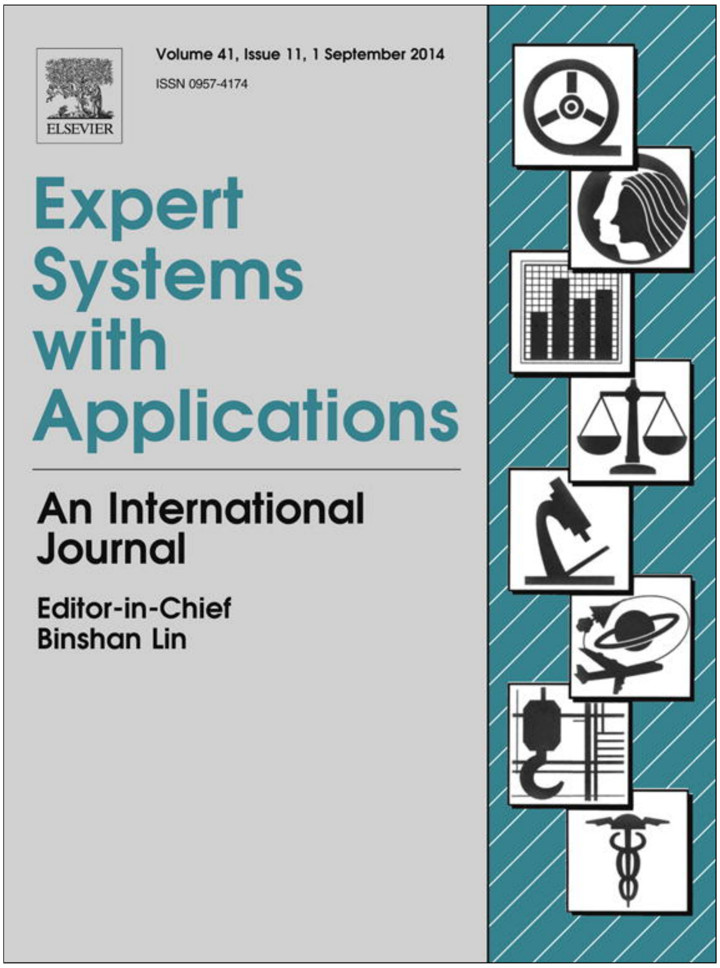
Volume 41, Issue 11, 1 September 2014
ISSN 0957-4174
ELSEVIER
Expert Systems with Applications
An International Journal
Editor-in-Chief
Binshan Lin

Expert Systems with Applications 41 (2014) 5030–5040

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Verifying soundness of business processes: A decision process Petri nets approach

Julio Clempner *

Faculty of Physics and Mathematics, National Polytechnic Institute, Edificio 9 U.P. Adolfo Lopez Mateos, Col. San Pedro Zacatenco, 07730 Mexico City, Mexico

ARTICLE INFO

Keywords:
Soundness
Workflow nets
Decision-process Petri nets
Stability
Lyapunov methods
Optimization

ABSTRACT

This paper presents a trajectory-tracking approach for verifying soundness of workflow/Petri nets represented by a decision-process Petri net. Well-formed business processes correspond to sound workflow nets. The advantage of this approach is its ability to represent the dynamic behavior of the business process. We show that the problem of finding an optimum trajectory for validation of well-formed business processes is solvable. To prove our statement we use the Lyapunov stability theory to tackle the soundness verification problem for decision-process Petri nets. As a result, applying Lyapunov theory, the well-formed verification (soundness) property is solved showing that the workflow net representation using decision process Petri nets is uniformly practically stable. It is important to note that in a complexity-theoretic sense checking the soundness property is computationally tractable, we calculate the computational complexity for solving the problem. We show the connection between workflow nets and partially ordered decision-process Petri net used for business process representation and analysis. Our computational experiment of supply chains demonstrate the viability of the modeling and solution approaches for solving computer science problems.

1. Introduction

1.1. Brief review

Companies' success depends on the ability to evolve with the market, not just respond to it. In response to the competitive pressures applied by the customer demands and the constant changes on the conditions of the environment, many companies are re-thinking the way they do business (Hammer, 1990). The ambient turbulence has created a need for dynamic business processes and companies are looking for models that can evolve and adapt efficiently business processes to the changing conditions and the changing business strategies. As a consequence, research interest in business process modeling has increased dramatically over the past decades.

Organizations needs very complicated configuration and arrangements, it has been claimed that carefully developed models are necessary for describing, analyzing and/or enacting the underlying business processes (van Hee, Sidorova, & van der Werf, 2013). The most critical point in the development of a business process depends largely on the ability to choose a conceptual model to represent the problem domain in a coherent and natural fashion and, ensure validation ability (van der Aalst, 2013). Validation of well-formed business process models is very important in the context of business process re-engineering (BPR), because the task of BPR is to evaluate the current processes with the goal of radically revising them, in order to accommodate their improvement to new organizational needs or goals. Formal models that capture and organize knowledge about a business environment can facilitate solutions to this problem (Fahland & van der Aalst, 2012). Petri nets are used for business process representation, taking advantage of the well-known properties of this approach, namely, formal semantic, graphical display and wide acceptance by practitioners of workflow nets (Clempner & Retchkiman, 2005; Chen, Ha, & Zhang, 2013; Fahland & van der Aalst, 2012; van Hee et al., 2013; Li & Iijima, 2007; van der Aalst, 2011, 2013).

Loosely speaking, a workflow net is a Petri net with an initial place and a distinguished final place called sink. Well-formed business processes correspond to sound workflow nets (van der Aalst, 2007). Petri nets have been extensively studied since the mid nineties, as an abstraction of the workflow, to check the soundness property (van der Aalst, 1998, 2007, 2011; Bashkin & Lomazova, 2013; Barkaoui & Petrucci, 1998; Barkaoui & Ayed, 2011; Basu & Blanning, 2000; Basu & Kumar, 2002; Bi & Zhao, 2004; Clempner & Retchkiman, 2005; Clempner, 2014; Dehnert & Rittgen, 2001;

* Tel.: +52 5548775074.
E-mail address: julio@clempner.name

http://dx.doi.org/10.1016/j.eswa.2014.03.005
0957-4174/

van Dongen, van der Aalst, & Verbeek, 2005; Fu, Bultan, & Su, 2002, 2004; van Hee, Serebrenik, Sidorova, & Voorhoeve, 2005, 2004; Karamanolis, Giannakopoulou, Magee, & Wheater, 2000; Kindler, Martens, & Reisig, 2000; Liu, Du, & Yan, 2012; Liu, 2013; Lohmann, Massuthe, Stahl, & Weinberg, 2006; Martens, 2005a, 2005b; Mendling, Neumann, & van der Aalst, 2007; Sadiq & Orlowska, 2000; Salimifard & Wright, 2001; Vanhatalo, Völzer, & Leymann, 2007; Verbeek, Basten, & van der Aalst, 2001; Wombacher, 2006; Wynn, Edmond, van der Aalst, & ter Hofstede, 2005, 2006). In theses studies authors have proposed alternative notions of soundness and more sophisticated language, making these notions undecidable.

From a practical point of view, workflow nets became a standard way to analyze workflows. They are used to guarantee the soundness property. A workflow process determines a set of activities and the specific order they are to be performed to reach a common goal. Such processes apply in different application domains such as: manufacturing, finance, marketing, etc. Unfortunately, current commercial systems do not incorporate verification techniques of workflows (van der Aalst, 2011). Therefore, the need for an analytical method to verify the correctness of workflow specification is becoming a fundamental task. Designers have the propensity to make many errors in process modeling. For example, the report in Mendling et al. (2007) and van der Aalst (2011), based on more than 2000 process models, demonstrated that more than 10 percent of these models have errors. Moreover, many errors were discovered using workflow nets in the SAP reference model (Mendling et al., 2006, Mendling, Verbeek, van Dongen, van der Aalst, & Neumann, 2008; van der Aalst, 2011), and more than 20 percent have mistakes. Fixing such mistakes can be a process that implies time and high labor costs.

Therefore, a challenging problem for Petri nets is to provide analytical methods able to develop useful procedures for showing the soundness of the workflow nets. To our knowledge, there are only two analytical methods reported in the literature. Barkaoui and Ayed (2011) show the ability of structure theory of Petri nets to conduct a uniform verification for large subclasses of parameterized workflow nets modeling control flow patterns associated with complex synchronization mechanisms, routing constructs and resource allocation constraints. Clempner (2014) solves the classical soundness property for workflow nets from a structural point of view applying the Lyapunov theory of stability, showing that a finite and nonblocking workflow net satisfies the sound property if and only if its corresponding PN is stable, i.e., given the incidence matrix A of the corresponding Petri Net there exists a Φ strictly positive vector such that $A\Phi \leqslant 0$. In this work, we present a complete different method from a trajectory-tracking approach, showing that a finite and nonblocking Decision Process Petri net (DPPN) validate a well-formed business process if and only if its corresponding DPPN is uniformly practically stable, i.e. the Petri net is tracked forward from its source place and a natural form of termination is ensured by a sink.

DPPN allows a dynamical model representation to be expressed in terms of difference equations. The advantage of this approach is its ability to represent the dynamic behavior of the business process. A decision-process Petri net model of a workflow net gives a specific and unambiguous description of the behavior of the business process. Its solid mathematical foundation has resulted in different analysis methods and tools. In spite of the formal background, DPPN models are easy to understand. DPPN corresponds to a series of strategies which guide the selection of actions that lead to a final (decision) state. By taking into account different possible courses of action, the overall utility of each strategy is considered. The utility function of each business process is represented by a Lyapunov-like function. Conditions of equilibrium and stability for the DPPN are analyzed.

In this contribution DPPN theory is used as an abstraction of the workflow to check the soundness property. We present an analytical method and its theoretical limits for workflow verification:

- We use the Lyapunov stability theory to tackle the soundness verification problem for decision-process Petri nets: the well-formed verification (soundness) property is solved showing that the workflow net representation using decision-process Petri nets is uniformly practically stable.
- We show that the problem of finding an optimum trajectory for validation of well-formed business processes is solvable: given a workflow net the computation can always be completed, that is, it is possible to show that a process initiated in the source place and regardless of how the computation proceeds at the beginning, the DPPN has always a trajectory able to reach the sink place of the Petri net.
- We demonstrate that checking the soundness property is computationally tractable, calculating the computational complexity of finding an optimum trajectory for solving the problem.
- We prove the connection between workflow nets and partially ordered decision-process Petri nets used for business process representation and analysis.
- We validate the proposed method successfully, by a numerical example related with supply chains

1.2. Main results

This paper presents a trajectory-tracking approach for verifying soundness of workflow/Petri nets represented by a DPPN (Clempner, 2010). Well-formed business processes correspond to sound workflow nets (van der Aalst, 2011, 2013). The advantage of this approach is its ability to represent the dynamic behavior of the business process. It is important to note that in a complexity-theoretic sense checking the soundness property is computationally tractable and the use of a Lyapunov-like function U guarantee a convergence in a time step bounded by $O(U_0/\epsilon)$ where $\epsilon = \min\{\epsilon_i\}$ equals the length of the shortest-path. The results are summarized as follows:

Theorem. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking workflow net. Then, the DPPN satisfies the soundness property iff $U(p_{i+1}) - U(p_i) \leqslant 0$, i.e. it is uniformly practically stable.*

Theorem. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite an nonblocking workflow net. The problem of finding an optimum trajectory for validation of soundness of a workflow net is solvable.*

Theorem. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net and let $(p_0, p_1 \ldots, p_n)$ be a realized trajectory which converges to p^* such that $\exists \epsilon_i : |U_{i+1} - U_i| > \epsilon_i$ (with $\epsilon_i > 0$). Let $\epsilon = \min\{\epsilon_i\}$, then an optimum point p^* is reached in a time step bounded by $O(U_0/\epsilon)$.*

1.3. Organization of the paper

The rest of the paper is structured in the following manner. The next section presents the necessary mathematical background and terminology on Petri nets needed to understand the rest of the paper. In Section 3, we motivate the need for the soundness workflow verification technique, the goal is not to formally present the method but to provide a high-level overview of how it works. We present the basic notion of a workflow net and soundness followed by the definition of soundness. We also describe and exemplify the finite and nonblocking conditions established for the Petri net. Section 4 describes the basic formalism of DPPN. Section 5 outlines the core content of the paper presenting the basic notions of stability and the main result of the paper about the soundness property by trajectory. Here we present a formal approach of how the soundness property can be calculated over a finite and nonblocking Petri nets that represents a workflow net. We also make emphasis on the reasons which are why the finite and nonblocking conditions can not be relaxed. Section 6 describes the connection between workflow nets and partially ordered transition DPPN. In Section 7 we present application examples which pragmatically illustrates the application of the method. Finally, in Section 8 some concluding remarks and future work are outlined.

2. Preliminaries

In this section, we present some well-established definitions and properties which will be used later.

Notation. Let $\mathbb{N} = \{0, 1, 2, \ldots\}$, $\mathbb{N}_+^{n_0} = \{n_0, n_0 + 1, \ldots, n_0 + k, \ldots\}$, $n_0 \geqslant 0$, $\mathbb{R} = (-\infty, \infty)$ and $\mathbb{R}_+ = [0, \infty)$.

Consider systems of first ordinary difference equations given by

$$x(n+1) = f[n, x(n)], x(n_o) = x_0, n \in \mathbb{N}_{n_0+} \quad (1)$$

where $n \in \mathbb{N}_+^{n_0}$, $x(n) \in \mathbb{R}^d$ and $f : \mathbb{N}_+^{n_0} \times \mathbb{R}^d \to \mathbb{R}^d$ is continuous in $x(n)$.

Definition 1. The n-vector valued function $\Phi(n, n_0, x_0)$ is said to be a solution of (1) if $\Phi(n_0, n_0, x_0) = x_0$ and $\Phi(n+1, n_0, x_0) = f(n, \Phi(n, n_0, x_0))$ for all $n \in \mathbb{N}_+^{n_0}$.

Definition 2. The system (1) is said to be

(i) Practically stable, if given (λ, A) with $0 < \lambda < A$, then

$$|x_0| < \lambda \Rightarrow |x(n, n_0, x_0)| < A, \forall n \in \mathbb{N}_{n_0+}, n_0 \geqslant 0;$$

(ii) Uniformly practically stable, if it is practically stable for every $n_0 \geqslant 0$.

The following class of function is defined.

Definition 3. A continuous function $\alpha : [0, \infty) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if $\alpha(0) = 0$ and it is strictly increasing.

2.1. Methods for practical stability

Consider (Lakshmikantham, Leela, & Martynyuk, 1990, 1991) the vector function $v(n, x(n)), v : \mathbb{N}_+^{n_0} \times \mathbb{R}^d \to \mathbb{R}_+^p$ and define the variation of v relative to (1) by

$$\Delta v = v(n+1, x(n+1)) - v(n, x(n)) \quad (2)$$

Then, the following result concerns the practical stability of (1).

Theorem 4. *Let $v : \mathbb{N}_+^{n_0} \times \mathbb{R}^d \to \mathbb{R}_+^p$ be a continuous function in x, define the function $v_0(n, x(n)) = \sum_{i=1}^{p} v_i(n, x(n))$ such that it satisfies the estimates.*

$$b(|x|) \leqslant v_0(n, x(n)) \leqslant a(|x|) \quad \text{for } a, b \in \mathcal{K} \quad \text{and}$$

$$\Delta v(n, x(n)) \leqslant w(n, v(n, x(n)))$$

for $n \in \mathbb{N}_+^{n_0}, x(n) \in \mathbb{R}^d$, where $w : \mathbb{N}_+^{n_0} \times \mathbb{R}_+^p \to \mathbb{R}^p$ is a continuous function in the second argument.

Assume that: $g(n, e) \triangleq e + w(n, e)$ is nondecreasing in e, $0 < \lambda < J$ are given and finally that $a(\lambda) < b(A)$ is satisfied. Then, the practical stability properties of

$$e(n+1) = g(n, e(n)), e(n_0) = e_0 \geqslant 0. \quad (3)$$

imply the corresponding practical stability properties of the system (1).

Corollary 5. *In Theorem 4*

1. If $w(n, e) \equiv 0$ we obtain uniform practical stability of (1).
2. If $w(n, e) = -c(e)$, for $c \in \mathcal{K}$, we obtain uniform practical asymptotic stability of (1).

2.2. Petri nets

A (marked) Petri net is a 5-tuple $PN = (P, Q, F, W, M_0)$ where: $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of *places*, $Q = \{q_1, q_2, \ldots, q_n\}$ is a finite set of *transitions* with $P \cap Q = \emptyset$ and, P and Q are nonempty such that $P \cup Q \neq \emptyset, F \subseteq (P \times Q) \cup (Q \times P)$ is a set of arcs and determines a *flow relation*, $W : F \to \mathbb{N}_+^1$ is a *weight function*, $M_0 : P \to \mathbb{N}$ is the *initial marking*. We adopt the standard rules about representing nets as directed graphs, namely places are represented as circles, transitions as rectangles, the flow relation by arcs, and markings are shown by placing tokens within circles. At any time a place contains zero or more tokens, drawn as black dots (Murata, 1989).

For each transition or place z we will denote $\bullet z := \{y \in P \cup Q | (y, z) \in F\}$, the preset of z. Analogously we will denote $z\bullet = \{y \in P \cup Q | (z, y) \in F\}$ the postset of z. A source place is a place $p_0 \in P$ such that $\bullet p_0 = \emptyset$ (there are no incoming arcs into place p_0). A sink place is a place $p \in P$ such $p\bullet = \emptyset$ (there are no outgoing arcs from p).

A *Petri net structure* without any specific initial marking is denoted by PN. A Petri net with the given initial marking is denoted by (PN, M_0). Notice that if $W(p, q) = a$ or $W(q, p) = b$ for $a, b \in \mathbf{N}_+^1$ then, this is often represented graphically by a, (b) arcs from p to q (q to p) each with no numeric label.

Let $M_k(p_i)$ denote the marking (i.e., the number of tokens) at place $p_i \in P$ at time k and let $M_k = [M_k(p_1), \ldots, M_k(p_m)]^T$ denote the marking (state) of PN at time k. A transition $q_j \in Q$ is said to be *enabled* at time k if $M_k(p_i) \geqslant W(p_i, q_j)$ for all $p_i \in P$ such that $(p_i, q_j) \in F$ $(\forall p_i \in \bullet q_j)$. It is assumed that at each time k there exists at least one transition to fire. If a transition is enabled then, it can fire. If an enabled transition $q_j \in Q$ fires at time k then, the next marking M_{k+1}, written as $M_k \xrightarrow{q_j} M_{k+1}$, for $p_i \in P$ is given by

$$M_{k+1}(p_i) = M_k(p_i) + W(q_j, p_i) - W(p_i, q_j). \quad (4)$$

Let $A = [a_{ij}]$ denote an $n \times m$ matrix of integers, called the *incidence matrix*, where $a_{ij} = a_{ij}^+ - a_{ij}^-$ with $a_{ij}^+ = W(q_i, p_j)$ and $a_{ij}^- = W(p_j, q_i)$. Let $u_k \in \{0, 1\}^n$ denote a *firing vector*where if $q_j \in Q$ is fired then, its corresponding firing vector is $u_k = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ with the one in the j^{th} position in the vector and zeros everywhere else. The matrix equation (nonlinear difference equation) describing the dynamical behavior represented by a Petri net is:

$$M_{k+1} = M_k + A^T u_k \quad (5)$$

where if at step $k, a_{ij}^- < M_k(p_j)$ for all $p_j \in P$ then, $q_i \in Q$ is enabled and if this $q_i \in Q$ fires then, its corresponding firing vector u_k is utilized in the difference equation (5) to generate the next step. Notice that if M' can be reached from some other marking M and, if we fire some sequence of d transitions with corresponding firing vectors $u_0, u_1, \ldots, u_{d-1}$ we obtain that

$$M' = M + A^T u, u = \sum_{k=0}^{d-1} u_k. \quad (6)$$

Given $\sigma = q_1, q_2, \ldots, q_n \in Q^*$ (i.e. $q_i \in Q$), where Q^* is the reflexive transitive closure of Q, we write $M_0 \xrightarrow{\sigma} M_n$ if there exists markings $M_1, \ldots, M_{n-1}$ such that $M_0 \xrightarrow{q_1} M_1 \xrightarrow{q_2} M_2 \ldots, M_{n-1} \xrightarrow{q_n} M_n$. Then, we say that M_n is *reachable*. The set of reachable markings of *PN* is denoted by $R(PN, M_0)$, called the *reachability set*, and is defined by

$$R(PN, M_0) = \left\{ M | \exists \sigma \in Q^* M_0 \xrightarrow{\sigma} M_k : 0 \leqslant k \leqslant n \right\}$$

A Petri net *PN* is *s-bounded* if $M(p) \leqslant s$ for every reachable marking M and every place p of *PN*, and *bounded* if it is s-bounded for some $s \geqslant 0$. A 1-bounded net is also called *safe*.

A Petri net is *strongly connected* if for every two nodes n_1 and $n_2, n_1, n_2 \in P \cup Q$, there exists a directed path leading from n_1 to n_2.

A Petri net *PN* is a *free-choice* Petri net if for every two transitions $q_i, q_j \in Q, \bullet q_i \cap \bullet q_j \neq \emptyset$ implies $\bullet q_i = \bullet q_j$.

Let $(\mathbf{N}_+^{n_0}, d)$ be a metric space where $d : \mathbf{N}_+^{n_0} \times \mathbf{N}_+^{n_0} \to \mathbb{R}_+$ is defined by

$$d(M_1, M_2) = \sum_{i=1}^{m} \zeta_i |M_1(p_i) - M_2(p_i)|; \quad \zeta_i > 0, \ i = 1, \ldots, m. \quad (7)$$

3. Motivation

A workflow net is a Petri net with two distinguished input and output places without input and output transitions respectively, and such that the addition of a reset transition leading back from the output to the input place makes the net strongly connected. Formally,

Definition 6. A Petri net $PN = (P, Q, F, W, M)$ is a *workflow net* if:

- there exist places $i, o \in P$ such that $\bullet i = \emptyset = o \bullet, M(p) = 1$ for $p = i$ and $M(p) = 0$ otherwise
- every node is in a path from i to o, i.e. for any $x \in P \cup Q : (i, x) \in F^*$ and $(x, o) \in F^*$ where F^* is the reflexive-transitive closure of relation F.

Then, the resulting Petri net is strongly connected.

A workflow net *PN* is *sound* if it is live and bounded (van der Aalst, 1998, 2011).

Definition 7. Let *PN* be a workflow net. *PN* is sound if the following three requirements are satisfied:

1. For every state M reachable from state M_i, there exists a ring sequence leading from state M to state M_o:

   for all $M : (M_i \xrightarrow{\sigma} M) \Rightarrow (M \xrightarrow{\sigma} M_o)$
2. State M_o is the only state reachable from state M_i with at least one token in place M_o:

   for all $M : (M_i \xrightarrow{\sigma} M \wedge M \geqslant 0) \Rightarrow (M = M_o)$
3. There are no dead transitions in *PN*:

   for all $q \in Q$, there exist $M, M\prime : (M_i \xrightarrow{\sigma} M \xrightarrow{q} M\prime)$

The first requirement states that starting from the initial state M_i, it is always possible to reach the state with one token in place o. The second requirement states that the moment a token is put in place o, all the other places should be empty. The third requirement has been added to avoid activities and conditions which do not contribute to the processing of cases. Nevertheless it is looked-for soundness of workflow nets, many of the real models with conditional behavior will not satisfy third requirement: "no dead transitions" in *PN*. The problem is usually produced by the operations needed to be modeled and not necessarily by the structure of the net. In this sense, a workflow satisfies the *soundness* property if given its corresponding Petri net (finite and nonblocking) which is tracked forward, if one starts with a single token in the source and regardless of how the computation proceeds at start, it is always possible to reach a state with the token in the sink place.

Definition 8. Let *PN* be a workflow net. *PN* is sound if the following two requirements are satisfied:

1. For every state M reachable from state M_i, there exists a firing sequence leading from state M to state M_o:

   for all $M : (M_i \xrightarrow{\sigma} M) \Rightarrow (M \xrightarrow{\sigma} M_o)$
2. State $M(o)$ is the only state reachable from state M_i with at least one token in place M_o:

   for all $M : (M_i \xrightarrow{\sigma} M \wedge M \geqslant 0) \Rightarrow (M = M_o)$

The soundness notions discussed so far consider all possible execution paths and if for one path the desired. The *PN* represented in Fig. 1 presents a cycle. It has the property of stability, but the end state is not reachable, so the net is not sound.

The PN represented in Fig. 2 represents a block. It has the property of stability. But, the sink of the PN never can be reached.

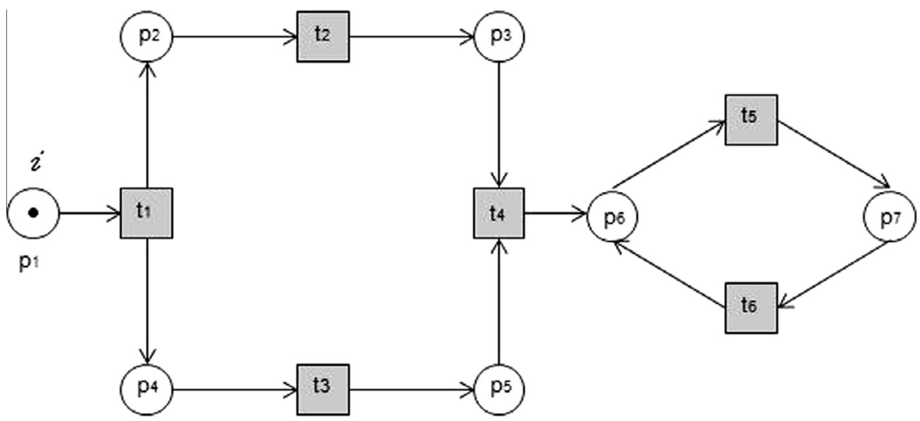

Fig. 1. Petri net cycle.

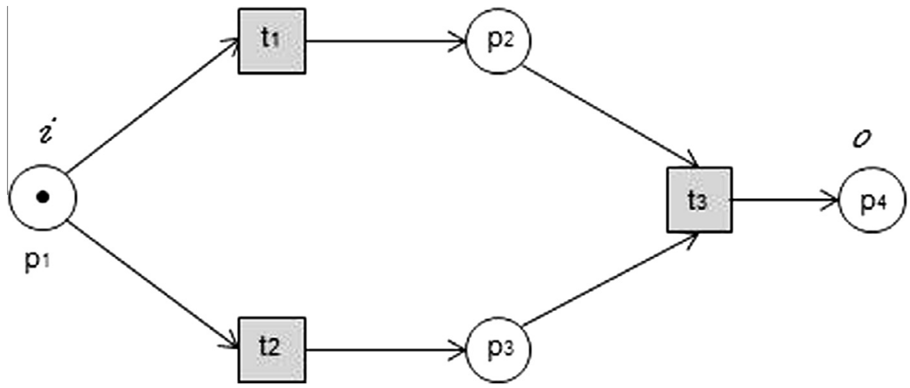

Fig. 2. Petri net block.

Soundness requires that a workflow net can always terminate in the sink of the *PN*. Therefore, if we want to use stability as theoretic notion for finding soundness of a workflow net, it will be required to impose two conditions over the corresponding *PN*: finite (no cycles) and nonblocking.

4. Decision processes Petri nets

In this section, we present some definitions and properties in DPPN (Clempner, 2010) which will be used later.

Definition 9. A Decision Process Petri net is a 7-tuple $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ where

- $P = \{p_0, p_1, p_2, \ldots, p_m\}$ is a finite set of places,
- $Q = \{q_1, q_2, \ldots, q_n\}$ is a finite set of transitions,
- $F \subseteq I \cup O$ is a set of arcs where $I \subseteq (P \times Q)$ and $O \subseteq (Q \times P)$ such that $P \cap Q = \emptyset$ and $P \cup Q \neq \emptyset$,
- $W : F \to N_1^+$ is a weight function,
- $M_0 : P \to N$ is the initial marking,
- $\pi : I \to \mathbb{R}_+$ is a routing policy representing the probability of choosing a particular transition, such that for each $p \in P$, $\sum_{q_j:(p,q_j)\in I} \pi((p, q_j)) = 1$,
- $U : P \to \mathbb{R}_+$ is a utility function.

In Figs. 3 and 4 we have represented partial routing policies π that generates a transition from state p_1 to state p_2 where $p_1, p_2 \in P$:

- case 1. In Fig. 3 the probability that q_1 generates a transition from state p_1 to p_2 is 1/3. But, because q_1 transition to state p_2 has two arcs, the probability to generate a transition from state p_1 to p_2 is increased to 2/3.

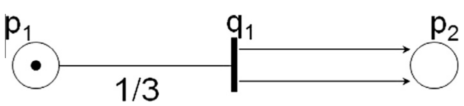

Fig. 3. Routing policy case 1.

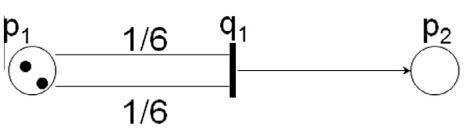

Fig. 4. Routing policy case 2.

- case 2. In Fig. 4 we set by convention for the probability that q_1 generates a transition from state p_1 to p_2 is 1/3 (1/6 plus 1/6). However, because q_1 transition to state p_2 has only one arc, the probability to generate a transition from state p_1 to p_2 is decreased to 1/6.
- case 3. Finally, we have the trivial case when there exists only one arc from p_1 to q_1 and from q_1 to p_2.

Definition 10. The utility function U with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is represented by the equation

$$U_k^{q_j}(p_i) = \begin{cases} U_k(p_0) & \text{if } i = 0, k = 0 \\ L(\alpha) & \text{if } i > 0, k = 0 \,\&\, i \geqslant 0, k > 0 \end{cases} \tag{8}$$

where

$$\alpha = \left[\begin{array}{c} \sum_{h\in\eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k^{q_{j_0}}(p_h), \\ \sum_{h\in\eta_{ij_1}} \Psi(p_h, q_{j_1}, p_i) * U_k^{q_{j_1}}(p_h), \ldots, \\ \sum_{h\in\eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k^{q_{j_f}}(p_h) \end{array} \right] \tag{9}$$

the function $L : D \subseteq \mathbb{R}_+^n \to \mathbb{R}_+$ is a Lyapunov like function which optimizes the utility through all possible transitions (i.e. trough all the possible trajectories defined by the different q_js), D is the decision set formed by the j's; $0 \leqslant j \leqslant f$ of all those possible transitions $(q_j p_i) \in O$, $\Psi(p_h, q_j, p_i) = \pi(p_h, q_j) * \frac{FN(q_j, p_i)}{FN(p_h, q_j)}$, η_{ij} is the index sequence of the list of previous places to p_i through transition q_j, p_h $(h \in \eta_{ij})$ is a specific previous place of p_i through transition q_j.

Definition 11. A final decision point $p_f \in P$ with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a place $p \in P$ where the infimum or the minimum is attained, i.e. $U(p) = 0$ or $U(p) = C$.

Definition 12. An optimum point $p^{\triangle} \in P$ with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a final decision point $p_f \in P$ where the best choice is selected 'according to some criteria'.

Proposition 13. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net and let $p^{\triangle} \in P$ an optimum point. Then $U(p^{\triangle}) \leqslant U(p), \forall p \in P$ such that $p \leqslant_U p^{\triangle}$.*

Proof. We have that $U(p^{\triangle})$ is equal to the minimum or the infimum. Therefore, $U(p^{\triangle}) \leqslant U(p)\ \forall p \in P$ such that $p \leqslant_U p^{\triangle}$. □

Theorem 14. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net. If $p^{\triangle} \in P$ is an optimum point then iff it is a final decision point.*

Proof. (⇒) Since $p^{\triangle}$ is an optimum point, by definition, the best choice is selected "according to some criteria." However, this implies that the routing policy attached to the transition (s) that follows $p^{\triangle}$ is 0, (in case there is such a transition (s) i.e., worst case). Therefore, its utility can not be modified and since $U(p_i)$ is a decreasing function an infimum or a minimum is attained. Then, $p^{\triangle}$ is a final decision point.

(⇐) If p_f is a final decision point, since the *DPPN* is finite, there exists a k such that $U_k(p_f) = C$. Let us suppose that p_f is not an optimum point. Then, it is not the last place in the net. So, it is possible to modify the utility over p_f. As a result, it is possible to find routing policy attached to the transition (s) that follows p_f different that 0 and obtain a lower utility than C. Its a contradiction. □

Definition 15. A strategy with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is identified by σ and consists of the routing policy transition sequence represented in the *DPPN* graph model such that some point $p \in P$ is reached.

Definition 16. An optimum strategy with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is identified by $\sigma^{\triangle}$ and consists of the routing policy transition sequence represented in the *DPPN* graph model such that an optimum point $p^{\triangle} \in P$ is reached.

Equivalently we can represent (8) and (9) as follows:

$$U_k^{\sigma_{hj}}(p_i) = \begin{cases} U_k(p_0) & \text{if } i = 0, k = 0 \\ L(\alpha) & \text{if } i > 0,\ k = 0\ \&\ i \geqslant 0,\ k > 0 \end{cases} \tag{10}$$

$$\alpha = \begin{bmatrix} \sum_{h \in \eta_{ij_0}} \sigma_{hj_0}(p_i) * U_k^{\sigma_{hj_0}}(p_h), \\ \sum_{h \in \eta_{ij_1}} \sigma_{hj_1}(p_i) * U_k^{\sigma_{hj_1}}(p_h), \ldots, \\ \sum_{h \in \eta_{ij_f}} \sigma_{hj_f}(p_i) * U_k^{\sigma_{hj_f}}(p_h) \end{bmatrix} \tag{11}$$

where $\sigma_{hj}(p_i) = \Psi(p_h, q_j, p_i)$. The rest is as previous defined.

Notation 17. With the intention to facilitate even more the notation we will represent the utility function U as follows:

1. $U_k(p_i) \triangleq U_k^{q_j}(p_i) \triangleq U_k^{\sigma_{hj}}(p_i)$ for any transition and any strategy,
2. $U_k^{\triangle}(p_i) \triangleq U_k^{q_j^{\triangle}}(p_i) \triangleq U_k^{\sigma_{hj}^{\triangle}}(p_i)$ for an optimum transition and optimum strategy.

5. Validation well-formed workflow Petri nets

Theorem 18. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite an nonblocking workflow net. Then, the DPPN satisfies the soundness property iff $U(p_{i+1}) - U(p_i) \leqslant 0$, i.e. it is uniformly practically stable.*

Proof. (⇒) Let us choose $v = Id(U(p_i))$ then $\Delta v = U(p_{i+1}) - U(p_i) \leqslant 0$, then by the autonomous version of Theorem 4 and Corollary 5, the DPPN is stable.

(⇐) We want to show that the workflow net is practically stable, i.e., given $0 < \lambda < A$ we must show that $|U(p_i)| < A$. We know that $U(p_0) < \lambda$ and since U is non-decreasing we have that $|U(p_i)| < |U(p_0)| < \lambda < A$. □

Remark 19. The finite and nonblocking conditions over the workflow net cannot be relaxed and reinforce the definition of workflow (Definition 7):

1. If the workflow is into a cycle it will satisfy the theoretic notion of stability, but it will never reach the sink place of the net. If we required termination without this assumption, all nets allowing loops in their execution sequences would be called unsound, which is clearly not desirable.
2. If we suppose that the workflow net blocks at some place p it will also satisfies the theoretic notion of stability, but it will never reach the sink place of the net.

Proposition 20. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be workflow net. The finite and nonblocking (unless $p \in P$ is an optimum point) condition over the DPPN workflow net can not be relaxed:*

Proof.

(1) Let us suppose that the workflow net is not finite, i.e. p is in a cycle then, the Lyapunov-like function converges when $k \to \infty$, to zero i.e., $L(p) = 0$ but the DPPN has no final place therefore, it is not an optimum point.

(2) Let us suppose that the workflow net blocks at some place (not an optimum point) $p \in P$. Then, the Lyapunov-like function has a minimum at place p, lets say $L(p) = C$ but p is not an optimum point, because it is not necessary to have a sink in the net. □

Definition 21. Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net. A trajectory ω is an (finite or infinite) ordered subsequence of places $p_{\varsigma(1)} \leqslant_{U_k} p_{\varsigma(2)} \leqslant_{U_k} \cdots \leqslant_{U_k} p_{\varsigma(n)} \leqslant_{U_k} \cdots$ such that a given strategy σ holds.

Definition 22. Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net. An optimum trajectory ω is an (finite or infinite) ordered subsequence of places $p_{\varsigma(1)} \leqslant_{U_k^{\triangle}} p_{\varsigma(2)} \leqslant_{U_k^{\triangle}} \cdots \leqslant_{U_k^{\triangle}} p_{\varsigma(n)} \leqslant_{U_k^{\triangle}} \cdots$ such that the optimum strategy $\sigma^{\triangle}$ holds.

Theorem 23. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a nonblocking workflow net (unless $p \in P$ is an optimum point) then we have that:*

$$U_k^{\triangle}(p^{\triangle}) \leqslant U_k(p), \forall \sigma, \sigma^{\triangle}$$

Proof. We have that

$$U_k^{\sigma_{hj}}(p_i) = \begin{cases} U_k(p_0) & \text{if } i = 0, k = 0 \\ L(\alpha) & \text{if } i > 0,\ k = 0\ \&\ i \geqslant 0,\ k > 0 \end{cases}$$

$$\alpha = \begin{bmatrix} \sum_{h \in \eta_{ij_0}} \sigma_{hj_0}(p_i) * U_k^{\sigma_{hj_0}}(p_h), \\ \sum_{h \in \eta_{ij_1}} \sigma_{hj_1}(p_i) * U_k^{\sigma_{hj_1}}(p_h), \ldots, \\ \sum_{h \in \eta_{ij_f}} \sigma_{hj_f}(p_i) * U_k^{\sigma_{hj_f}}(p_h) \end{bmatrix}$$

Then, starting from p_0 and proceeding with the iteration, eventually the trajectory ω given by $p_0 = p_{\varsigma(1)} \leqslant_{U_k} p_{\varsigma(2)} \leqslant_{U_k} \cdots \leqslant_{U_k} p_{\varsigma(n)} \leqslant_{U_k} \ldots$ which converges to $p^{\triangle}$, i.e., the optimum trajectory, is obtained. Since at the optimum trajectory the optimum strategy $\sigma^{\triangle}$ holds, we have that $U_k^{\triangle}(p^{\triangle}) \leqslant U_k(p), \forall \sigma, \sigma^{\triangle}$. □

Corollary 24. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a nonblocking workflow net (unless $p \in P$ is an optimum point) and let $\sigma^{\triangle}$ an optimum strategy. Set $L = \min_{i=1,\ldots,|\alpha|}\{\alpha_i\}$ then, $U_k^{\triangle}(p)$is equal to:*

$$\underbrace{\begin{array}{|c|c|c|c|}\hline \sigma^{\triangle}_{0j_m}(p_{\varsigma(0)}) & \sigma^{\triangle}_{1j_m}(p_{\varsigma(0)}) & .. & \sigma^{\triangle}_{nj_m}(p_{\varsigma(0)}) \\ \hline \sigma^{\triangle}_{0j_n}(p_{\varsigma(1)}) & \sigma^{\triangle}_{1j_n}(p_{\varsigma(1)}) & .. & \sigma^{\triangle}_{nj_n}(p_{\varsigma(1)}) \\ \hline \ldots & \ldots & .. & \ldots \\ \hline \sigma^{\triangle}_{0j_v}(p_{\varsigma(i)}) & \sigma^{\triangle}_{1j_v}(p_{\varsigma(i)}) & .. & \sigma^{\triangle}_{nj_v}(p_{\varsigma(i)}) \\ \hline \ldots & \ldots & .. & \ldots \\ \hline \end{array}}_{\sigma^{\triangle}} \; \underbrace{\begin{array}{|c|}\hline U_k(p_0) \\ \hline U_k(p_1) \\ \hline \ldots \\ \hline U_k(p_i) \\ \hline \ldots \\ \hline \end{array}}_{U} \tag{12}$$

where p is a vector whose elements are those places which belong to the optimum trajectory ω given by $p_0 \leqslant p_{\varsigma(1)} \leqslant_{U_k} p_{\varsigma(2)} \leqslant_{U_k} \cdots \leqslant_{U_k} p_{\varsigma(n)} \leqslant_{U_k} \ldots$ which converges to $p^{\triangle}$.

Definition 25. A Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is said to be symmetric if it is possible to decompose it into some finite number (greater that 1) of sub-Petri nets in such a way that there exists a bijection ψ between all the sub-Petri nets such that

$$(p, q) \in I \iff (\psi(p), \psi(q)) \in I \quad \text{and}$$

$$(q, p) \in O \iff (\psi(q), \psi(p)) \in O$$

for all of the sub-Petri nets.

Corollary 26. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a nonblocking (unless p is an optimum point) symmetric workflow net and let $\sigma^{\triangle}$ be an optimum.strategy. Set $L = \min_{i=1,\ldots,|\alpha|}\{\alpha_i\}$ then,*

$$\sigma^{\triangle} U \leqslant \sigma U \qquad \forall \sigma, \sigma^{\triangle}$$

where the σ and $\sigma^{\triangle}$ are represented by a matrix and U is represented by a vector.

Given a nonblocking (unless p is an equilibrium point) Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$, the optimum trajectory planning consists on finding the firing transition sequence u such that the optimum target state M_t, associated to the optimum point, is achieved. The target state M_t belong to the reachability set $R(M_0)$, and satisfies that it is the last and final task processed by the DPPN with some fixed starting state M_0 with utility U_0.

Theorem 27. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite an nonblocking workflow net. The problem of finding an optimum trajectory for validation of soundness of a workflow net is solvable.*

Proof. From what was shown in Theorem 23 for each step we find $U_k^{\triangle}(p_{\varsigma(1)}), \ldots, U_k^{\triangle}(p_{\varsigma(i)}), \ldots, U_k^{\triangle}(p^{\triangle})$. Define a mapping (see Notation 17)

$$u_r(U_k^{q_j^{\triangle}}(p_{\varsigma(i)})) = [0, \ldots, 0, 1, 0, \ldots, 0] \tag{13}$$

with "1" in position $j^{\triangle}$ and zero everywhere else, and set $u = \sum_r u_r((U_k^{q_j^{\triangle}}(p_{\varsigma(i)}))$, where the index r runs over all the transitions associated to the subsequence $\varsigma(i)$ such that $p_{\varsigma(i)}$ converges to p^{Δ}, then, by construction the optimum point is attained. □

Remark 28. The order in which the transitions are fired, is given by the order of the transitions, inherited from the order of the subsequence $p_{\varsigma(i)}$.

Theorem 29. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a workflow net and let $(p_0, p_1 \ldots, p_n)$ be a realized trajectory which converges to p^* such that $\exists \epsilon_i : |U_{i+1} - U_i| > \epsilon_i$ (with $\epsilon_i > 0$). Let $\epsilon = \min\{\epsilon_i\}$, then an optimum point p^* is reached in a time step bounded by $O(U_0/\epsilon)$.*

Proof. Let us suppose that p^* is never reached, then, p^* is not the last place in the DPPN. So, it is possible to find some output transition to p^*. Therefore, it is possible to reduce the trajectory function value over p^* by at least ϵ. As a result, it is possible to obtain a lower value than C (that is a contradiction). □

Remark 30. The complexity time $O(U_0/\epsilon)$ differs with that of the Dijkstra's algorithm.

Remark 31. Each path in *DPPN* corresponds to a trajectory of a given system. The trajectory-tracking function value of U at the source place (U_0) divided by $\epsilon = \min\{\epsilon_i\}$ equals the length of the shortest-path. Then, the infimum is equivalent to the infimum length over all paths in *DPPN*.

Theorem 32. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a workflow net. Then, U converges to a point p^*.*

Proof. We have to show that U converges to a point p^*. By the previous theorem the optimum point p^* is reached in a time step bounded by $O(U_0/\epsilon)$, therefore U converges to p^*. □

6. Connection between workflow nets and partially ordered transition DPPN

In business process modeling (Clempner & Retchkiman, 2005), high-level business strategies are refined up to the point when they reach a tactical business strategy level, described only in terms of goals and strategies.[1]

Business strategy decomposition represents a hierarchy of objective/decision-points, varying from the high-level business strategy with the maximum long-term impact to the more refined operational business strategy (goal, strategy) with relative short-term impact.

The business strategy refinement process concludes when a resulting business strategy can be transformed into an executable action. In this sense, the nodes found in the lowest levels of the business strategy decomposition tree are usually mapped into actions.

A business process is regarded as a set of activities. Activities are considered as operationalizations of goals and are applied in accordance with the strategies to achieve the goals. Strategies determine the legal sequentially movements that can be made from any activity to another. The structure of each node in the business strategy decomposition is a complex object, defined by the ordered pair goal-strategy.

For completeness let us recall some basic notations of ordering. Given a poset $(X, \preceq)$ a successor of an element $x \in X$ is an element y such that $x \preceq y$, but $x \neq y$ and there is no third element u between x and y. x is a predecessor of y if y is a successor of x. In symbols, for any $x \in X$.

[1] For simplification, we decompose the business strategy in goals and strategies, which we consider is adequate from an operational point of view.

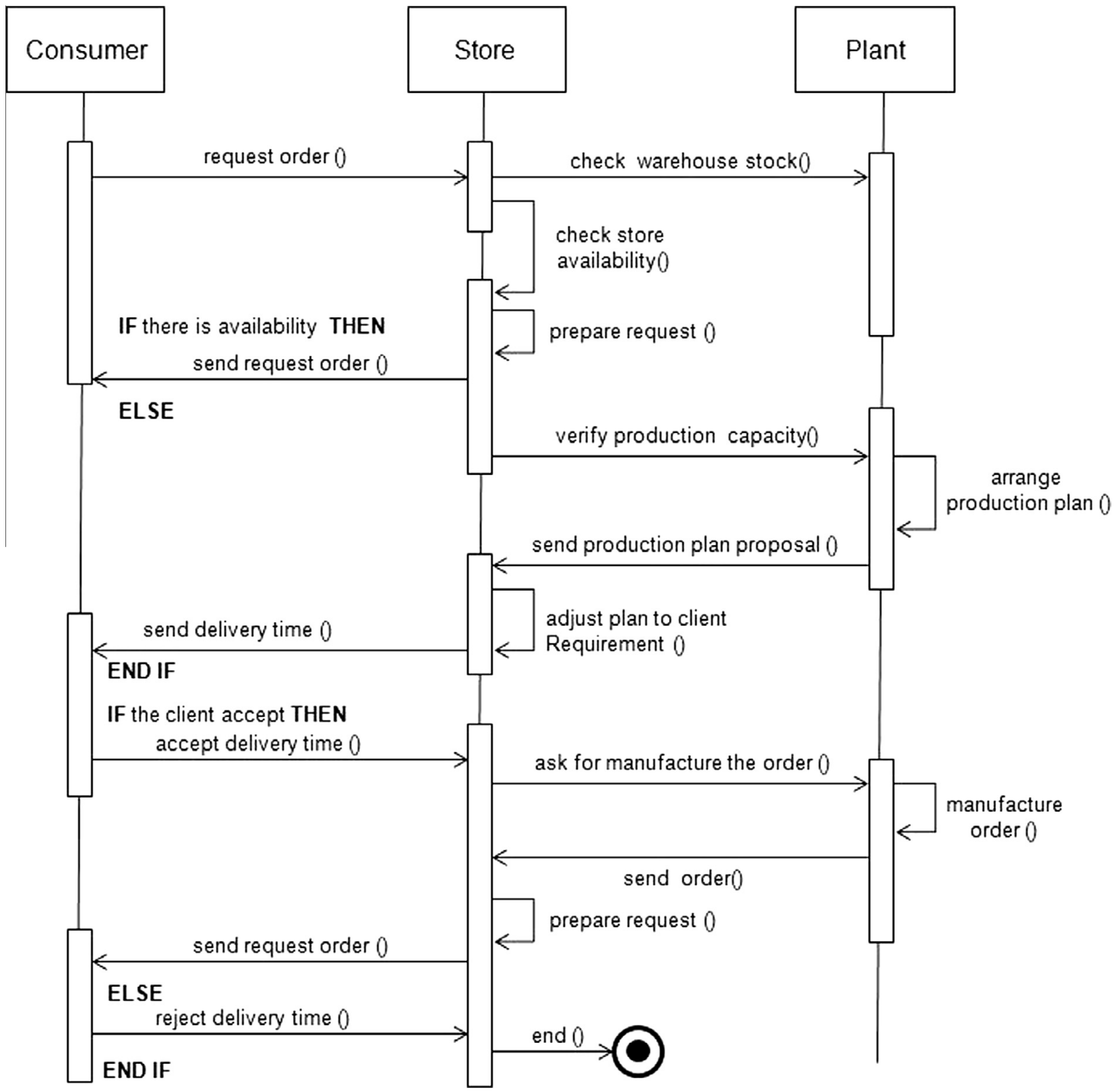

Fig. 5. Sequence diagram of the supply chains.

Successors of x: $y \in suc(x)$ iff $x \neq y, x \preceq y$ and $\forall u : x \preceq u \preceq y \Rightarrow (u = x) \vee (u = y)$.

Predecessors of x: $y \in pre(x)$ iff $y \neq x, y \preceq x$ and $\forall u : y \preceq u \preceq x \Rightarrow (u = y) \vee (u = x)$.

The graph of the ordering is the graph whose vertices are the points in X and each pair (x, y) where y is a successor of x determines an edge. The graph corresponding to the ordering "$\preceq$" defined is a directed acyclic graph (DAG).

The minimal elements are those with no predecessors, i.e. nodes with null inner degree in the *DAG*. The maximal elements are those with no successors, i.e. nodes with null outer degree in the *DAG*. In this ordering the conditions with no input transitions correspond to the minimal elements, and the conditions with no output transitions correspond to the maximal elements.

Since the business strategy decomposition determines actions sequence applications, a process can be ordered as follows.

Let X be a process and $x, y \in X$ two activities.

Definition 33. We say that the activity y "depends on" the activity x, and we denoted it by $x \preceq y$, if the corresponding decomposed node of x is upper than that of y in the business strategy decomposition tree.

Property 1. *Clearly, "$\preceq$" establishes a partial ordering.*

The partial order concept guarantees that the nodes found in the lowest levels of the business strategy decomposition tree, are already partially ordered and ready to be mapped into what next, is defined to be a partially ordered *DPPN*.

Definition 34. A partially ordered transition Decision Process Petri net is a duple $(DPPN, \preceq)$ where *DPPN* is a Decision Process Petri net and $\preceq_q$ is the partial order defined on the elements of the set of transitions Q such that the following conditions hold:

- $q_1 \prec_q q_2$ iff $q_1 \preceq_q q_2$ and $\neg(q_2 \preceq_q q_1)$
- $q_1 \sim_q q_2$ iff $q_1 \preceq_q q_2$ and $q_2 \preceq_q q_1$

Note that the order of the *DPPN* is the order established by the "depends on" relationship (see the definition of $\preceq$).

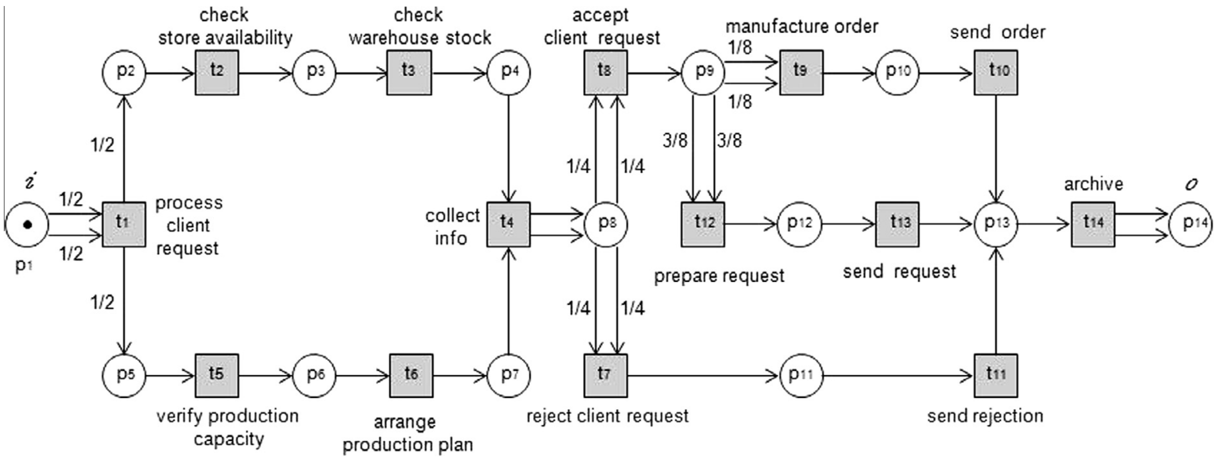

Fig. 6. Workflow net of the supply chains.

Events are actions which take place in a process. The occurrence of these events is controlled in part by the state of the process. The state of a process can be described as a set of conditions. The minimal elements of the net are those conditions associated to the initial marking. Since events are actions, they may occur, for an event to occur, it may be necessary that certain preconditions hold. Each transition has associated a strategy that determines the preconditions to hold or not and may cause post-conditions to become true.

Proposition 35. *Let us suppose that all the condition of Theorem 23 are satisfied and let us suppose that the DPPN is a partially ordered Decision Process Petri net. Then, Eq. (12) reduces to*

$$\underbrace{\begin{bmatrix} 1 & 0 & \ldots & \ldots & \ldots \\ \sigma^{\triangle}_{0j_m}(p_{\varsigma(1)}) & 0 & \ldots & \ldots & \ldots \\ \ldots & \sigma^{\triangle}_{0j_n}(p_{\varsigma(2)}) & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & 0 & \ldots \\ \ldots & \ldots & \ldots & \sigma^{\triangle}_{nj_v}(p_{\varsigma(n)}) & 0 \end{bmatrix}}_{\sigma^{\triangle}} * \underbrace{\begin{bmatrix} U_k(p_0) \\ U_k(p_1) \\ \ldots \\ U_k(p_i) \\ \ldots \end{bmatrix}}_{U}$$

Proof. It follows from Corollary 24 and the fact that the Decision Process Petri net is partially ordered. □

7. Application example

Let us consider a supply chains system involved in moving and manufacturing a product from supplier to customer (see Fig. 5). In this case, the supply chains activities involve the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption. In our case, the main products are cosmetics. The main goal of this company is to fulfill customer demands through the most efficient use of resources, seeking to match demand with supply and do so with the minimal inventory. The process begins with the costumer submitting an order of cosmetics to the store. Receiving an order by the store, results in the checking of the availability of the products locally and results in the verifying of the existence in the warehouse of the plant. If the store is able to fulfill the order, it send the requested products to the customer without delay. If the store can not fulfill the order, it asks the plant for a proposal of the availability time of the products. Developing a proposal for a production order by the plant involves realizing several logistics management activities, as well as manufacturing operations consideration, and the coordination of processes and activities with and across the manufacturing process. Once the proposal is ready, the store provides the customer with information on product availability and scheduling, indicating the delivery time of the requested products. If the customer accepts the proposal, the store asks the plant for manufacturing the order. Next, the plant draws up a strategic plan to support the manufacturing flow management process and the development of the products. Finally, if the manufacturer's processing has been successful the plant submits the order to the store and subsequently to the customer. The workflow net of the supply chains is shown in Fig. 6.

1. Optimum strategy:Define the Lyapunov like function L in terms of the Entropy $H(p_i) = -p_i \ln p_i$ as $L = \max_{i=1,\ldots,|\alpha|}(-\alpha_i \ln \alpha_i)$ then,
2. The optimum strategy $\sigma^{\triangle}$ for the supply chains is represented by:

$$U_{k=0}(p_1) = 1$$

$$U^{\sigma_{hj}}_{k=0}(p_2) = L[\sigma_{1,2}(p_2) * U^{\sigma_{1,2}}_{k=0}(p_1)] = L[2/3 * 1, 1/3 * 1] = \max H[2/3, 1/3] = 0.270$$

$$U^{\sigma_{hj}}_{k=0}(p_3) = L[\sigma_{2,3}(p_3) * U^{\sigma_{2,3}}_{k=0}(p_2)] = L[1 * 0.270] = \max H[0.270] = 0.353$$

$$U^{\sigma_{hj}}_{k=0}(p_4) = L[\sigma_{3,4}(p_4) * U^{\sigma_{3,4}}_{k=0}(p_3)] = L[1 * 0.353] = \max H[0.353] = 0.367$$

$$U^{\sigma_{hj}}_{k=0}(p_5) = L[\sigma_{1,5}(p_5) * U^{\sigma_{1,5}}_{k=0}(p_1)] = L[2/3 * 1, 1/3 * 1] = \max H[2/3, 1/3] = 0.270$$

$$U^{\sigma_{hj}}_{k=0}(p_6) = L[\sigma_{5,6}(p_5) * U^{\sigma_{5,6}}_{k=0}(p_5)] = L[1 * 0.270] = \max H[0.270] = 0.353$$

$$U^{\sigma_{hj}}_{k=0}(p_7) = L[\sigma_{6,7}(p_7) * U^{\sigma_{6,7}}_{k=0}(p_6)] = L[1 * 0.353] = \max H[0.353] = 0.367$$

$$U^{\sigma_{hj}}_{k=0}(p_8) = L[\sigma_{4,8}(p_8) * U^{\sigma_{4,8}}_{k=0}(p_4) + \sigma_{7,8}(p_8) * U^{\sigma_{7,8}}_{k=0}(p_7)] = L[(1 * 0.367 + 1 * 0.367) * 2] = \max H[1.470] = 0.566$$

$$U^{\sigma_{hj}}_{k=0}(p_9) = L[\sigma_{8,9}(p_9) * U^{\sigma_{8,9}}_{k=0}(p_8)] = L[(1/4 * 0.566, 1/8 * 0.566) * 2] = \max H[0.283, 0.141] = 0.357$$

$$U^{\sigma_{hj}}_{k=0}(p_{10}) = L[\sigma_{9,10}(p_{10}) * U^{\sigma_{9,10}}_{k=0}(p_9)] = L[1 * 0.357] = \max H[0.357] = 0.367$$

$$U^{\sigma_{hj}}_{k=0}(p_{11}) = L[\sigma_{8,11}(p_{11}) * U^{\sigma_{8,11}}_{k=0}(p_8)] = L[(1/4 * 0.566, 1/8 * 0.566) * 2] = \max H[0.283, 0.141] = 0.357$$

$$U^{\sigma_{hj}}_{k=0}(p_{12}) = L[\sigma_{9,12}(p_{12}) * U^{\sigma_{9,12}}_{k=0}(p_9)] = L[1/4 * 0.357, 3/8 * 0.357] = \max H[0.089, 0.133] = 0.269$$

$$U^{\sigma_{hj}}_{k=0}(p_{13}) = L[\sigma_{10,13}(p_{13}) * U^{\sigma_{10,13}}_{k=0}(p_{10}) + \sigma_{11,13}(p_{13}) * U^{\sigma_{11,13}}_{k=0}(p_{11}) + \sigma_{12,13}(p_{13}) * U^{\sigma_{12,13}}_{k=0}(p_{12})] = L[1 * 0.367 + 1 * 0.357 + 1 * 0.269] = \max H[0.993] = 0.006$$

By Theorem 18 we find that the DPPN is uniformly practically stable concluding soundness.

8. Conclusion

The main purpose of workflow nets is to support the definition, execution and control of workflow processes. A workflow process determines a set of activities and the specific order they are to be performed to reach a common goal. Regrettably, current commercial systems do not incorporate verification techniques of workflows (van der Aalst, 2011). Therefore the need for analytical method to verify the correctness of workflow specification is becoming a fundamental task. In this paper we reason about a basic property that any workflow-process definition should satisfy: the soundness property. This paper provided a framework for solving the soundness property verification problem of workflow nets using a trajectory-tracking approach represented by a decision-process Petri net. Using the Lyapunov stability theory on Petri nets, we have identified an analytical method for which soundness can be structurally characterized and solved effectively: a workflow net satisfies the soundness property if its PN representation is tracked forward from its source place and a natural form of termination is ensured by a sink. Validity of the proposed method was successfully demonstrated both theoretically and by a numerical example related with supply chains, where decision-process properties and validation were shown to hold was addressed.

It is important to note that the main contribution of the paper is the trajectory-tracking analytical method itself: we showed that a finite and nonblocking DPPN validate a well-formed business processes if and only if its corresponding DPPN is uniformly practically stable. We also showed that the problem of finding an optimum trajectory for validation of soundness of a workflow net is solvable. The convergence of the suggested method was analyzed. Finally, we showed the connection between workflow nets and partially ordered decision-process Petri nets used for business process representation and analysis.

There are open research questions in this area. Based on this contribution, we identified major topics for future steps. Several authors have proposed alternative notions of soundness. As a future work we will investigate these approaches in the presence of different extensions of the analytical method. The use of the Lyapunov theory can produce better results for verifying soundness in Petri nets theory. In this sense, we will extend the present idea to support colored Petri nets verification techniques. In particular, we leave open comparing the stability method efficiency setting a structural vs. a trajectory-dynamic approach. We will as well show the theoretical limits of the workflow soundness verification technique. Moreover, this paper has motivating suggestions for those working in process mining because an interesting open research challenge is that the process mining method can be improved using a Lyapunov theoretical approach in Petri nets.

References

Barkaoui, K., & Petrucci, L. (1998). Structural analysis of workflow nets with shared resources. In W. M. P. van der Aalst, G. De Michelis, & C. A. Ellis (Eds.), *Proceedings of workflow management: Net-based concepts, models, techniques and tools (WFM'98)*, Vol. 98/7 of computing science reports (pp. 82–95). Lisbon, Portugal, Eindhoven University of Technology, Eindhoven.

Barkaoui, K., & Ayed, R. B. (2011). Uniform verification of workflow soundness. *Transactions of the Institute of Measurement and Control, 33*(1), 133–148.

Bashkin, V. A., & Lomazova, I. A. (2013). On the decidability of soundness of workflow nets with an unbounded resource. *Modelirovanie i Analiz Informatsionnykh Sistem, 20*(4), 23–40.

Basu, A., & Blanning, R. W. (2000). A formal approach to workflow analysis. *Information Systems Research, 11*(1), 17–36.

Basu, A., & Kumar, A. (2002). Research commentary: Workflow management issues in e-business. *Information Systems Research, 13*(1), 1–14.

Bi, H. H., & Zhao, J. L. (2004). Applying propositional logic to workflow verification. *Information Technology and Management, 5*(3-4), 293–318.

Chen, L., Ha, W., & Zhang, G. (2013). Reliable execution based on cpn and skyline optimization for web service composition. *The Scientific World Journal*. http://dx.doi.org/10.1155/2013/729769.

Clempner, J. (2010). A hierarchical decomposition of decision process petri nets for modeling complex systems. *International Journal of Applied Mathematics and Computer Science, 20*(2), 349–366.

Clempner, J. (2014). An analytical method for well-formed workflow/petri nets verification: Classical soundness. International Journal of Applied Mathematics and Computer Science. Accepted for publication.

Clempner, J., & Retchkiman, Z. (2005). Business process modeling using decision process petri net. *WSEAS Transactions on Business and Economics, 2*(2), 36–44.

Dehnert, J., & Rittgen, P. (2001). Relaxed soundness of business processes. In K. R. Dittrich, A. Geppert, & M. C. Norrie (Eds.), *Proceedings of the 13th international conference on advanced information systems engineering (CAiSE'01). Lecture notes in computer science* (Vol. 2068, pp. 157–170). Berlin: Springer-Verlag.

Fahland, D., & van der Aalst, W. M. P. (2012). Repairing process models to reflect reality. In A. Barros, A. Gal, & E. Kindler (Eds.), *Proceedings of the international conference on business process management (BPM '12). Lecture notes in computer science* (Vol. 7481, pp. 229–245). Berlin, Germany: Springer-Verlag.

Fu, X., Bultan, T., & Su, J. (2002). Formal verification of e-services and workflows. In R. Hull, S. McIlraith, M. Orlowska, B. Pernici, C. Bussler, & J. Yang (Eds.), *Web services, e-business, and the semantic web, CAiSE 2002 international workshop (WES 2002). Lecture notes in computer science* (Vol. 2512, pp. 188–202). Berlin: Springer-Verlag.

Fu, X., Bultan, T., & Su, J. (2004). Analysis of interacting bpel web services. In *International World Wide Web conference: Proceedings of the 13th international conference on World Wide Web* (pp. 621–630). New York, NY, USA.

Hammer, M. (1990). Reengineering work don't automate obliterate. *Harvard Business Review*, 104–112.

Karamanolis, C., Giannakopoulou, D., Magee, J., & Wheater, S. (2000). Model checking of workflow schemas. In *Proceedings of the fourth international enterprise distributed object computing conference (EDOC'00)* (pp.170–181). Los Alamitos, CA, USA.

Kindler, E., Martens, A., & Reisig, W. (2000). Inter-operability of workflow applications: Local criteria for global soundness. In J. Desel, W. M. P. van der Aalst, & A. Oberweis (Eds.), *Business process management: Models, techniques, and empirical studies. Lecture notes in computer science* (Vol. 1806, pp. 235–253). Berlin: Springer-Verlag.

Lakshmikantham, V., Leela, S., & Martynyuk, A. A. (1990). *Practical stability of nonlinear systems*. Singapore: World Scientific.

Lakshmikantham, V., Matrosov, V. M., & Sivasundaram, S. (1991). *Vector Lyapunov functions and stability analysis of nonlinear systems*. Dordrecht: Kluwer Academic Publishers.

Li, B., & Iijima, J. (2007). A situation calculus based approach to dynamic management of e-business services. *Journal of Systems Science and Systems Engineering, 16*(3), 336–355.

Liu, G. J. (2013). *IEEE transactions on services computing some complexity results for the soundness problem of workflow nets. IEEE computer society digital library*. IEEE Computer Society<http://doi.ieeecomputersociety.org/10.1109/TSC.2013.36>.

Liu, W., Du, Y. Y., & Yan, C. (2012). Soundness preservation in composed logical time workflow nets. *Enterprise Information Systems, 6*(1), 95–113. http://dx.doi.org/10.1080/17517575.2011.617472.

Lohmann, N., Massuthe, P., Stahl, C., & Weinberg, D. (2006). Analyzing interacting bpel processes. In J. Fiadeiro, S. Dustdar, & A. Sheth (Eds.), *International conference on business process management (BPM 2006). Lecture notes in computer science* (Vol. 4102). Berlin: Springer-Verlag.

Martens, A. (2005a). Analyzing web service based business processes. In M. Cerioli (Ed.), *Proceedings of the eighth international conference on fundamental approaches to software engineering (FASE 2005). Lecture notes in computer science* (Vol. 3442, pp. 19–33). Berlin: Springer-Verlag.

Martens, A. (2005). Consistency between executable and abstract processes. In *Proceedings of international IEEE conference on e-technology, e-commerce, and e-services (EEE'05)* (pp. 60–67).

Mendling, J., Moser, M., Neumann, G., Verbeek, H. M. W., van Dongen, B. F., & van der Aalst, W. M. P. (2006). Faulty EPCs in the SAP reference model. In S. Dustdar, J. L. Fiadeiro, & A. Sheth (Eds.), *International conference on business process management (BPM 2006). Lecture notes in computer science* (Vol. 4102, pp. 451–457). Berlin: Springer-Verlag.

Mendling, J., Neumann, G., & van der Aalst, W. (2007). Understanding the occurrence of errors in process models based on metrics. In M. Cerioli (Ed.), *Proceedings of the OTM conference on cooperative information systems (CoopIS 2007). Lecture notes in computer science* (Vol. 4803, pp. 113–130). Berlin: Springer-Verlag.

Mendling, J., Verbeek, H. M. W., van Dongen, B. F., van der Aalst, W. M. P., & Neumann, G. (2008). Detection and prediction of errors in EPCs of the SAP reference model. *Data and Knowledge Engineering, 64*(1), 312–329.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Automatica, 77*(4), 541–580.

Sadiq, W., & Orlowska, M. (2000). Analyzing process models using graph reduction techniques. *Information Systems, 25*(2), 117–134.

Salimifard, K., & Wright, M. (2001). Petri net-based modelling of workflow systems: An overview. *European Journal of Operational Research, 134*(3), 664–676.

van der Aalst, W. M. P. (1998). The application of petri nets to workflow management. *The Journal of Circuits, Systems and Computers, 8*(1), 21–66.

van der Aalst, W. (2007). Workflow verification: Finding control-flow errors using petri-net-based techniques. In J. Desel, W. M. P. van der Aalst, & A. Oberweis (Eds.), *Business process management: Models, techniques, and empirical studies* (pp. 161–183). Berlin: Springer-Verlag.

van der Aalst, W. M. P. (2011). Soundness of workflow nets: Classification, decidability, and analysis. *Formal Aspects of Computing, 23*(3), 333–363.

van der Aalst, W. M. P. (2013). Business process management: A comprehensive survey. *ISRN Software Engineering*. http://dx.doi.org/10.1155/2013/507984. Article ID 507984, 37 p.

van Dongen, B. F., van der Aalst, W., & Verbeek, H. (2005). Verification of epcs: Using reduction rules and petri nets. In O. Pastor & J. F. e Cunha (Eds.), *Proceedings of the 17th conference on advanced information systems engineering (CAiSE'05). Lecture notes in computer science* (Vol. 3520, pp. 372–386). Berlin: Springer-Verlag.

Vanhatalo, J., Völzer, H., & Leymann, F. (2007). Faster and more focused control-flow analysis for business process models through sese decomposition. In K. Lin, B. Kramer, & P. Narasimhan (Eds.), *Proceedings of service-oriented computing (ICSOC 2007). Lecture notes in computer science* (Vol. 4749, pp. 43–55). Berlin: Springer-Verlag.

van Hee, M., Sidorova, N., & van der Werf, J. M. (2013). Business process modeling using petri nets. *Transactions on petri nets and other models of concurrency VII. Lecture notes in computer science* (Vol. 7480, pp. 116–161).

van Hee, K. M., Serebrenik, A., Sidorova, N., & Voorhoeve, M. (2005). In G. Ciardo & P. Darondeau (Eds.), *Applications and theory of petri nets. Lecture notes in computer science* (Vol. 3536, pp. 250–267). Berlin: Springer-Verlag.

van Hee, K. M., Sidorova, N., & Voorhoeve, M. (2004). Generalised soundness of workflow nets is decidable. In J. Cortadella & W. Reisig (Eds.), *Application and theory of petri nets. Lecture notes in computer science* (Vol. 3099, pp. 197–215). Berlin: Springer-Verlag.

Verbeek, H. M. W., Basten, T., & van der Aalst, W. (2001). Diagnosing workflow processes using woflan. *The Computer Journal, 44*(4), 246–279.

Wombacher, A. (2006). Decentralized consistency checking in cross-organizational workflows. In *Proceedings of international conference on e-technology, e-commerce and e-service (CEC/EEE 2006)* (pp. 39–46).

Wynn, M. T., Edmond, D., van der Aalst, W., & ter Hofstede, A. (2005). Achieving a general, formal and decidable approach to the or-join in workflow using reset nets. In G. Ciardo & P. Darondeau (Eds.), *Applications and theory of petri nets. Lecture notes in computer science* (Vol. 3536, pp. 423–443). Berlin: Springer-Verlag.

Wynn, M. T., van der Aalst, W., ter Hofstede, A., & Edmond, D. (2006). Verifying workflows with cancellation regions and or-joins: An approach based on reset nets and reachability analysis. In J. Fiadeiro, S. Dustdar, & A. Sheth (Eds.), *International conference on business process management (BPM 2006). Lecture notes in computer science* (Vol. 4102, pp. 389–394). Berlin: Springer-Verlag.