sum_{1 \leq n \leq \hat{n}} r_{t^{[n]}}(\mu^{[n-1]})$ $= \sum_{0 \leq n < \hat{n}} \rho(\mu^{[n]}) \cdot (\theta^{[n+1]} - \theta^{[n]}) + \rho(\mu[\hat{n}]) \cdot (\theta - \theta^{[\hat{n}]}) + \sum_{1 \leq n \leq \hat{n}} r_{t^{[n]}}(\mu^{[n-1]})$ where the reward rates $\rho : \mathbb{N}^{|P|} \rightarrow \mathbb{R}$ describe the rate at which reward is accumulated in each marking and the reward impulses $r : (T \times \mathbb{N}^{|P|}) \rightarrow \mathbb{R}$ describe the impulse accumulated when each transition fires in each marking.

It is then possible for the reward process to be well-defined, even when the basic process is not.

Definition 5.4 A SPN is *well-defined with respect to a reward structure* (ρ, r) if $\Pr\{Y(\theta) = y\}$ is completely determined by the elements of the SPN.

Corollary 5.1 A well-defined SPN is well-defined with respect to any reward structure.

Note that the SPN might not be well-defined even if the underlying untimed PN does not contain conflicts or confusions. In the first example of Fig. 1, if t_1, t_2, and t_3 are in different weight classes, α and β are not defined. Most reward processes of interest are not affected by these choices, so we could fire t_1 and t_2 "concurrently". Equivalently, we could enforce a particular sequentialization of t_1, t_2, and t_3, for example (t_1, t_2, t_3), and ignore the other "equivalent" ones. Detecting these situation is one of the goals of the structural analysis of [6]. However, the value of α and β is relevant for the reward process $Y(\theta) =$ "the number of firings of t_3 occurring while t_1 is disabled, during the interval $(0, \theta]$". In this case, t_1, t_2, and t_3 must be in the same weight class, resulting in $\alpha = \hat{w}_{t_1|\{t_1,t_2\}}(1100)$ and $\beta = \hat{w}_{t_1|\{t_1,t_3\}}(1001)$.

On the other hand, weights might not have to be defined in a SPN even if the underlying PN exhibits conflict and confusions. For an extreme case, consider a constant zero reward process, for which the SPN is always well-defined. More interestingly, consider the reward process $Y(\theta) =$ "the amount of time $\#(p1) = \#(p2)$, during the interval $(0, \theta]$" in the second SPN, or any reward process which disregards the firings of t_1 in the third SPN. The two SPNs are well-defined with respect to these processes, even if they are not well-defined (with respect to their basic processes).

We can focus on the instants of transition firings, since the semantics of the SPN evolution during instants of time where there is no firing is determined by the race behavior. In other words, stochastic conflicts and confusions are more likely to occur in DDP-SPNs [10] (or in other SPN definitions with discrete or mixed-time distributions) than in GSPNs [4, 6], but they can be treated with a uniform approach.

In the next section, we examine an algorithm to determine whether a SPN is well-defined with respect to a reward process (strictly speaking, it is a semi-algorithm because it is not guaranteed to terminate if the number of states to be considered is infinite, a common limitation for any approach based on the enumeration of the state space).

We observe that the definition of the basic and reward processes given above have a strong "simulation flavor". If indeed simulation is used for the solution, our algorithm can be applied every time a vanishing state is encountered. This will ensure that the SPN is well-defined or that, if it is not, none of the unresolved stochastic conflicts and confusions was encountered during the simulation runs. Either way, the results of the simulation are "reliable".

If the SPN has sufficient restrictions on the firing times, its underlying process might be a DTMC, CTMC, or a Markov-regenerative process (MRGP), also known as semi-regenerative process Numerical methods can then be employed for the transient or steady-state solution of the processes of interest, provided the number of reachable markings is finite [11, 10, 7, 8, 12]. In all cases, the algorithm requires to generate a "state-space" where the RFT information is implicitly encoded in the discrete part of the state (the marking, or the marking plus the "phase" of the RFT if discrete or continuous time phase distributions are used).

Regardless of whether simulation or a numerical method is employed, stochastic conflicts and confusions can arise only under these conditions: (1) multiple transitions attempt to fire at the same time (for GSPNs and related models, this can happen only with immediate transitions, since contemporary firings of timed transitions have probability zero), (2) at least two of them are in different weight classes, and (3) they have no post-selection priority defined between them. Our algorithm checks whether the SPN is not well-defined by testing for these conditions. If they are satisfied, it can further check whether the reward process(es) of interest are not well-defined as well.

6 The algorithm

The algorithm described in this section examines every possible ordering of contemporary transition firings starting from a vanishing state s and until tangible states are reached. The input is:

- a SPN $(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}, F, \tau^{[0]}, \nsucc, C, w)$, with a vanishing initial state, $s = s(0^-) = (\mu^{[0]}, \tau^{[0]})$, and
- a set of impulse reward functions $M = \{r^1, ..., r^{|M|}\}$, where $r_t^m(\mu) \in \mathbb{R}$ is the impulse reward obtained when firing transition t in marking μ according to the m-th reward structure, $1 \le m \le |M|$ (reward rates are not needed).

If the SPN is well-defined, the output is the set of tangible states $\mathcal{T}_s$ reachable from s in zero time $\mathcal{T}_s = \{\bar{s} : \Pr\{s(0) = \bar{s} \mid s(0^-) = s\} > 0\}$ and the set $\mathcal{P}_s = \{(\gamma_{\bar{s}}, \delta_{\bar{s}}) | \bar{s} \in \mathcal{T}_s\}$, where $\gamma_{\bar{s}} \in \mathbb{R}^{|M|}$ and $\delta_{\bar{s}} \in (0, 1]$. $(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_s$ represents all possible paths to a single tangible state $\bar{s} \in \mathcal{T}_s$ starting from the vanishing state s, where $\delta_{\bar{s}}$ is the sum of the path probabilities, and $\gamma_{\bar{s}} = (\gamma_{\bar{s}}^1, ..., \gamma_{\bar{s}}^{|M|})$ is a vector containing, for every impulse reward function $r^m \in M$, the accumulated reward value $\gamma_{\bar{s}}^m$ along these paths. The probability of reaching tangible state $\bar{s}$ in zero time from s is $\Pr\{s(0) = \bar{s} | s(0^-) = s\} = \delta_{\bar{s}}$, and the m-th expected accumulated impulse reward in reaching it is $E[Y^m(0) \mid (s(0^-) = s \wedge s(0) = \bar{s})] = \gamma_{\bar{s}}^m$.

If the SPN is not well-defined, the behavior of the algorithm depends on the type of problems encountered:

- If the probability $\delta_{\bar{s}}$ of reaching some state $\bar{s} \in \mathcal{T}_s$ from s cannot be determined, the algorithm assumes, conservatively, that all reward processes derived from the SPN are also not well-defined and it issues an error message. It could indeed be the case that the marking reached does not affect the further evolution of the reward process, but this would require a more global understanding of the stochastic process and of the reward structure than it seems reasonable to assume. We call this case *probability-confusion.*
- If the probability of reaching the states in $\mathcal{T}_s$ from s can be determined, but multiple firing sequences can lead to a given state $\bar{s} \in \mathcal{T}_s$ and the SPN does not provide enough information to compute their probability individually, the reward structures determine whether the derived reward processes are well-defined or not. If the different firing sequences result in the same accumulated impulse rewards, there is no need to distinguish among them, and the reward processes are well-defined. Otherwise, if different firing sequences leading to $\bar{s}$ accumulate different reward impulses and the SPN does not provide a way to compute their probability individually, $E[Y^m(0) | s(0^-) = s \wedge s(0) = \bar{s}]$ itself cannot be computed. In this case, the SPN is guaranteed to be not well-defined with respect to the corresponding reward structure (ρ, r^m) and the algorithm issues an error. We call this case *reward-confusion.*

For simplicity, the algorithm assumes that there exists a unique x for which $F_{t,u}(\mu, \tau_u, x) = 1$. If this is not the case, the firing of t results in multiple states, differing only in the RFTs, and the algorithm should be adjusted accordingly.

The reachability graph exploration could be restarted after a probability- or reward-confusion is found and the user has supplied the missing information to resolve it. However, the same task could be performed "on-line", continuing the exploration after the user redefines some weight classes or post-selection priorities, as long as the new information is "consistent" with that previously available. Fig. 2 shows the algorithm. Parameter calls can be *by value* (in), *by reference* (out), or *by value-reference* (inout).

The initial call is "trav(s; $\mathcal{T}_s$; $\mathcal{P}_s$)", where s is the initial vanishing state and $\mathcal{T}_s$ is initially empty. The first loop of the procedure "trav" partitions all candidate transitions of s into sets of candidate transitions $\hat{C}_i = \mathcal{C}(s) \cap C_i$ belonging to the same weight class.

The second (innermost) loop calculates, for every set $\hat{C}_i$, the set of paths $\mathcal{P}_{s,\hat{C}_i} \subseteq \mathcal{P}_s$, which start with the

```
procedure trav( in: $s \equiv (\mu, \tau)$; inout: $\mathcal{T}_s$; out: $\mathcal{P}_s$ )
$\mathcal{P}_s = \emptyset$;
foreach $C_i \in C$ do
  $\hat{C}_i = \mathcal{C}(s) \cap C_i$;   $\mathcal{P}_{s,\hat{C}_i} = \emptyset$;
  foreach $t \in \hat{C}_i$ do
    $\mu' = \mu - D^-_{\bullet,t}(\mu) + D^+_{\bullet,t}(\mu)$;
    $f_t = \hat{w}_{t|\hat{C}_i}(\mu)$;
    $\forall u \in T, \tau'_u =$ "new value according to $F_{t,u}(\mu, \tau_u, \cdot)$";
    $s' = (\mu', \tau')$;
    if $\mathcal{C}(s') \neq \emptyset$ then     # $s'$ IS VANISHING
      trav($s'$; $\mathcal{T}_s$; $\mathcal{P}_{s'}$);
      $\mathcal{P}_{s,t} = \bigcup_{(\gamma'_{\bar{s}}, \delta'_{\bar{s}}) \in \mathcal{P}_{s'}} \{ (\gamma_{\bar{s}}, \delta'_{\bar{s}} f_t) \mid \forall m \in \{1, \ldots, |M|\},$
                          $\gamma^m_{\bar{s}} = ({\gamma'}^m_{\bar{s}} + r^m_t(\mu) \delta'_{\bar{s}}) f_t \}$;
    else     # $s'$ IS TANGIBLE
      if $s' \notin \mathcal{T}_s$ then $\mathcal{T}_s = \mathcal{T}_s \cup \{s'\}$;
      $\mathcal{P}_{s,t} = \{ (\gamma_{s'}, f_t) \mid \forall m \in \{1, \ldots, |M|\}, \gamma^m_{s'} = r^m_t(\mu) f_t \}$;
    $\mathcal{P}_{s,\cap} = \bigcup_{(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_{s,t} : (\gamma'_{\bar{s}}, \delta'_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_i}} \{(\gamma_{\bar{s}} + \gamma'_{\bar{s}}, \delta_{\bar{s}} + \delta'_{\bar{s}})\}$;
    $\mathcal{P}_{s,\hat{C}_i} = \bigcup_{(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_{s,t} \cup \mathcal{P}_{s,\hat{C}_i} : (\gamma'_{\bar{s}}, \delta'_{\bar{s}}) \notin \mathcal{P}_{s,\cap}} \{(\gamma_{\bar{s}}, \delta_{\bar{s}})\} \cup \mathcal{P}_{s,\cap}$;
  if $\mathcal{P}_s = \emptyset$ then $\mathcal{P}_s = \mathcal{P}_{s,\hat{C}_i}$;
  else if $\mathcal{P}_s \neq \mathcal{P}_{s,\hat{C}_i}$ then stop;     # ERROR
end procedure
```

Figure 2: Algorithm for well-defined SPNs.

firing of transitions from $\hat{C}_i$. Every transition $t \in \hat{C}_i$ is fired according to its firing probability f_t leading to the new marking μ'. The new RFTs τ' are obtained from τ by applying the appropriate race policy [1], resulting in the new state $s' = (\mu', \tau')$.

If s' is vanishing, the recursive call "trav(s'; $\mathcal{T}_s$; $\mathcal{P}_{s'}$)" explores further reachable states and computes $\mathcal{P}_{s'}$, assuming that the vanishing reachability graph created from s is acyclic and that no probability- or reward-confusion occurred. The effect of firing t, which led to s', is added to $\mathcal{P}_{s'}$, resulting in the subset of all paths $\mathcal{P}_{s,t} \subseteq \mathcal{P}_s$, which start with the firing of t in s. Thus, the instantaneous impulse reward $r_t(\mu)\delta'_{\bar{s}}$ gained by firing t in μ, reaching tangible state $\bar{s}$, is added to the corresponding accumulated impulse rewards $\gamma'_{\bar{s}}$ of $\mathcal{P}_{s'}$. Then, all accumulated impulse rewards and the path probability of going from s' to $\bar{s}$ are multiplied by the firing probability f_t of the transition t which fired in s.

If s' is tangible, the recursion terminates, s' is added to the set of tangible states $\mathcal{T}_s$, if not already there, and a single initial direct path from s to s' is created in $\mathcal{P}_{s,t}$. The instantaneous impulse reward gained by the firing of t with probability f_t in μ is stored in $\gamma_{s'}$.

Finally, for every iteration over all $t \in \hat{C}_i$, the set $\mathcal{P}_{s,t}$ is obtained and aggregated in $\mathcal{P}_{s,\hat{C}_i}$ which is initially empty for every set $\hat{C}_i$. During a single iteration $\mathcal{P}_{s,t}$ and the old $\mathcal{P}_{s,\hat{C}_i}$ are unified in two steps, because multiple paths going to the same tangible state have to be eliminated and grouped into one, resulting in the new $\mathcal{P}_{s,\hat{C}_i}$:

1. Paths in both $(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_{s,t}$ and $(\gamma'_{\bar{s}}, \delta'_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_i}$ going to the same tangible state $\bar{s}$ are aggregated in the intersection $\mathcal{P}_{s,\cap}$, where the corresponding accumulated impulse rewards and path probabilities are added, $(\gamma_{\bar{s}} + \gamma'_{\bar{s}}, \delta_{\bar{s}} + \delta'_{\bar{s}})$.
2. All remaining paths of $\mathcal{P}_{s,t}$ and $\mathcal{P}_{s,\hat{C}_i}$ not included in $\mathcal{P}_{s,\cap}$, namely paths going to different tangible states, are aggregated in the new $\mathcal{P}_{s,\hat{C}_i}$, together with $\mathcal{P}_{s,\cap}$.

Probability- or reward-confusion cannot occur among transitions of $\hat{C}_i$: $\forall \hat{C}_i, \sum_{(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_i}} \delta_{\bar{s}} = 1$. The SPN is not well-defined if a vanishing state s is encountered such that $\mathcal{P}_{s,\hat{C}_i} \neq \mathcal{P}_{s,\hat{C}_j}$, for different sets of candidate transitions $\hat{C}_i \neq \hat{C}_j$. The depth-first search ensures that a minimal set of transitions causing the SPN to be not well-defined is found, since the test is performed in every vanishing state. Then, probability-confusion and reward-confusion correspond, respectively, to the conditions

$$\exists(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_i},\ \exists(\gamma'_{\bar{s}}, \delta'_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_j} :\ \delta_{\bar{s}} \neq \delta'_{\bar{s}} \text{ and}$$

$$\exists m, \exists(\gamma_{\bar{s}}, \delta_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_i},\ \exists(\gamma'_{\bar{s}}, \delta'_{\bar{s}}) \in \mathcal{P}_{s,\hat{C}_j} :\ \gamma^m_{\bar{s}} \neq {\gamma'}^m_{\bar{s}}.$$

The SPN is instead well-defined with respect to the reward structures if the following condition holds for all vanishing states s generated by the algorithm: $\forall \hat{C}_i, \hat{C}_j,\ \mathcal{P}_{s,\hat{C}_i} = \mathcal{P}_{s,\hat{C}_j}$.

In the description, we assume that the SPN starts at time zero in state $s(0^-)$ with probability $\Pr\{s(0^-) = s\} = 1$ and that it has no accumulated reward initially, $E[Y(0^-)] = 0$. When our algorithm is used within the overall state space exploration, these values can be used appropriately. This is possible because our definition assumes time-homogeneous distributions and reward structures.

Before concluding this section, we observe that the complexity of the proposed algorithm as stated is proportional to the number of paths going through the vanishing markings and ending in a tangible marking. However, if the set of vanishing states is kept during the recursive calls, the complexity is reduced to be proportional to the number of arcs in the vanishing portion of the reachability graph, thus analogous to any other algorithm for the exploration of the state space.

7 Conclusion

We extended Sanders's work [17], which is appropriate for constant reward impulses and constant cardinality arcs. Our marking-dependent generalizations require a non-trivial extension of all definitions, since firing the same set of transitions in a different order might lead to different markings. [17], and more recently [16], allow for an unspecified probability when choosing which transition to fire next, and examine after the fact whether that choice makes a difference (if it does, the model is not "well-specified"). We instead say that the SPN is not "well-defined" if the probabilistic evolution of the underlying stochastic process cannot be determined. This includes the sequencing of firings occurring at the same clock time so it is more restrictive.

We then introduce the notion of "well-defined with respect to a reward structure". A well-specified SPN according to [17], may be not well-defined with respect to a reward structure, if marking-dependent impulse rewards are used, while a not well-specified SPN may be well-defined with respect to a reward structure, if no reward impulses are used in the reward structure.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analyis of Stochastic Petri Nets. *IEEE Trans. Softw. Eng.*, 15(7):832–846, July 1989.

[2] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. Generalized Stochastic Petri Nets revisited: random switches and priorities. In *Proc. 2nd Int. Workshop on Petri Nets and Performance Models (PNPM'87)*, pages 44–53, Madison, Wisconsin, Aug. 1987. IEEE Comp. Soc. Press.

[3] M. Ajmone Marsan, G. Balbo, G. Chiola, G. Conte, S. Donatelli, and G. Franceschinis. An introduction to Generalized Stochastic Petri Nets. *Microelectronics and Reliability*, 31(4):699–725, 1991.

[4] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of Generalized Stochastic Petri Nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comp. Syst.*, 2(2):93–122, May 1984.

[5] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In G. Rozenberg, editor, *Adv. in Petri Nets 1987, Lecture Notes in Computer Science 266*, pages 132–145. Springer-Verlag, 1987.

[6] G. Chiola, M. Ajmone Marsan, G. Balbo, and G. Conte. Generalized Stochastic Petri Nets: a definition at the net level and its implications. *IEEE Trans. Softw. Eng.*, 19(2):89–107, Feb. 1993.

[7] H. Choi, V. G. Kulkarni, and K. S. Trivedi. Markov regenerative stochastic Petri nets. In G. Iazeolla and S. S. Lavenberg, editors, *Performance '93*. North-Holland, Sept. 1993.

[8] H. Choi, V. G. Kulkarni, and K. S. Trivedi. Transient analysis of deterministic and stochastic Petri nets. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993, Lecture Notes in Computer Science*, volume 691, pages 166–185. Springer-Verlag, 1993.

[9] G. Ciardo. Petri nets with marking-dependent arc multiplicity: properties and analysis. In R. Valette, editor, *Application and Theory of Petri Nets 1994, Lecture Notes in Computer Science 815 (Proc. 15th Int. Conf. on Applications and Theory of Petri Nets, Zaragoza, Spain)*, pages 179–198. Springer-Verlag, June 1994.

[10] G. Ciardo. Discrete-time Markovian stochastic Petri nets. In W. J. Stewart, editor, *Numerical Solution of Markov Chains '95*, pages 339–358, Raleigh, NC, Jan. 1995.

[11] G. Ciardo, A. Blakemore, P. F. J. Chimento, J. K. Muppala, and K. S. Trivedi. Automated generation and analysis of Markov reward models using Stochastic Reward Nets. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov Chains, and Queueing Models*, volume 48 of *IMA Volumes in Mathematics and its Applications*, pages 145–191. Springer-Verlag, 1993.

[12] G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Trans. Softw. Eng.*, 20(7):506–515, July 1994.

[13] A. Cumani. ESP - A package for the evaluation of stochastic Petri nets with phase-type distributed transitions times. In *Proc. Int. Workshop on Timed Petri Nets*, Torino, Italy, July 1985.

[14] M. K. Molloy. *On the integration of delay and throughput measures in distributed processing models.* PhD thesis, UCLA, Los Angeles, CA, 1981.

[15] M. K. Molloy. Discrete time stochastic Petri nets. *IEEE Trans. Softw. Eng.*, 11(4):417–423, Apr. 1985.

[16] M. A. Qureshi, W. H. Sanders, A. P. A. van Morsel, and R. German. Algorithms for the generation of state-level representations of stochastic activity networks with general reward structures. In *Proc. Int. Workshop on Petri Nets and Performance Models (PNPM'95)*, pages 180–190, Durham, NC, Oct. 1995. IEEE Comp. Soc. Press.

[17] W. H. Sanders. *Construction and solution of performability models based on Stochastic Activity Networks.* PhD thesis, Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI, 1988.