Critical paths in the Partial Order Unfolding of a Stochastic Petri Net

Anne Bouillard, Stefan Haar, Sidney Rosario

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Critical paths in the Partial Order Unfolding of a Stochastic Petri Net

Anne Bouillard and Stefan Haar and Sidney Rosario

N° 7003

juillet 2009

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--7003--FR+ENG

INRIA
RENNES

Critical paths in the Partial Order Unfolding of a Stochastic Petri Net

Anne Bouillard * and Stefan Haar † and Sidney Rosario ‡

Thème COM — Systèmes communicants
Projet DistribCom

Rapport de recherche n° 7003 — juillet 2009 — 14 pages

Abstract: In concurrent real-time processes, the speed of individual components has a double impact: on the one hand, the overall latency of a compound process is affected by the latency of its components. But, if the composition has race conditions, the very outcome of the process will also depend on the latency of component processes. Using stochastic Petri nets, we investigate the probability of a transition occurrence being critical for the entire process, *i.e.* such that a small increase or decrease of the duration of the occurrence entails an increase or decrease of the total duration of the process. The first stage of the analysis focuses on occurrence nets, as obtained by partial order unfoldings, to determine criticality of events; we then lift to workflow nets to investigate criticality of transitions inside a workflow.

Key-words: Stochastic Petri nets, partial order

* Anne.Bouillard@bretagne.ens-cachan.fr
† Stefan.Haar@inria.fr
‡ Sidney.Rosario@inria.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex (France)
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Chemins critiques dans le dépliage partiellement ordonné d'un réseau de Petri stochastique

Résumé : Dans les processus concurrents à temps-réel, la vitesse des composants individuels a un double impact : d'une part, le délai de bout-en-bout d'un processus est affecté par les délais de ses composants. Mais si la politique d'ordonnancement est la *race policy*, le résultat final lui-même dépendra des délais des composants. En utilisant les réseaux de Petri stochastiques, nous calculons la probabilité que l'occurrence d'une transition soit *critique*, c'est-à-dire qu'une légère variation de sa durée de tir modifie la durée totale du processus global. Dans un premier temps, nous nous intéressons aux réseaux d'occurrence qui, dépliés, nous permettent d'obtenir ordre partiel sur les événements et ainsi de déterminer les événements critiques. Ensuite, nous nous plaçons au niveau des *workflow nets* pour trouver les transitions critiques dans ce modèle.

Mots-clés : Réseau de Petri stochastique, ordre partiel

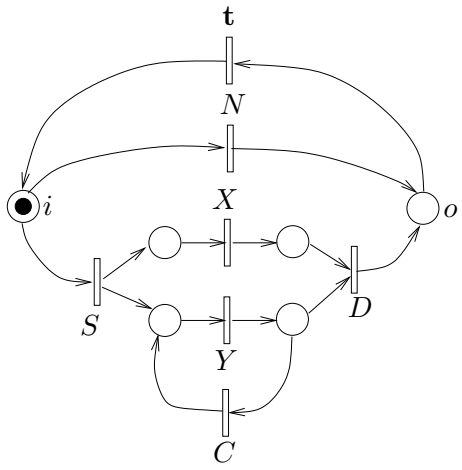

Figure 1: A workflow net.

1 Introduction

This paper studies the impact of component performances - measured by transition delays - on the global performance of a composite workflow. This impact analysis is complicated by the presence of concurrency and of conflict, both of which may either hide individual delays or accentuate their impact. To capture these effects, we consider continuous time processes within the framework of partial order unfolding semantics [8, 9, 13] of Petri nets.

To motivate the ideas, consider a machine servicing workflow, represented as a Petri net in Figure 1. A token in the initial place represents a client requesting that his machine be serviced. A client can revoke his request (by firing transition N), but this has to be done before the servicing process has been started (by the firing of S). The machine has two components C_X and C_Y, the operations servicing them are denoted by the transitions X and Y respectively. The component C_Y degrades when it is idle and has to be shipped to the client (denoted by transition D) as soon as possible after its servicing. If the machine can not be delivered (either because component C_X's servicing has not yet finished or because the shipping process has not yet begun), after a certain time the component C_Y has to be sent for servicing again (denoted by the firing of C).

Now, the latency of individual events has a double impact on the configurations. Firstly, the overall latency of a configuration is affected by the latency of its individual events: the latency of a configuration is a max-plus combination of the latencies of its individual events. A second impact of the latencies of the individual events is the choice of *configuration* itself, since an event with a shorter latency can pre-empt the occurrence of a conflicting event whose delay is larger. The authors of [15] have analyzed first-passage time in event structures for a *fixed* configuration; here, we also take into account the second impact of real-time durations, namely, on *choice*. A concurrent system generally has several qualitatively possible evolutions (or configurations) that could occur. By 'qualitatively' we mean the difference between runs that have *different sets of events* (rather than just different durations for the same set of events, which would be a merely quantitative distinction). Which one among the possible configurations actually occurs, depends in general on non-predetermined choices. In [3, 4, 6, 19], this is treated as a logical choice, or conflict; no timing issues are considered. Our approach combines the two orthogonal viewpoints, and considers timing and choice *jointly* rather than separately: the very shape and outcome of the process will depend on the latency of component processes.

We capture the random and asynchronous character of such behaviours, and the dependencies encountered, *e.g.*, in orchestrated processes, in Petri nets with stochastic delays on transitions. Although the work in this paper was initially motivated by Web-Services orchestrations, the scope of application encompasses all concurrent real-time processes.

This paper is organized as follows. In Section 2, we recall the definitions for Petri Nets and their unfoldings. In Section 3, we introduce stochastic delays in those structures, which are then used to compute the occurrence probability of an event; this is lifted in Section 4 to computing the probability for an event to be critical in an unfolding. Section 5 then lifts the analysis to the level of workflows and finishes the discussion of the running example; finally, Section 6 concludes.

2 Petri Nets

Let denote by $\mathbb{N} = \{0, 1, \ldots\}$ the set of natural numbers. We now formally define the models we use in the rest of this paper.

Definition 1. *A **net** is a tuple $N = (P, T, F)$ where*

- *$P \neq \emptyset$ is a set of **places**,*
- *$T \neq \emptyset$ is a set of **transitions** such that $P \cap T = \emptyset$,*
- *$F \subseteq (P \times T) \cup (T \times P)$ is a set of **flow arcs**.*

*A **marking** is a multiset m of places,* i.e. *a map from P to $\mathbb{N}$. A **Petri net** is a tuple $\mathcal{N} = (P, T, F, m)$, where*

- *(P, T, F) is a finite net, and*
- *$m : P \to \mathbb{N}$ is an **initial marking**.*

Elements of $P \cup T$ are called the *nodes* of $\mathcal{N}$. For a transition $t \in T$, we call ${}^\bullet t = \{p \mid (p, t) \in F\}$ the *preset* of t, $t^\bullet = \{p \mid (t, p) \in F\}$ the *postset* of t. A transition t is *enabled* in marking m if $\forall p \in {}^\bullet t,\ m(p) > 0$. This enabled transition can *fire*, resulting in a new marking $m' = m - {}^\bullet t + t^\bullet$; this firing relation is denoted by $m[t\rangle m'$. A marking m is *reachable* if there exists a sequence of transitions $t_0, t_1 \ldots t_n$ such that $m_0[t_0\rangle m_1[t_1\rangle \ldots [t_n\rangle m$. A net is *safe* if for all reachable markings m, $m(p) \subseteq \{0, 1\}$ for all $p \in P$. From now on, we will consider only safe nets, and consider markings simply as place sets, *i.e.* we use the symbol m for the *set* $\{p \in P \mid m(p) = 1\}$.

Let $\prec$ the transitive closure of F and $\preceq$ the reflexive closure of $\prec$. The *set of causes* or *prime configuration* of $x \in P \cup T$ is $[x] \triangleq \{y \mid y \preceq x\}$. Further, write $t_1 \#_{im} t_2$ for transitions t_1 and t_2 if and only if $t_1 \neq t_2$ and ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$; the *conflict relation* $\# \subseteq (T \cup P)^2$ is given by

$$a \# b \quad \Leftrightarrow \quad \exists t_a, t_b \in T :\ t_a \#_{im} t_b \ \wedge\ t_a \preceq a \ \wedge\ t_b \preceq b. \tag{1}$$

Definition 2. *A net $ON = (B, E, G)$ is an **occurrence net** if and only if it satisfies*

1. *$\preceq$ is a partial order;*
2. *for all $b \in B$, $|{}^\bullet b| \in \{0, 1\}$;*
3. *for all $x \in B \cup E$, $[x]$ is finite;*
4. *no self-conflict,* i.e. *there is no $x \in B \cup E$ such that $x \# x$;*
5. *the set of $\prec$-minimal nodes C_0 is contained in B and finite.*

The nodes of E are called *events*, those of B *conditions*. In the model below, durations are associated only to events, not to conditions; the starting instant of an event is thus determined by its predecessor *events* alone. We therefore define for convenience, for every e in E, ${}^\circ e$ by ${}^\circ e = {}^{\bullet\bullet} e$ and e° by $e^\circ = e^{\bullet\bullet}$. As for each place b, $|{}^\bullet b| \leq 1$, the firing of event e requires that all events in ${}^\circ e$ have fired previously. We also suppose that E contains an an *initial* event $\perp$ such that ${}^\circ e = \{\perp\}$ iff e is $\prec$-minimal in $E \setminus (\{\perp\} \cup {}^\bullet \perp)$.

A *prefix* of ON is any subnet spanned by a downward closed subset $\pi \subseteq B \cup E$, *i.e.* such that for every $x \in \pi$, $[x] \subseteq \pi$. Prefix κ is a *configuration* if and only if it is conflict-free, *i.e.* $x \in \kappa$ and $x \# y$ imply $y \notin \kappa$. Denote as $\mathcal{C}(ON)$ the set of ON's configurations. Call any $\subseteq$-maximal element of $\mathcal{C}(ON)$ a **run** of ON. Denote the set of ON's runs as $\Theta(ON)$, or Θ if no confusion can arise. A pair $(x, y) \in (B \cup E)^2$ of nodes is *concurrent*, written x **co** y, if and only if neither $x \preceq y$ nor $y \preceq x$ nor $x \# y$ hold. Further, any set of conditions $W \subseteq B$ such that all conditions in W are pairwise concurrent, is called a *co-set*. A $\subseteq -$maximal co-set is a *cut*.

Occurrence nets are the mathematical form of the *partial order unfolding semantics* for Petri nets [9]; although more general applications are possible, we will focus here on unfoldings of *safe* Petri nets only.

If $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$ are nets, a *homomorphism* is a mapping $h : P_1 \cup T_1 \to P_2 \cup T_2$ such that

- $h(P_1) \subseteq P_2$ and
- for every $t_1 \in T_1$, the restriction of h to ${}^\bullet t_1$ is a bijection between the set ${}^\bullet t_1$ in N_1 and the set ${}^\bullet h(t_1)$ in N_2, and similarly for $t_1{}^\bullet$ and $(h(t_1))^\bullet$.

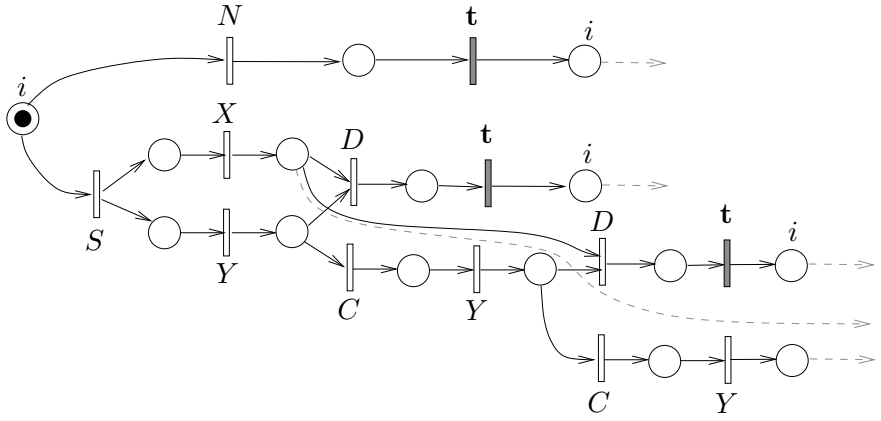

Figure 2: The (partial) unfolding of the workflow net of Figure 1.

A *branching process* of safe Petri net $\mathcal{N} = (N, m_0)$ is a pair $\beta = (ON, \pi)$, where $ON = (B, E, G)$ is an occurrence net, and π is a homomorphism from ON to N such that:

1. The restriction of π to C_0 is a bijection from C_0 to m_0, and
2. for every $e_1, e_2 \in E$, if ${}^\bullet e_1 = {}^\bullet e_2$ and $h(e_1) = h(e_2)$ then $e_1 = e_2$.

Branching processes $\beta_1 = (ON_1, \pi_1)$ and $\beta_2 = (ON_2, \pi_2)$ for $\mathcal{N}$ are isomorphic iff there exists a bijective homomorphism $h : ON_1 \to ON_2$ such that $\pi_1 = \pi_2 \circ h$. The unique (up to isomorphism) maximal branching process $\beta = (\mathcal{U}_\mathcal{N}, \pi)$ of $\mathcal{N}$ is called the *unfolding* of $\mathcal{N}$.

Following [9], the unfolding of $\mathcal{N}$ can be computed using the canonical algorithm given below (we omit any cut-off criteria here since they are not essential for our purposes). Let $\beta = (ON_\beta, \pi_\beta)$ be a branching process of $\mathcal{N} = (P, T, F, m_0)$, where $ON_\beta = (B_\beta, E_\beta, G_\beta)$. Denote as $\mathbf{PE}(\beta) \subseteq T \times \mathcal{P}(B_\beta)$ the set of *possible extensions* of β, *i.e.* of the pairs (t, W) such that

- W is a co-set of ON_β,
- ${}^\bullet t = \pi_\beta(W)$,
- E_β contains no event e such that $\pi_\beta(e) = t$ and ${}^\bullet e = W$.

The unfolding procedure adapted from [9] for safe Petri net $\mathcal{N} = (N, m_0)$ is then:

- Let $C_0 \triangleq m_0 \times \{\emptyset\}$ and initialize $\beta = (C_0, \emptyset, \emptyset, \pi_\beta)$ with π_β sending all conditions in C_0 to the corresponding place in m_0.
- For given $\beta = (ON_\beta, \pi_\beta)$ with $ON_\beta = (B_\beta, E_\beta, G_\beta)$, compute $\mathbf{PE}(\beta)$ and replace
  - E_β by $E_\beta \cup \mathbf{PE}(\beta)$,
  - B_β by $B_\beta \cup V$, where $V \triangleq \{(p, e) \mid e \in \mathbf{PE}(\beta), p \in \pi_\beta(e)^\bullet\}$, and
  - G_β by $G_\beta \cup U$, where

$$\begin{aligned} U \quad \triangleq \quad & \{(b, (t, W)) \mid (t, W) \in \mathbf{PE}(\beta), b \in W\} \\ & \cup \{(e, (p, e)) \mid e \in \mathbf{PE}(\beta), p \in \pi_\beta(e)^\bullet\}; \end{aligned}$$

    finally, extend π_β to the new nodes in the natural way, *i.e.* $(t, W) \mapsto t$ and $(p, e) \mapsto p$.

Figure 2 shows a prefix of the unfolding for the net of Figure 1. Note the multiple occurrences of the looping transition **t** (shaded in the figure). In this figure, every occurrence of **t** corresponds to a distinct way in which the net of Figure 1 reaches the output transition o. Every occurrence of **t** is followed by the place i, which denotes that the net of Figure 1 has come back to its initial marking.

3 Adding Time and Probability

Until now, we have defined structures that model the concurrency between events. The applications and properties we are interested in (mainly Web-services) are strongly related to the timed behavior of those structures (cf. [10]).

3.1 Definitions and Assumptions

We will consider safe Petri nets $\mathcal{N} = (P, T, F, m_0)$ such that each transition $t \in T$ is equipped with a probability law $\mathbb{P}_t$ whose support is contained in $[0, \infty)$. $\mathbb{P}_t$ gives the law of the *delay* δ_t for firing t after t is enabled. If t becomes enabled at time τ, a new realization $\delta_t(\omega)$ of δ_t is drawn from $\mathbb{P}_t$, independently of other transitions and of previous realizations of δ_t. If t is continuously enabled during the interval $[\tau, \tau + \delta_t(\omega)]$, then t fires at time $\tau + \delta_t(\omega)$, otherwise it has been *preempted*. Upon unfolding $\mathcal{N}$, the events of $\mathcal{U}_\mathcal{N}$ inherit the delay law from the corresponding transitions of $\mathcal{N}$: we obtain a family $(\delta_e)_{e\in E}$ such that $\delta_e \sim \delta_{\pi(e)}$.

Note that our approach is not to be confounded with *Timed Event Structures*, see [11],where delays merely indicate when an event *may* occur (but is not forced to). Let $ON = (B, E, G)$ be the unfolding of $\mathcal{N}$. Each value $\omega = (\delta(e))_{e\in E}$ in the space $\Omega_E \triangleq [0, \infty)^E$ will be seen to yield a unique configuration θ of ON.

We make the following assumptions:

Assumption 1. *The measures $(\mathbb{P}_e)_{e\in E}$ are pairwise independent.*

Assumption 2. *No $\mathbb{P}_e$ has atoms: $\forall\ e \in E :\ \forall x \in [0, \infty) :\ \mathbb{P}_e(\{x\}) = 0$.*

Heights. The *height* of an event e is defined (see, *e.g.*, [12]) recursively by

$$\mathcal{H}(e, \omega) \triangleq \max_{e' \in {}^\circ e} \{\mathcal{H}(e', \omega)\} + \delta(e) \quad \text{and} \quad \mathcal{H}(\perp, \omega) = 0; \tag{2}$$

a configuration κ has height

$$\mathcal{H}(\kappa, \omega) \triangleq \max_{e\in\kappa} \{\mathcal{H}(e, \omega)\}. \tag{3}$$

Note that only the causality relation and the delays are relevant in the computation of $\mathcal{H}(e, \omega)$, conflicting events have no influence. In other words, for every event e of E, $\mathcal{H}(e, \omega)$ is defined, regardless of whether or not e actually occurs. We may thus apply different firing policies without modifying $\mathcal{H}(\bullet, \bullet)$; on the other hand, the *occurrence* of e will depend on ω through the firing policy. Here, all decisions will be made according to *race policy*: the first event whose delay expires first preempts its competitors. We use Ω instead of Ω_E.

For $\tau \in [0, \infty)$, denote as

$$E_\tau(\omega) \quad \triangleq \quad \{e \mid \mathcal{H}(e, \omega) \leqslant \tau\} \tag{4}$$

the random set of those events whose height is bounded by τ.

Theorem 1. *Under assumptions 1 and 2, the following properties hold.*

1. *$\mathcal{H}(e, \omega) < \infty$ for all $e \in E$ and almost all $\omega \in \Omega$.*
2. *$\mathcal{H}(e, \omega) \neq \mathcal{H}(e', \omega)$ almost surely for any $e, e' \in E$ such that $e \neq e'$.*
3. *For all $\tau \in [0, \infty)$, the set $E_\tau(\omega)$ is finite for almost all ω.*

Proof:

1. Obvious.
2. The claim is obvious for $e \prec e'$ or $e' \prec e$, so assume neither holds. Let $H(e)$ be the random variable given by $(\mathcal{H}(e, \omega))_{\omega\in\Omega}$, and define analogously $H(\kappa)$ for any configuration κ. Let $\check{x}$ be the configuration

$$\check{x} \quad \triangleq \quad [x]\backslash\{x\}.$$

Then $A \triangleq H(e) - H(\check{e})$ and $A' \triangleq H(e') - H(\check{e'})$ are independent of one another and of $H(\check{e})$ and $H(\check{e'})$. In particular, A and A' are independent of $B \triangleq H(\check{e'}) - H(\check{e})$, and thus A is independent of $A' + B$. Now, for any ω,

$$\mathcal{H}(e, \omega) = \mathcal{H}(e', \omega) \quad \Leftrightarrow \quad A(\omega) = A'(\omega) + B(\omega).$$

Now, if X and Y are two atomless independent real random variables, then $X - Y$ is also atomless (see [18, Chapter 5] for example). Then $\mathbb{P}(X = Y) = \mathbb{P}(X - Y = 0) = 0$. Setting $X \triangleq A$ and $Y \triangleq A' + B$, one can conclude.

3. Assume there exist $\varepsilon > 0$ and $\tau \in [0, \infty)$ such that

$$\mathbb{P}\{\omega : |E_\tau(\omega)| = \infty\} \geqslant \varepsilon, \tag{5}$$

and let $\tau_\varepsilon \triangleq \inf\{\tau \in [0,\infty) \mid (5) \text{ holds}\}$. If

$$\mathbb{P}\{\omega : |E_{\tau_\varepsilon}(\omega)| = \infty\} \geqslant \varepsilon, \tag{6}$$

then, by construction of τ_ε, there is a positive probability that an infinite number of firings must occur simultaneously at time τ_ε. However, $\mathcal{N}$ is safe and finite, therefore only a finite number of transition firings can be simultaneously enabled; from this contradiction, we conclude that

$$u \triangleq \mathbb{P}\{\omega : |E_{\tau_\varepsilon}(\omega)| = \infty\} < \varepsilon.$$

Thus, for every $\epsilon > 0$,

$$\mathbb{P}\{|E_{\tau_\varepsilon+\epsilon}(\omega)| = \infty \mid |E_{\tau_\varepsilon}(\omega)| < \infty\} \geqslant \frac{\varepsilon - u}{1 - u} > 0.$$

Since $\mathcal{N}$ is finite, this implies that there exists some transition t such that

$$\mathbb{P}\{t \text{ fires } \infty\text{ly often in } [\tau_\varepsilon, \tau_\varepsilon + \epsilon] \mid |E_{\tau_\varepsilon}(\omega)| < \infty\} > 0.$$

But since $\mathcal{N}$ is safe, no two occurrences of the same transition are enabled simultaneously. Hence, since the $\delta_t(n_k, \omega)$ are i.i.d., this implies the existence of a series $n_1 < n_2 < \ldots$ of indexes such that

$$\mathbb{P}\left\{\sum_{k=1}^{\infty} \delta_t(n_k, \omega) < \alpha_t(2^{-k})\right\} > 0, \tag{7}$$

where $\alpha_t(x)$ is the x-quantile of the distribution of δ_t. Note that by assumption 2, one has that $\mathbb{P}(\delta_t(n_k, \omega) = 0) = 0$, and therefore $\alpha_t(x) > 0$ for all $x > 0$. By construction,

$$\sum_{k=1}^{\infty} \mathbb{P}\{\delta_t(n_k, \omega) < \alpha_t(2^{-k})\} \leqslant \sum_{k=1}^{\infty} 2^{-k} = 1; \tag{8}$$

but then the Borel-Cantelli lemma[1] contradicts (7), and we are done.

□

3.2 Occurrence of an event

Let us define with the above notation the occurrence predicate $occ(e, \omega)$; it is true if and only if e effectively occurs under ω; that is, all of e's preconditions are satisfied under ω, and none of e's fast adversaries, occurs. Formally we have the following definition:

Definition 3. *Set $occ(\bot, \omega)$ to **true** for all $\omega \in \Omega$, and for any $\omega \in \Omega$, let recursively $occ(e, \omega)$ be **true** iff*

$$\begin{aligned} &\forall\ e' \in {}^\circ e \ : \ occ(e', \omega) \\ \wedge\ &\forall\ e' \in \mathbf{check}(e, \omega) \ : \ \neg occ(e', \omega), \end{aligned} \tag{9}$$

where

$$\mathbf{check}(e, \omega) \triangleq \{e' \mid e \# e' \wedge \mathcal{H}(e', \omega) \leqslant \mathcal{H}(e, \omega)\}. \tag{10}$$

[1] see *e.g.* Lemma 8.1 in P. Brémaud. Markov Chains. Gibbs Fields, Monte Carlo Simulation, and Queues. Texts in Applied Mathematics **31**, Springer 1999.

Further, for all $e \in E$, define $Occ(e) \triangleq \{\omega \mid occ(e,\omega)\}$. In other words, $occ(e)$ holds iff event e eventually occurs, under ω and the race policy. Letting $\mathbf{R}(\omega) \triangleq \{e \in E \mid occ(e,\omega)\}$ the set of events that occur under ω, we have:

Lemma 1. *For almost all* $\omega \in \Omega$, $\mathbf{R}(\omega) \in \Theta(ON)$.

Proof. $\mathbf{R}(\omega)$ is a configuration by construction. For maximality, suppose there exists $e \notin \mathbf{R}(\omega)$ such that $\mathbf{R}(\omega) \cup \{e\} \in \mathcal{C}(ON)$. By Theorem 1, e has finite height, and $\mathbf{check}(e,\omega)$ is almost surely finite. By choice of e,

1. ${}^\circ e \subseteq \mathbf{R}(\omega)$, and
2. there is no $\overline{e} \in \mathbf{R}(\omega)$ such that $e \# \overline{e}$.

But this implies that $\mathbf{check}(e,\omega)$ contains no e' with $occ(e',\omega)$, which implies $occ(e,\omega)$, contradicting our assumption. □

3.3 Probability of occurrence

The occurrence of an event e under any ω is determined by $[e]$ and the set of events $\{e' \mid e \# e'\}$ (see definition of $occ(e,\omega)$). In fact, the latter set can be further restricted to events e' that are in *minimal conflict* with e.

Minimal Conflict: If $e \# e'$, but there exist events $e_1 \preceq e, e_1' \prec e'$ such that $e_1 \# e_1'$, then $e \# e'$ can be seen as a conflict *derived* from e_1 and e_1'. The height of e' in any run ω, $\mathcal{H}(e',\omega)$, can not affect the occurrence of e which is decided by the race between the mutually conflicting events e_1 and e_1'. This inspires the following definition for minimal conflict [3, 4]:

Definition 4 (Minimal Conflict)**.** *Two events* $e, e' \in E$ *are in* minimal conflict*,* $e \#_\mu e'$ *iff:* $([e] \times [e']) \cap \# = \{(e,e')\}$.

Thus the set of events which completely determine the occurrence of an event e is a prefix containing e which is closed under minimal conflict. This set $B(e)$ is formally defined as:

1. $e \in B(e)$;
2. if $e_1 \in B(e)$ and $e_2 \prec e_1$, then $e_2 \in B(e)$;
3. if $e_1 \in B(e)$ and $e_1 \#_\mu e_2$, then $e_2 \in B(e)$.

Calculating $\mathbb{P}(Occ(e))$:

$$Occ(e) \;=\; \{\omega \mid occ(e,\omega)\} \;=\; \{\omega \mid e \in \mathbf{R}(\omega)\}.$$

$Occ(e)$ can be partitioned into equivalence classes of runs in the following way: in any equivalence class C, any two runs ω_1, ω_2 are such that $\mathbf{R}(\omega_1) \cap B(e) = \mathbf{R}(\omega_2) \cap B(e)$. For any run $\omega \in C$, the set of events $\{e' | occ(e',\omega), e' \in B(e)\}$ is the same, denoted by κ_C. Denote the set of equivalence classes of $Occ(e)$ by $Occ(e)_{/B(e)}$.

$$Occ(e) \;=\; \bigcup_{C \in Occ(e)_{/B(e)}} C$$

and so $\mathbb{P}(Occ(e)) = \sum_{C \in Occ(e)_{/B(e)}} \mathbb{P}(C)$. Let $p(\kappa_C)$ denote each term of this summation.

$B(e)$ is an occurrence net in itself. For every equivalence class C in $Occ(e)_{/B(e)}$, the set κ_C is a maximal configuration of $B(e)$ which contains e, and vice-versa. Hence

$$\mathbb{P}(Occ(e)) \;=\; \sum_{e \in \kappa_C \in \Theta_{B(e)}} p(\kappa_C). \tag{11}$$

We thus need to compute all possible ways in which a maximal configuration κ_C could occur in $B(e)$. This can be done for any occurrence net ON using a Markov chain, which is a graph of configurations κ of ON with probabilistic transitions. This graph is constructed as below:

1. The states of the graph are the configurations κ of ON.

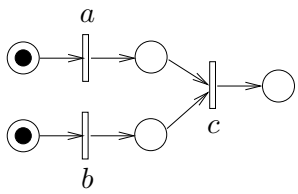

Figure 3: An occurrence net.

2. Define the set of events enabled in a configuration κ_1 as

$$\text{enab}(\kappa_1) \quad \triangleq \{e \in E \setminus \kappa_1 \mid \kappa_1 \cup \{e\} \in \mathcal{C}(ON)\} \tag{12}$$

The probability to go from state κ_1 to state $\kappa_2 = \kappa_1 \cup \{e\}$ is

$$\mathcal{P}_{\kappa_1,\kappa_2} = \frac{\lambda_e}{\sum_{e' \in \text{enab}(\kappa_1)} \lambda_{e'}}. \tag{13}$$

3. Transitions between any other states have zero probability.

The initial state of the Markov chain is the minimal configuration $\{\perp\}$ and the maximal states are the maximal configurations of ON. Let $\text{prec}(\kappa)$ denote the set of immediate predecessor states of κ in the graph. We obtain $p(\kappa)$ recursively as:

$$p(\kappa) = \sum_{\kappa' \in \text{prec}(\kappa)} p(\kappa').\mathcal{P}_{\kappa',\kappa}, \tag{14}$$

taking $p(\{\perp\}) = 1$.

In general, Equation (11) can be computed only when $B(e)$ is finite.

4 Critical Chains in Occurrence Nets

4.1 When is a critical for b?

We now turn to the central problem of *criticality*. As a simple example consider the occurrence net in Figure 3 and the only maximal configuration $u = \{a, b, c\}$. Any change in the delay δ_c of event c will affect the delay δ_u of configuration u. Event c is thus *critical* for the configuration u for all possible delay values of a, b and c. The same cannot be said for events a and b: if $\delta_a > \delta_b$, a increase or decrease in δ_b by an amount ϵ such that $\delta_a > \delta_b + \epsilon$ does not affect δ_u. Similarly, a is non-critical when $\delta_b > \delta_a$. Events a and b are thus critical for configuration u only in certain situations, depending on the delays of both a and b.

We will study first criticality of *events* for a *configuration*, and then move on to asking whether a *transition* is critical. The latter will become meaningful in the context of *workflow nets* and their unfoldings.

To formalize our question, let u_e be the tuple from $[0, \infty)^E$ whose e-component is 1 and all of whose other components are 0. We are interested in situations in which the delay of e is critical for the delay of configuration κ, in the following sense:

$$\mathit{crit}(e, \kappa, \omega) \quad \Longleftrightarrow \quad \forall\, \varepsilon > 0 :\ \mathcal{H}(\kappa, \omega) < \mathcal{H}(\kappa, \omega + \varepsilon \cdot u_e).$$

In the example of Figures 1 and 2, let us ask whether the first occurrence of X (called X_0) is critical for the first occurrence D_0 of D, assuming both occur. This is the case iff the delay required by X_0 is longer than that of Y_0. Here and in the remainder of the paper, let all exponential transition delay parameters be denoted by λ with the name of the transition as subscript, *i.e.*, λ_X for the delay parameter at transition X, etc. We obtain, by independence of the delays,

$$\mathbb{P}(\mathit{crit}(X_0, [D_0], \omega)) \quad = \quad \frac{\lambda_Y}{\lambda_X + \lambda_Y}.$$

An event e is in $Crit(e,\omega)$ if it is critical for the configuration that occurs with respect to ω. That is, for all positive but "small enough" increases of $e's$ delay, that increase is also "felt" by $\mathbf{R}$:

$$Crit(e,\omega) := \{e \mid \exists \eta \text{ s. th. } \forall\, \varepsilon \in]0,\eta[\colon\ \mathcal{H}(\mathbf{R}(\omega),\omega) < \mathcal{H}(\mathbf{R}(\omega+\varepsilon\cdot u_e),\omega+\varepsilon\cdot u_e)\}\,. \tag{15}$$

Indeed, due to Assumption 1), no pair of events can have the same height. Then, for sufficiently small increases of latencies the events that occur do not change and $R(\omega) = R(\omega+\varepsilon\cdot u_e)$. If ε is to large, there could be a change in the run that occurs and the total height of the occurring run could become smaller.

The definitions given for defining critical events are valid only for finite configurations, *i.e.* we consider all heights to be finite. The notion of critical event cannot be well-defined for an infinite configuration. For example, take a configuration $\kappa = e_0, f_1, e_1, f_1, \ldots$, where $\forall i,j \in \mathbb{N}, e_i < e_{i+1}$ and $f_j < f_{j+1}$ and $\neg(e_i \# f_j)$. If event e_i occurs at time t, then, at that time, events $e_0,\ldots,e_i$ are critical, and not event f_j. Conversely, if event f_j occurs at time t, events $f_0,\ldots,f_j$ are critical and not events e_i. Making t grow to infinity, should we consider that the critical events are all the events, or none? Whatever our choice, it will not articulate any meaningful information about our system.

Problem CRIT Given a finite configuration κ What is the probability $\mathbb{P}(\{\omega \mid crit(e,\kappa,\omega)\})$ for the delay of $e \in \kappa$ to be critical?

For notational convenience, write $x \ll_\omega y$ to say that the delay of x is critical for the height of $[y]$ in ω. More formally, we have the following definition.

Definition 5. *For all ω, let $\ll_\omega$ be the smallest reflexive relation on E that satisfies:*

1. *For any $u \in E$ and $x \in {}^\circ u$, $x \ll_\omega u$ if and only if $\forall\, e \in {}^\circ u \setminus \{x\}:\ \mathcal{H}(e,\omega) < \mathcal{H}(x,\omega)$.*
2. *For all ω, relation $\ll_\omega$ is transitive: $x \ll_\omega y \ll_\omega z \Rightarrow x \ll_\omega z$.*

*A **critical chain** of ON for ω is a maximal set $\mathbf{cc} \subseteq \{e \mid Crit(e,\omega)\}$ such that for all $x,y \in \mathbf{cc}$, either $x \ll_\omega y$ or $y \ll_\omega x$.*

Note that if there exists ω such that $x \ll_\omega y$, then $x < y$. In principle, there can be more than one critical chain for a given ω; however, under Assumption 1, the set of those ω has measure 0 under $\mathbb{P}$, *i.e.* $\ll_\omega$ is uniquely defined for almost all ω.

The following lemma ensures that every critical chain contains a minimal event and is the finite sequence of events $x_0,\ldots,x_n$ where $x_0 \in \min(E)$ and $\forall i \in \{1,\ldots,n\}$, $x_{i-1} \in {}^\circ x_i$. The following lemma is a direct consequence of Definition 5.

Lemma 2. *For every event y such that $Crit(y,\omega)$ and $y \neq \bot$, $\exists x \in E$ such that $x \ll_\omega y$ and $x \in {}^\circ y$.*

In the remainder of this section, we first give an algorithm to compute the critical events for a given configuration and given timings on the events, then we describe a method to solve Problem **CRIT**.

4.2 Critical events for a given configuration and given timings

For a given finite configuration κ and a given ω, one can find a critical chain of critical events using Algorithm 1. Indeed, from the definition, an event of κ that has the maximum height is critical. Then, one can find a critical chain that ends with that event. Then, from Lemma 2 and Definition 5, it is easy to see that at each step of the loop, one computes a critical event that is a predecessor of the last computed critical event. This gives a maximal critical chain (the condition $[e'] \setminus \{e'\} = \emptyset$ is equivalent to $e' \in \min(E)$).

4.3 Computation of the Criticality Probability

We now give a method to compute the probability of an event to be critical. Note that as the critical character of an event depends on the future of that event, we need the configurations to be finite and in finite number.

As stated in the previous section, the behavior of the net can be modeled by a Markov chain whose states are the configurations and we will use the notations already defined. Since a Markov chain can be seen as a directed graph labeled by the transition probabilities or by the events on the arcs, we will use graph theoretic terminology. Each maximal path of the chain (from the empty configuration to a run κ) defines an order of occurrence of the event in the configuration κ. From this order, one can define the critical chain on the path; if the events that occur of that path are in their occurrence order $e_1,\ldots,e_n$, we have:

Algorithm 1 Critical chain

Take $e \in \kappa$ such that $\mathcal{H}(e, \omega) = \mathcal{H}(\kappa, \omega)$;
cc $\leftarrow [e]$;
$\kappa \leftarrow [e] \setminus \{e\}$;
while $\kappa \neq \emptyset$ **do**
  Take $e' \in \kappa$ such that $\mathcal{H}(e', \omega) = \mathcal{H}(\kappa, \omega)$;
  cc $\leftarrow e' ::$ **cc**;
  $\kappa \leftarrow [e'] \setminus \{e'\}$;
end while
return cc.

1. e_n is critical;
2. if e_k is critical and if $\{i \mid i < k, e_i < e_k\}$ is non-empty, define $i_0 \triangleq \max\{i \mid i < k, e_i < e_k\}$. Event i_0 is then the last event to occur before e_k. This event does not exist (the considered set is empty) if e_k is a minimal event. Then
   - $e_{i_0} \ll_\omega e_k$ and
   - $\nexists j \in \{i_0 + 1, \ldots, k-1\}$ such that $e_{i_0} \ll_\omega e_j \ll_\omega e_k$: e_{i_0} is critical and there is no critical event between e_{i_0} and e_k.

Under Assumption 1, the critical chain constructed in this way is unique with probability 1. The occurrence of an event e depends only on the "past" of event e, whereas the critical character of an event depends only on the "future" of e (that is, the events that occur after it), thanks to the memoryless properties of the exponential distribution. Thanks to that property, the past and the future of e can be separated in the computations. We now develop an algorithm to compute that probability.

We will use the following notations:

- For two states κ_1, κ_2 of the chain $\mathcal{P}(\kappa_1, \kappa_2)$ is the probability to reach κ_2 from κ_1.
- We denote by $\mathcal{P}_{crit}(\kappa, e)$ the probability of e to be critical if starting from configuration κ, e is a minimal event (that is, if $\kappa \cup \{e\}$ is a configuration). $\mathcal{P}_{crit}(\kappa, e) = 0$ otherwise. For a maximal event e of a longest run κ, one has $\mathcal{P}_{crit}(\kappa - \{e\}, e) = \mathcal{P}_{\kappa - \{e\}, \kappa}$.

Let e be a maximal event – recall that the maximal events are those that may occur at the latest time in κ – of a run κ. From the above, configuration κ occurs and e is critical if and only if configuration κ occurs and e is the last event to occur. The probability for that event is $\mathcal{P}(\emptyset, \kappa - \{e\}) \cdot \mathcal{P}_{\kappa - \{e\}, \kappa}$.

Now, let us compute $\mathcal{P}_{crit}$ for the other arcs. Let (κ_1, κ_2) be an arc such that every arc successor of it has its $\mathcal{P}_{crit}$ computed. It is always possible to find such an arc because the graph is acyclic. Using a topological sort, one can find an order on the configurations such that this condition is always satisfied. Denote by e the unique event in $\kappa_2 - \kappa_1$.

From the choice of configuration κ_1, for a run κ that contains $\kappa_1 \cup \{e\}$, if e is the first event to occur, e is critical iff

$$\exists f \in e^\circ \cap \kappa \text{ such that } {}^\circ f \cap (\kappa - \kappa_1) = \{e\} \qquad \text{and } f \text{ is critical .} \tag{16}$$

Indeed, from Definition 5.1, e can be critical only if there is a critical event f in e°. Moreover, if there is a event $u \in {}^\circ f \cap (\kappa - \kappa_1)$, then $\mathcal{H}(u) > \mathcal{H}(e)$ and u is critical, not e.

Equation (16) leads to a method to recursively compute $\mathcal{P}_{crit}$. Let F be the set of events f satisfying Equation (16) for a run κ and $A_F = \{(\kappa^i, \kappa'^i), i \in \{1, \ldots, m\}\}$ be the set of arcs labeled by an event in F reachable from κ_2 (we denote the label of (κ^i, κ'^i) by f_i). We have

$$\mathcal{P}_{crit}(\kappa_1, e) = \sum_{i=1}^{n} \mathcal{P}(\kappa_1, \kappa_2)\mathcal{P}(\kappa_2, \kappa^i)\mathcal{P}(\kappa^i, \kappa'^i)\mathcal{P}_{crit}(\kappa^i, f_i).$$

This formula can be explained in the following way: event e can be critical from κ_1 if e is the first event to occur in the remaining of a run (by definition of $\mathcal{P}_{crit}$). Then, consider the next event f to occur in e°. From Equation (16), e can be critical only if f can be critical. Then, f is a label of an arc in A_F. Let (κ^i, κ'^i) be

this arc. The probability to effectively reach that arc is $\mathcal{P}(\kappa_2, \kappa^i)$ and the path that has been followed between configurations κ_2 and κ^i does not matter: the events occurring are concurrent to f ($f \in e^\circ$ and $\forall e' \in \kappa^i - \kappa_2$, $e' \notin {}^\circ f$) and they are the same for every path, so their order does not matter. For the rest of the formula, one has to remark that f being critical from κ^i is independent of what happens before conditionally to the occurrence of the state κ^i.

Then, the probability for an event e to be critical is

$$\mathbb{P}_{crit}(e) = \sum_{\kappa \subseteq \kappa \cup \{e\} \in \mathcal{C}(ON)} \mathcal{P}(\kappa)\mathcal{P}_{crit}(\kappa, e).$$

5 Criticality of a component in a Workflow

The above discussion covers the criticality of events in an occurrence net. However, it is of much greater relevance in practice to ask whether a given system component is critical for the performance of a compound system, in particular for systems and services that are to be frequently used. The knowledge of criticality in a complex system allows *e.g.* to allocate resources - maintenance, renewal, replacement by newer but costly equipment, etc - where they yield best global results: if only a limited budget for such interventions is available, one should strive to use it as much as possible on improvement of the performance in bottlenecks of the system.

Clearly, the above discussions on criticality in occurrence nets can serve as preparations for the system analysis here, in the sense that one wishes to *lift* statements on an occurrence of t being critical for an occurrence of t' in the unfolding $\mathcal{U}_\mathcal{N}$, to the net $\mathcal{N}$ itself and to saying that t is critical for t'. However, this is not very meaningful for general nets since the occurrences of t and t' may be only loosely coupled. We can, however, give a precise meaning to transition criticality in a particular class of Petri nets, called *workflow nets*.

The following definitions are based on [1].

Definition 6 (WF-Net). *A net $W_* = (P_*, T_*, F_*)$ is a* WF-net[2] *if and only if:*

1. *W has two special places,* source place *i and* sink place *o, such that ${}^\bullet i = o^\bullet = \emptyset$.*
2. *If we add a transition $\mathbf{t}$ to T_* that connects place o with i,* i.e. *${}^\bullet\mathbf{t} = \{o\}$ and $\mathbf{t}^\bullet = \{i\}$, then the resulting net $W = (P, T, F)$ - the* looped version *of W_* - is strongly connected.*

W is then called a looped WF-net*, and $\mathbf{t}$ is called the* loop transition *of W. WF-net W is* sound *iff*

1. *$m_0 = \{i\}$;*
2. *m_0 is a home marking,* i.e. *from every reachable marking of $\mathcal{N}$, m_0 is reachable;*
3. *$\mathcal{N}$ has no dead transitions,* i.e. *for every $t \in T$ there is a reachable marking m such that $m[t\rangle$.*

It is known that WF-net W is sound iff $\mathcal{N}$ is live and bounded [2]. Let us call any Petri net $\mathcal{N} = (W, m_0)$ for which W is a looped and sound WF-net a *WF Petri net*, or *WFPN*.

Consider the net in Figure 1. We see intuitively that, between two consecutive occurrences of the loop transition, transition N will be critical for the entire workflow every time it actually occurs, and transition X, Y and C can each be critical if N does not occur. More precisely, in that case both X and Y will occur, X exactly once, Y possibly several times; in fact, Y occurs exactly one more time than C does, before leaving the loop.

We will make this more precise now. To start, note that the successive occurrences of the loop transition provide a natural regeneration point for the stochastic behaviour of the net. The loop transition also marks the end of one execution of the workflow and the passage to the next execution. We will thus consider the criticality problem with a focus on the loop transition: at each new occurrence of it, look back to the period since the last occurrence, and ask which of the other transitions have, this time around, been critical for the total time spent.

The dynamics of WFPNs features a sequence of *rounds* separated by the successive occurrences $\mathbf{e}_1, \mathbf{e}_2, \ldots$ of $\mathbf{t}$. Formally, for any event e in the unfolding (ON, π) of $\mathcal{N}$, define the *round number* of e by $round(e) \triangleq |\pi^{-1}(\{\mathbf{t}\}) \cap [e]|$. Call $\mathbf{t}_n(\omega)$ the nth occurrence of $\mathbf{t}$ under ω; that is, one has $round(e) < n$ for all $e \prec \mathbf{t}$ and $round(e') \geq n$ for all $\mathbf{t} \preceq e'$. We will consider the following problem:

(P) Given a sound WFPN $\mathcal{N}$ and a transition $x \neq \mathbf{t}$ of $\mathcal{N}$, what is the probability $\mathbb{P}_{crit}(x, n)$ that the occurrence (if any) in round n of x is critical for $\mathbf{t}_n$?

[2] WorkFlow net

Observe that the loop transition $\mathbf{t}$ synchronises the flow at the end of each round, hence it is critical in every round. Due to this synchronisation, if x_n, the n^{th} occurrence of a transition x is critical for a round, then x_n remains critical for all successive rounds of the looped WF-Net.

As mentioned above, the synchronization at the end of a workflow round and in the firing of $\mathbf{t}$ induces a renewal of the underlying Markov processes. In particular, $\mathbb{P}_{crit}(x,n) = \mathbb{P}_{crit}(x,1)$ for any round n. We will therefore discard the round index n and represent the previous terms by $\mathbb{P}_{crit}(x)$, which denotes the probability that a transition x of $\mathcal{N}$ is critical for a round. The problem **(P)** can thus be restated as:

> Given a sound WFPN $\mathcal{N}$ and a transition $x \neq \mathbf{t}$ of $\mathcal{N}$, what is the probability $\mathbb{P}_{crit}(x)$ that the occurrence (if any) of x is critical in an execution round of $\mathcal{N}$?

Solving this problem for the example of Figure 1 and its unfolding in Figure 2, we obtain the following results:

- Transition N is critical in a round iff $\delta_N < \delta_S$, hence $\mathbb{P}_{crit}(N) = \frac{\lambda_N}{\lambda_N+\lambda_S}$.
- If $\delta_N > \delta_S$, both S and D are critical in that round, hence $\mathbb{P}_{crit}(S) = \mathbb{P}_{crit}(D) = \frac{\lambda_S}{\lambda_N+\lambda_S}$.
- For criticality of X, Y and C, the number of firings of C in a round is central. Denote by Num_C, this number of firings of C in a given round. We note that *(i)* Y fires $Num_C + 1$ times in this round, and *(ii)* X is critical in this round iff it fires after the last firing of Y, otherwise Y is critical in this round. For convenience, for transitions P and Q, let $\mathbb{P}_{P/Q} = \frac{\lambda_P}{\lambda_P+\lambda_Q}$, the probability of the delay of P being lesser than that of Q. We then have for the probability of X being critical in a round:

$$\begin{aligned}\mathbb{P}_{crit}(X) &= \mathbb{P}_{crit}(S) \cdot \sum_{i=0}^{\infty} \left[\mathbb{P}_{Y/X} \cdot \mathbb{P}_{C/X}\right]^i \cdot \mathbb{P}_{Y/X} \cdot \mathbb{P}_{X/C} \cdot \mathbb{P}_{D/C} \\ &= \mathbb{P}_{crit}(S) \cdot \mathbb{P}_{Y/X} \cdot \mathbb{P}_{X/C} \cdot \mathbb{P}_{D/C} \cdot \frac{1}{1 - \mathbb{P}_{Y/X} \cdot \mathbb{P}_{C/X}}\end{aligned}$$

  Since Y is critical in a round whenever X is not critical, we have

$$\mathbb{P}_{crit}(Y) = \mathbb{P}_{crit}(S) - \mathbb{P}_{crit}(X).$$

  Finally, C is critical in a round whenever Y is critical, except for the case when $Num_C = 0$, when C does not occur. We thus have,

$$\mathbb{P}_{crit}(C) = \mathbb{P}_{crit}(Y) - \mathbb{P}_{crit}(S) \cdot \mathbb{P}_{X/Y} \cdot \mathbb{P}_{D/C}.$$

In practice, it will be acceptable for X to be critical but not for transitions that may have to be iterated a large number of times, such as Y and C. Therefore, one will strive to increase $\mathbb{P}_{Y/X}$ to keep $\mathbb{P}_{crit}(X)$ large.

6 Conclusion and Outlook

We have established several properties of distributed Markovian systems allowing to exhibit which are the *critical* events of a non-deterministic process, and studied how to lift this analysis to workflow nets. Note that we have used a timed Markovian model in computations, whose execution traces are linearly ordered sequences. One might therefore think that we could have dropped the use of partial orders entirely and have simply used interleaved semantics throughout. However, only the causal semantics provided by unfoldings allows to retrieve the dependencies which are crucial in finding critical events : the fact that event e *occurs before* event e' in itself does not imply that e is critical for e', since the ordering of the two events may result merely from the contingent delay values. In that case, both events evolve independently of one another, and modifications in the component corresponding to e would have no impact on e'. Criticality implies *causal* ordering, hence in order to analyze criticality, the investigation of partial order unfoldings cannot be avoided. Identification and prediction of likely bottlenecks in composite processes allows to anticipate possible performance deterioration. Conversely, once the bottlenecks of an intended composite application are known, resource allocation can be optimized so that attention is focused on latency-critical components by reducing the critical local latencies. More delicate analyses, such as concerning *monotonicity* (see [7]) and *robustness* of global performance with respect to local performances, are under way or part of future work.

References

[1] W.M.P. van der Aalst. The application of Petri Nets to Workflow Management. *Journal of Circuits, Systems and Computers* **8**(1):21–66, 1998.

[2] W.M.P. van der Aalst. Verification of Workflow nets. *Proc. ICATPN 1997*, LNCS **1248**:407–426, Springer Verlag.

[3] S. Abbes and A. Benveniste. True-concurrency probabilistic models: Markov nets and a law of large numbers. *Theor. Comput. Sci.* **390**(2-3), pp. 129–170, 2008. http://dx.doi.org/10.1016/j.tcs.2007.09.018.

[4] S. Abbes and A. Benveniste. Probabilistic models for true-concurrency: branching cells and distributed probabilities for event structures. *Information and Computation* **204** (2), p. 231-274. February 2006.

[5] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalized Stochastic Petri Nets.* Parallel Computing Series, Wiley, 1995.

[6] A. Benveniste, E. Fabre, and S. Haar. Markov Nets: Probabilistic Models for distributed and concurrent systems. *IEEE Trans. Aut. Control* **48**(11):1936–1950, November 2003.

[7] A. Bouillar, S. Rosario, A. Benveniste, S. Haar. Monotonicity in Service Orchestrations. *Petri Nets 2009*, http://petrinets2009.lip6.fr/.

[8] J. Engelfriet. *Branching Processes of Petri Nets.* Acta Informatica **28**:575–591, 1991.

[9] J. Esparza, S. Römer, and W. Vogler. An improvement of McMillan's unfolding algorithm. *Formal Methods in System Design* **20**(3):285-310, 2002.

[10] P. Haas. Stochastic Petri Nets. Modelling, Stability, Simulation. Springer Series in Operations Research, Berlin 2002.

[11] J.-P. Katoen, C. Baier and D. Latella. Metric semantics for true concurrent real time. *Theoretical Computer Science* **254**, pp. 501-542, 2001.

[12] J. Mairesse and S. Gaubert. Modeling and Analysis of Timed Petri Nets using Heaps of Pieces. *IEEE Trans. Autom. Control* **44**(4):683–697, 1999.

[13] K. McMillan. Using Unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. *4th Workshop on Computer Aided Verification* 164–174, 1992.

[14] M. Nielsen, G. Plotkin G. Winskel. Petri nets, event structures, and domains, Part I. *TCS* **13**:85–108, 1981.

[15] T.C. Ruys, R. Langerak, J.-P. Katoen, D. Latella, M. Massink. First Passage Time Analysis of Stochastic Process Algebra Using Partial Orders. *TACAS 2001*, *LNCS* **2031**:220-235, Springer-Verlag, 2001.

[16] S. Rosario, A. Benveniste, S. Haar, C. Jard. Probabilistic QoS and soft contracts for transaction based Web services. Proceedings *ICWS* 2007: 126–133, 2007.

[17] S. Rosario, D. Kitchin, A. Benveniste, W. Cook, S. Haar and C. Jard. Event Structure Semantics of Orc. In: WS-FM, 2007; long version as INRIA Research Report Nr 6221.

[18] H.G. Tucker. An Introduction to probability and mathematical statistics. *Academix Press*, 1962.

[19] D. Varacca, H. Völzer and G. Winskel. Probabilistic event structures and domains. *Theor. Comput. Sci.* **358**(2-3): 173–199, 2006.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399