A New Reduction Method for the Analysis of Large Workflow Models

Loucif Zerguini & Kees Max van Hee

{l.zerguini, k.m.v.hee}@tue.nl

Abstract: This paper presents a new net-reduction methodology to facilitate the analysis of large workflow models. We propose an enhanced algorithm based on reducible subnet identification which preserves both soundness and completion time distribution. Moreover we outline an approach to model the dynamic behavior of business processes by exploiting the power of a class of non-Markovian stochastic Petri net models.

1 Introduction

Recently, business process redesign (BPR) has been receiving a lot of attention. In addition to the literature on the management aspects of BPR [AH02],[HC93], several modeling frameworks and software tools to support the redesign trajectory have been presented. Typically, they offer little support for the analytical verification of desirable model properties.

This can be addressed by modeling business processes using Petri nets [Mur89]. The suitability of Petri nets for this field is discussed in [Aal98]. However, few results in workflow systems take into account the notion of time [AHR00],[Zer02] and offer analysis as a means to evaluate the completion time distribution. We need to compute the distribution of completion time. Using the computed completion time distribution we can check whether a design satisfies user requirements. For example, the probability can be computed of violating a certain deadline constraint. It should be clear that such information cannot be obtained if the analysis only yields the average completion time.

Two methods have been proposed for developing a framework to the completion time distribution of workflows. In [AHR00], a compositional approach based on the method of Fast Fourier Transform for analyzing discrete stochastic Petri nets was developed. In [Zer02] a reductional method based on Petri net transformations and closure properties of phase type distribution was proposed. These two methods seem to be very suitable for certain classes of workflows. As an alternative version of these two methods, we extend the repertoire of available analyzable workflows to take into account more complex interaction structures.

Much research has been reported about reduction techniques for general Petri nets. In particular, stepwise refinement and abstraction methods [SM83],[Val79] have been developed

for Petri nets, where the reduction technique can be used as a "divide-and-conquer" approach for the analysis of liveness, boundedness, resource requirements, etc., for complex Petri nets. Our new reduction rule is similar in spirit to this, in that it preserves at the same time correctness and stochastic behavior in workflow modeling. Moreover a general methodology for the modeling and evaluation of workflows, based on a Stochastic Workflow Nets (SWF-net) is presented. The SWF-net is a stochastic counterpart of a class of untimed Petri nets called "Workflow nets" (WF-net) introduced by Van der Aalst in [Aal98] for workflow analysis.

The major contributions of our work are the following:
• the definition of the SWF-net modeling framework that accommodates in quite a natural way the evolution of the workflow instances.
• the design of an efficient algorithm based on successive reductions of a Petri net to determine the completion time distribution of large workflow models. Since analytic-numeric methods such as the ones proposed here are rather fast (as compared to simulation), they can be repeated to assess the effect of changes in system design or even to obtain an optimal design.

The paper is organized as follows: Section 2 defines Petri nets, workflow nets, and verification criteria. Section 3 introduces the SWF-net framework and apply it to an example of workflow that serves as a case study throughout the paper. In Section 4 the concept of reducibility is given and the reduction algorithm is developed. The general analytical solution of the SWF-net models of Workflows is then discussed in Section 5 and an application is shown in Section 6. Finally, conclusions are given in Section 7.

2 Workflow Nets

The classical Petri net is a directed bipartite graph with two nodes types called *places* and *transitions*. The nodes are connected via directed arcs. Connections between two nodes of the same type are not allowed. Places are represented by circles and transitions by rectangles.

Definition 1 . *A Petri net is a triple (P, T, Ξ):*
- P is a finite set of places,
- T is a finite set of transitions ($P \cap T = \emptyset$),
- $\Xi \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation)

A place p is called an *input place* of a transition t iff there exists a directed arc from p to t. Place p is called an *output place* of transition t iff there exists a directed arc from t to p. A set of input (resp. output) transitions of a place $p \in P$ is denoted by ${}^\bullet p$ (resp. $p^\bullet$) and the set of input (resp.output) places of a transition $t \in T$ is denoted by ${}^\bullet t$ (resp. $t^\bullet$); for $X \subseteq (P \cup T)$, ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$.

At any time a place contains zero or more tokens, drawn as black dots. The *state*, often referred to as marking, is the distribution of tokens over places. We will represent a state as follows: $p_1 + 2p_2 + p_3$ is the state with one token in place p_1, two tokens in p_2, one

token in p_3. A Petri net *PN* and its initial marking are denoted by (*PN* , M_0).

Transitions change the state of the net according to the following firing rule: (1) A transition t is said to be *enabled iff* each input place p of t contains at least one token.
(2) An enabled transition may *fire*. If transition t fires, then t *consumes* one token from each input place p of t and *produces* one token for each output place p of t.

The firing rule specifies how a Petri net can move from one state to the next one. A firing sequence $\sigma = t_1 t_2 ... t_n$ is enabled if, starting from the initial marking, it is possible to subsequently fire $t_1, t_2, ...t_n$. A marking M is reachable from the initial marking if there exists a enabled firing sequence resulting in M. The reachability set $R(M_0)$ is the set of all markings reachable from the initial marking M_0. The reachability graph associated with a reachability set can be constructed as follows: Represent each state by a vertex and place a directed edge from vertex V_i to vertex V_j if the state V_j can result from the firing of some transition enabled in V_i

Definition 2 *(Live). A Petri net (PN , M_0) is live iff, for every reachable state M' and every transition t there is a state M" reachable from M' which enables t.*

Definition 3 *(Bounded, Safe). A Petri net (PN , M_0) is bounded iff for each place p there is a naturel number n such that for every reachable state the number of tokens in p is less than n. The net is safe iff for each place p and for every reachable state the maximum number of tokens in p does not exceed 1.*

Definition 4 *(Path) Let PN be a Petri net. A path C from a node n_1 to a node n_k is a sequence $< n_1, n_2, ..., n_k >$ such that $< n_i, n_{i+1} > \in \Xi$ for $1 \leq i \leq k-1$.*

Definition 5 *(Strongly connected). A Petri net is strongly connected iff, for every pair of nodes (i.e., places and transitions) x and y, there is a path leading from x to y.*

A Petri net which models the control-flow dimension of a workflow, is called a *Workflow net* (WF-net). It should be noted that a WF-net specifies the dynamic behavior of a single case in isolation.

Definition 6 *(WF-net). A Petri net PN = (P, T , Ξ) is a WF-net (Workflow net) if and only if: (i) There is one source place $i \in P$ such that $\bullet i = \emptyset$*
(ii) There is one sink place $o \in P$ such that $o\bullet = \emptyset$
(iii) Every node $x \in P \cup T$ is on a path from i to o.

A WF-net has one input place (*i*) and one output place (*o*) because any case handled by the procedure represented by the WF-net is created when it enters the workflow management system and is deleted once it is completely handled by the workflow management system, i.e., the WF-net specifies the life-cycle of a case. The third requirement in *Definition 6* has been added to avoid dangling tasks and/or conditions, i.e., tasks and conditions which do not contribute to the processing of cases.

The three requirements stated in *Definition 6* can be verified statically, i.e., they only relate to the structure of the Petri net. However, there is another requirement which should be satisfied:

For any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in place o and all other places are empty.

Moreover, there should be no dead tasks, i.e., it should be possible to execute an arbitrary task by following the appropriate route through the WF-net. These two additional requirements correspond to the so-called soundness property.

Definition 7 *(Sound). A procedure modeled by a WF-net PN = (P , T , Ξ) is sound if and only if:*

(i) For every state M reachable from state i, there exists a firing sequence leading from state M to state o.

(ii) State o is the only state reachable from state i with at least one token in place o.

(iii) There are no dead transitions in (PN, i).

In [Aal98] it is shown that soundness corresponds to liveness and boundedness . To link soundness to liveness and boundedness, an extended net $\overline{PN} = (\overline{P}, \overline{T}, \overline{\Xi})$ was defined. $\overline{PN}$ is the Petri net obtained by adding an extra transition t^* which connect o to i. The extended Petri net $\overline{PN} = (\overline{P}, \overline{T}, \overline{\Xi})$ is defined as follows: $\overline{P}$ = P, $\overline{T}$ = $T \cup \{t^*\}$, and $\overline{\Xi}$ = $\Xi \cup \{$ (*o, t**), (*t*, i*)$\}$. In the remainder we will call such an extended net the short- circuited net of *PN*. Note that $\overline{PN}$ is strongly connected.

Theorem 1 *. A WF-net PN is sound if and only if $(\overline{PN}, i)$ is live and bounded.*

Proof 1 *See [Aal98].*

We refer the reader to [AH02] for more details and explanations about all these notions.

3 Stochastic Modeling of Workflows

Modeling the complex dynamic behavior of workflows requires highly representative and expressive tools. To attack the problem, we resort to the SWF-net models, which combine representation power with modeling features that significantly improve their expressiveness and allow for a compact modeling of workflow aspects. SWF-net models are defined as follows:

Definition 8 *A SWF-net is a tuple (P, T, Ξ, W, F)*

1. (P, T, Ξ) *is a sound WF-net.*
2. $W : T \longrightarrow I\!R^+ - \{0\}$ *(weight function).*

3. *F: $T \longrightarrow (\mathbb{R}^+ \longrightarrow [0,1])$ (delay function) such that for $t \in T$, F_t is a probability distribution function over $\mathbb{R}^+$*

The weight function W is added to each transition to resolve conflicts, we denote the weight $W(t)$ of a transition $t \in T$ with w_t. The delay function F will be used to sample transition delays that represent the firing time of actual transition execution. A transition $t \in T$ is called $timed$ if $F_t(0) < 1$. A transition $t \in T$ is called $immediate$ if $F_t(0) = 1$. Pictorially, square boxes represent stochastic timed transitions and thin bars represent immediate transitions. An enabled transition can fire after time delay sampled from a delay function F. The probabilistic study of a WF-net is made by examining the marking process M(t), obtained by constructing a reachability graph for the net. Markings in which at least one immediate transition is enabled are called vanishing markings. On the other hand, markings in which only timed transitions are enabled are called tangible markings. In the case of tangible marking, any enabled transition can fire next, conflicts are solved on basis of the weights of the enabled transitions, we assume implicitly $preselection\ policy$, this in contrast to the race semantics. The preselection policy is reasonable in the context of workflow management, since tasks are often performed by human resources whose work typically cannot (or will not) be cancelled upon completion of other tasks. (see [MBBC89] for a comprehensive treatment of the preemption policies modeling). However, for a vanishing marking, only the enabled immediate transitions are allowed to fire and conflicts are solved on the basis of weights. As the time spent by the process in the latter is zero, since they enable at least one immediate transition, they can be eliminated from a study of the marking process as they have no effect on the state probabilities. We thus obtain a reduced reachability graph in which the vanishing states are no longer explicitly present. A SWF-net captures the dynamic behavior of a single case in isolation within a workflow. The stochastic process that it induces expresses the way that a specific case or workflow instance is handled. This is the reason that the net is uncolored: all tokens refer to the same case.

As it is well known, the addition of stochastic times with infinite support distribution to Petri nets does not change the logical behavior of the model, and therefore all the well known results obtained for WF-nets still apply to SWF-nets. All the considered SWF-nets are supposed to be sound. In this paper we focus on the time between the start and the end of the processing of a single case. Informally stated, this is the completion time. For a meaningful semantics of the completion time, we will insist on any SWF-net to be sound. In the sequel a SWF-net will be simply denoted by (P, T, Ξ).

In the following, we present the guidelines for the SWF-net modeling of workflows through an example of a workflow application which includes most of the typical features of the workflows we are considering here.

3.1 Example

The dynamic behavior of a workflow can be described as follows. The tasks and the precedence relations among them that constitute a business process are described by a WF-net, in which each task is a transition $t \in T$. We distinguish transitions $T_i \in T$, which stand for tasks, and transitions $t_i \in T$ that do not have a counterpart in the workflow system. Transition $T_i \in T$ have assigned enabling delays with a general distribution $F_{T_i}(t)$ with support $(0, \infty)$, whereas transitions $t_i \in T$ do not consume time for their enabling (timeless), but can have assigned a probability $p(t_i)$ (calculated on basis of weights) to model random selection of tokens for firing (random switches).

Fig. 1 shows a workflow model of one business process (Order Processing). The execution of the task (T_1) spawns two tasks (T_2, T_3) that can be processed independently. As an example let (T_1) represent the registration of the customer order that invokes two subsequent tasks. After registering the order there is a check to see if the desired goods are available (T_2). In parallel a bill is sent to the customer (T_3). If the goods are available, they are shipped to the customer (T_5). If the goods are not available, there are two possibilities. If the missing products have already been reordered, no replenishment is needed, if the goods have not been reordered yet, replenishment is needed (T_4). The occurrence of each of the two previous possibilities is followed by the execution of the task update (T_8). The immediate transitions (timeless) t_1, t_2 and t_3 have been added to model the three possible outcomes of executing task T_2. After sending the bill, two things can happen: either the payment is received (T_6) or a reminder needs to be sent (T_7). This process is repeated each period‘ (i.e., a customer can receive many bills for the same order) and it is assumed that eventually the customer will pay. Similarly, the goods are eventually shipped. Therefore, eventually, the customer pays and the goods are shipped. After this the order is archived (T_9).

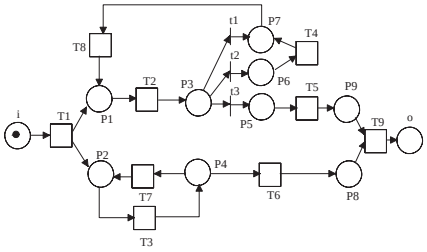

Figure 1: Order Processing

The SWF-net shown in Fig.1 clearly illustrates that we focus on the control-flow dimension. We abstract from resources, applications, and technical platforms. Each task has an unlimited number of resources. The availability of resources is an important factor in the completion time of an actual workflow; as the lack of resources may cause queueing times. We focus on the intrinsic quality of the workflow by not regarding resources. According to [AH02], the usual way to design a business process is to design its structure first and

then to allocate resources to it. Therefore, the potential completion time characteristic is used during the design phase, after which appropriate resources are assigned to obtain a completion time behavior that is satisfactory.

4 Reduction Rules

SWF-net reduction rule is used to transform SWF-nets into smaller SWF-nets which preserve soundness and completion time distribution. In this section we introduce a notion of reducible subnet which is pertinent for tackling the problem of complexity.

4.1 Reduction rule 1:

Reducible subnet (RSN): Let $G = (P, T, \Xi)$ be a SWF-net and $X \subseteq P \cup T$ be a set of nodes, then $N = (P \cap X, T \cap X, \Xi \cap (X \times X))$ is a RSN of G, *iff* $\exists p, q \in P$ such that,

1. $|T \cap X| \geq 2$ (large enough, at least two transitions)

2. $p^\bullet \subseteq X \wedge q^\bullet \cap X = \emptyset$ (p is the unique input place and q is the unique output place)

3. $\forall$ x $\in X$- $\{p, q\}$: ${}^\bullet x \cup x^\bullet \subseteq X$ (nodes in X - $\{p, q\}$ are disconnected from all other nodes outside of X)

4. N is safe

$Rule.$ A *RSN* N may be replaced by a single transition t_f with input set ${}^\bullet t_f = \{p\}$, with output set $t_f^\bullet = \{q\}$ and as its firing time distribution the completion time distribution of the considered RSN (see Fig.2). (In section 5 an analytic method is proposed to show how to estimate the completion time distribution.). The reduced net derived from G by applying the *RSN* reduction rule w.r.t. set X of nodes of G is:
$G' = (\{P - X\} \cup \{p, q\}, \{T - X\} \cup \{t_f\}, \{\Xi - (X \times X)\} \cup \{(p, t_f), (t_f, q)\})$

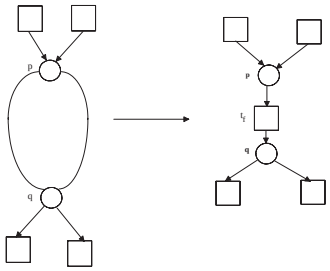

Figure 2: RSN reduction rule

We illustrate the necessity of the fourth condition in Fig.3. The shaded net meets the first three conditions of the RSN reduction rule, but violates the fourth condition, although the global net is sound.

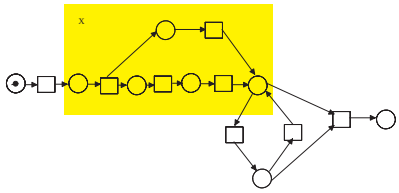

Figure 3: Illustration of necessity of fourth condition of the RSN reduction rule

Theorem 2 *The transformation rule RSN preserves soundness, i.e. if a WF-net is sound, then the WF-net transformed by the RSN rule is also sound.*

Proof: Let *PN* be a sound WF-net. Let *PN'* be a net which is obtained by applying the transformation rule *RSN1* on *PN*. We have to prove that *PN'* is a sound WF-net. It is easy to see that *PN'* is a WF-net. There is still one input and one output place and the $\overline{PN'}$ is strongly connected. ($\overline{PN'}$ is the short cutted Petri net *PN'* with an extra transition t^* which connect place o and place i.) By theorem 1 we know that to prove soundness, it suffices to show $(\overline{PN'}, i)$ is live and bounded. $(\overline{PN}, i)$ is live and bounded. Therefore, we have to prove that the *RSN* reduction rule preserves liveness and boundedness: this follows from results in[Val 79] and related work.

Theorem 3 *The transformation rule RSN preserves completion time distribution, i.e., a completion time distribution of a SWF-net equals that of the SWF-net transformed by the RSN rule.*

Proof: Let SWF-net G' be derived from SWF-net G by applying the *RSN* reduction rule w.r.t. some set X of nodes of G, and let $\{t_f\}$ be the fused transitions of G'.

Let's consider our *RSN* (with the unique input place p and a unique sink place q) in isolation. According to lemma 1, every token which enters the *RSN* from a place p will eventually reach a place q and the moment there is a token in place q all other places of the *RSN* are empty since the short-circuited net (that is, the net obtained after adding a transition with the output place q as it is only input and the input place p as it is only output) is live and safe. Therefore, the completion time distribution of the isolated *RSN* equals the firing time distribution of $\{t_f\}$ which obviously imply the preservation of the global completion time distribution of both nets G and G'.

4.2 Reduction rule 2:

Reducible subnet (RSN): Let $G = (P, T, \Xi)$ be a SWF-net and $X \subseteq P \cup T$ be a set of nodes, then $N = (P \cap X, T \cap X, \Xi \cap (X \times X))$ is a RSN of G, *iff* $\exists p, q \in T$ such that,

1. $|T \cap X| \geq 2$ (large enough, at least two transitions)

2. $p^\bullet \subseteq X \wedge q^\bullet \cap X = \emptyset$ (p is the unique input transition and q is the unique output transition)

3. $\forall$ x $\in X$- $\{p, q\}$: ${}^\bullet x \cup x^\bullet \subseteq X$ (nodes in X - $\{p, q\}$ are disconnected from all other nodes outside of X)

4. N is safe.

$Rule.$ A *RSN* N may be replaced by a single transition t_f with input set ${}^\bullet p$, with output set $q^\bullet$ and as its firing time distribution the completion distribution of the considered RSN (see Fig.4).

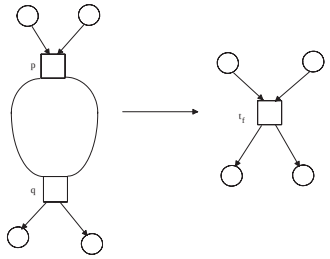

Figure 4: RSN reduction rule

Theorem 2 and theorem 3 are still valid for a transition bordered RSN.

Both types of reducible subnets can be identified on an efficient manner. We have a detection algorithm that is quadratic on the size of the input net.
In the following we present a general method development technique which entails isolating the RSN from the original model, analyzing the isolated subsystem and replacing the subsystem by an equivalent atomic net (timed transition with one input place and one output place) whose completion time distribution equals that of the isolated RSN.

5 The Analytical Technique

This approach is based on the general solution method for Semi Markov Process (*SMP*) [Cin75], which applies for the solution of any SWF-net model. By taking into account the special structure of the workflow models, we greatly enhance the efficiency of the completion time computation.

Definition 9 *A SWF-net is an Semi-Markov stochastic Petri nets (SMSPN) if its underlying marking process is a SMP.*

Definition 9 does not give an immediate description of the modeling features allowed for *SMSPN* . Since Markov processes are special cases of *SMP*, the SPN and GSPN classes of

Petri nets belong in fact to *SMSPN* class. However, to guarantee that the marking process is indeed an *SMP*, some restrictions must be imposed on the concurrent enabling of the transitions.

5.1 *SMP* Structure of the Marking Process

Because of their generality, it is not easy to check whether a given SWF-net model is a *SMSPN* or not. In general, the marking process must be inspected, but this can be a very costly check for complex models. A way to build SWF-net that are *SMSPN* analytically solvable models is to prescribe conditions on the net structure in such a way that a specific structure is definitely found in the underlying marking processes. To this aim, a sufficient constraint to be imposed on the net structure, is the following one:

Condition: The firing time of all concurrent transitions must be exponentially distributed. In this case, the Markov property holds (at least) each time a general transition fires or it is disabled. This constraint is the one usually adopted [DTGN84], for it provides an easily checked condition to test for the existence of SMP structure in the underlying marking process. This condition is at least satisfied for a SWF-net state machine (SWF-net where each transition has exactly one input place and one output place).

In the following, let's denote the Laplace Stieltjes transform (LST) of a function $F(t)$ by $F^{\sim}(s) = \int_0^{\infty} e^{-st} dF(t)$ ([Tri02]).

Theorem 4 *Let $G = (P, T, \Xi)$ be a SWF-net which is a SMSPN , and $Q(t)$ be a semi-Markov kernel over the reachability set $R(M_0)$. The LST of case completion time is given by:*

$$F^{\sim}(s) = e\ (I - Q^{\sim}(s))^{-1}\ e'$$

where $e = (1, 0, ..., 0)$, $e' = (0, ...0, 1)^T$ and I is the identity matrix.

proof : Let $Q = \{Q_{(i,j)}(t) : i, j \in R(M_0), t \in I\!R^+\}$ be a semi-Markov kernel over the reachability set $R(M_0)$ {finite set of all markings that can be generated from the initial marking M_0 } and let $F_{(i,j)}(t)$ be the distribution of the first passage time from state i to state j. It follows from [Cin75] that: $F^{\sim}_{(j,j)}(s) = 1 - \frac{1}{R^{\sim}_{(j,j)}(s)}$ and $F^{\sim}_{(i,j)}(s) = \frac{R^{\sim}_{(i,j)}(s)}{R^{\sim}_{(j,j)}(s)}$, where $R^{\sim}(s) = (I - Q^{\sim}(s))^{-1}$, $R^{\sim}(s) = [R^{\sim}(s)]_{i,j}$, $Q^{\sim}(s) = [Q^{\sim}_{(i,j)}(s)]_{i,j}$ and I is the identity matrix.

The LST of the distribution of the first passage time from the root node to the sink node in the considered reachability graph, which is in fact the distribution of the completion time of a workflow instance, follows immediately.

Remarks:
1- The symbolic evaluation of $(I - Q^{\sim}(s))^{-1}$ is computationally hard. To alleviate this inconvenient, we can use an indexed equivalence over semi-Markov process as specified

and defined by [Bra02], which is based on a state aggregation of the *SMP* . This agregation is $O(n^3)$ for all states in the worst case if heavily interconnected.
2- It may happen that an input place p of the RSN $N = (P \cap X, T \cap X, \Xi \cap (X \times X))$ is not a source place, i.e, ${}^\bullet p \cap X \neq \emptyset$. To obtain its SWF-net version with the same completion time distribution, it suffices to add one place and one immediate transition adjacent to the input place of the place bordered RSN. Therfore, Theorem 4 still applies to any RSN.
3-The completion time distribution of a transition bordered RSN can be estimated by simply adding one place and one immediate transition adjacent to the input transition of the RSN and one transition and one place adjacent to the output transition of the RSN.

6 Application

Our procedure is best explained through the example of SWF-net modeling the processing of orders (Fig.1).

Step 1: Detection of *RSNs*: *RSN1* (see Fig.5)

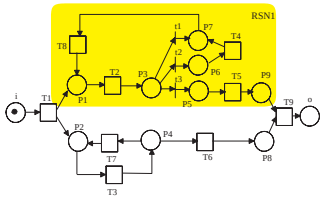

Figure 5: Detection of *RSNs*

Step 2: Analyze *RSN1* in isolation

Removing the vanishing markings leads to the semi-Markov model as shown in Fig.6.

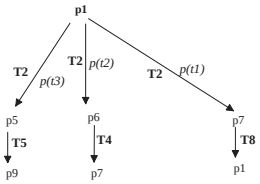

Figure 6: Reduced reachability graph of RSN1

Let's denote by $p(t_i) = \frac{w_{t_i}}{w_{t_1}+w_{t_2}+w_{t_3}}$ a probability to fire an immediate transition t_i.

The corresponding semi-Markov kernel is given by:

$$\begin{aligned} Q(p_1, p_5, t) &= p(t_3)F_{T_2}(t) \\ Q(p_1, p_6, t) &= p(t_2)F_{T_2}(t) \\ Q(p_1, p_7, t) &= p(t_1)F_{T_2}(t) \\ Q(p_5, p_9, t) &= F_{T_5}(t) \\ Q(p_6, p_7, t) &= F_{T_4}(t) \\ Q(p_7, p_1, t) &= F_{T_8}(t) \end{aligned}$$

Hence, for $s \geq 0$, we obtain

$$\begin{aligned} Q^{\sim}_{(p_1,p_5)}(s) &= p(t_3)F^{\sim}_{T_2}(s) \\ Q^{\sim}_{(p_1,p_6)}(s) &= p(t_2)F^{\sim}_{T_2}(s) \\ Q^{\sim}_{(p_1,p_7)}(s) &= p(t_1)F^{\sim}_{T_2}(s) \\ Q^{\sim}_{(p_5,p_9)}(s) &= F^{\sim}_{T_5}(s) \\ Q^{\sim}_{(p_6,p_7)}(s) &= F^{\sim}_{T_4}(s) \\ Q^{\sim}_{(p_7,p_1)}(s) &= F^{\sim}_{T_8}(s) \end{aligned}$$

Then, using (theorem 4) the LST of the completion time distribution of RSN1 is given by:

$$\psi_1(s) = \frac{p(t_3)F^{\sim}_{T_2}(s)F^{\sim}_{T_5}(s)}{1-p(t_2)F^{\sim}_{T_2}(s)F^{\sim}_{T_4}(s)F^{\sim}_{T_8}(s)-p(t_1)F^{\sim}_{T_2}(s)F^{\sim}_{T_8}(s)}$$

Step 3 Replace RSN1 by its corresponding atomic SWF-net (see Fig.7)

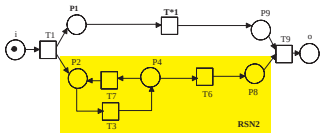

Figure 7: Reduced SWF-net

Step 4 Analyze RSN2 in isolation

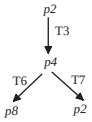

Figure 8: Reachability graph of RSN2

The probability β for a case to exit from a place P_4 is given by:

$\beta = w_{T_6}/w_{T_6} + w_{T_7}$

The corresponding semi-Markov kernel is given by:

$$\begin{array}{lcl} Q(p_2, p_4, t) & = & F_{T_3}(t) \\ Q(p_4, p_8, t) & = & \beta F_{T_6}(t) \\ Q(p_4, p_2, t) & = & (1-\beta) F_{T_7}(t) \end{array}$$

Hence, for $s \geq 0$, we obtain

$$\begin{array}{lcl} Q^{\sim}_{(p_2,p_4)}(s) & = & F^{\sim}_{T_3}(s) \\ Q^{\sim}_{(p_4,p_8)}(s) & = & \beta F^{\sim}_{T_6}(s) \\ Q^{\sim}_{(p_4,p_2)}(s) & = & (1-\beta) F^{\sim}_{T_7}(s) \end{array}$$

Using (theorem 4), the LST of the completion time distribution of RSN2 is given by:

$$\psi_2(s) = \frac{\beta F^{\sim}_{T_3}(s) F^{\sim}_{T_6}(s)}{1-(1-\beta) F^{\sim}_{T_3}(s) F^{\sim}_{T_7}(s)}$$

Step 5 Replace RSN2 by its corresponding atomic SWF-net (see Fig.9)

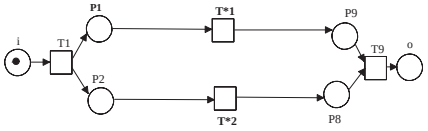

Figure 9: Reduced SWF-net

Thus, the original net with 9 tasks has been reduced to an equivalent simple net with only 4 tasks. Our reductional approach decrease greatly the computation effort to analyze the resulting net with more advanced approximative methods [AHR00],[Zer02], or numerical approaches supported by the fairly recent developments in Laplace inversion algorithms [AW92] that can provide numerically stable inversions even for discontinuous functions.

7 Conclusion

Due to the state explosion, a wide range of modeling problems concerning the evaluation of complex workflows, according to aspects like soundness and completion time distribution, are very difficult to handle if they are not decomposed into separate submodels. Our reductional approach offers a suitable solution by means of structural decomposition and aggregation. This approach is based on the identification of general reducible subnets. We have designed an efficient algorithm to perform subnet identification and aggregation, which can be readily incorporated into existing workflow modeling tools. Moreover, the analytical approach presented here can be extended to a larger class of non-Markovian stochastic Petri nets [PST98].

8 References

[Aal98]— W.M.P.van.der Aalst, The application of Petri nets to workflow management. The Journal of Circuits, Systems and Computers, vol.8, pp. 21–66, 1998.

[AH02]— W.M.P. van der Aalst and K.M. van Hee. Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge, 2002.

[AHR00]— W.M.P. van der Aalst and K.M. van Hee and H.A. Reijers. Analysis of discrete-time stochastic Petri nets. Statistica Neerlandica, vol.54, pp. 237–255, 2000.

[AW92]— J. Abate and W. Whitt, The Fourier-series method for inverting transforms of probability distributions. Queueing Systems, vol. 10,pp. 5–88, 1992.

[Bra02]— J. T. Bradley. A passage-time preserving equivalence for semi-Markov processes. Proc. Computer performance evaluation (Tools'02), London, UK, pp. 178–187, April 2002.

[Cin75]— E.Cinlar. Introduction to Stochastic Processes, North western University, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[DTG84]— J. B. Dugan, K. S. Trivedi, R. Geist, and V.F. Nicola. Extended stochastic Petri nets: applications and analysis. Proc. Performance, pp. 507–519, Paris 1984.

[HC93]— M. Hammer and J. Champy. Reengineering the Corporation. Nicolas Brealey Publishing, London 1993.

[MBBC89]— M. A, Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte and A.Cumani, The effect of execution policies on the semantics and analysis of stochastic Petri nets. IEEE Transactions on Software engineering. vol.15, pp. 832–846, 1989.

[Mur89] — T. Murata, Petri nets: properties, analysis and applications. Proc. IEEE, vol.77, pp. 541–580, April 1989.

[PST98]— A. Puliafito, M. Scarpa and K. S. Trivedi, Petri nets with k simultaneously enabled generally distributed timed transitions. Performance Evaluation, vol. 32, pp. 1–34, 1998.

[SM83] — I. Suzuki and T. Murata, A method for stepwise refinements and abstractions of Petri nets. J. comput. Syst. Sci., vol. 27,no.1, pp. 51-76,1983.

[Tri02]— K. S. Trivedi, Probability and Statistics with Reliability, Queueing and Computer Science Applications, John Wiley and Sons, INC. 2002

[Val79]— R. Valette, Analysis of Petri nets by stepwise refinements. J.comput. syst. Sci., vol.18, pp.35-46, 1979.

[Zer02]— L. Zerguini, Approximate computation of response time distribution in workflows. Proc. High Performance Computing Symposium (HPC'02), San Diego, CA, pp. 280–287, April 2002.