AN ANALYTICAL METHOD FOR WELL-FORMED WORKFLOW/PETRI NET VERIFICATION OF CLASSICAL SOUNDNESS

JULIO CLEMPNER

Center for Economics, Management and Social Research
National Polytechnic Institute, Lauro Aguirre 120, Col. Agricultura, Miguel Hidalgo, 11360, Mexico City, Mexico
e-mail: julio@clempner.name

In this paper we consider workflow nets as dynamical systems governed by ordinary difference equations described by a particular class of Petri nets. Workflow nets are a formal model of business processes. Well-formed business processes correspond to sound workflow nets. Even if it seems necessary to require the soundness of workflow nets, there exist business processes with conditional behavior that will not necessarily satisfy the soundness property. In this sense, we propose an analytical method for showing that a workflow net satisfies the classical soundness property using a Petri net. To present our statement, we use Lyapunov stability theory to tackle the classical soundness verification problem for a class of dynamical systems described by Petri nets. This class of Petri nets allows a dynamical model representation that can be expressed in terms of difference equations. As a result, by applying Lyapunov theory, the classical soundness property for workflow nets is solved proving that the Petri net representation is stable. We show that a finite and non-blocking 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 PN, there exists a $\Phi$ strictly positive $m$ vector such that $A\Phi \leq 0$. The key contribution of the paper is the analytical method itself that satisfies part of the definition of the classical soundness requirements. The method is designed for practical applications, guarantees that anomalies can be detected without domain knowledge, and can be easily implemented into existing commercial systems that do not support the verification of workflows. The validity of the proposed method is successfully demonstrated by application examples.

Keywords: Petri nets, decidability, workflow nets, Lyapunov stability, soundness, verification.

1. Introduction

1.1. Brief review. A workflow model is put to use by feeding it to a workflow management system (zur Muehlen, 2004; Weske, 2007). The heart of a workflow management system is the workflow engine, which does the actual management (Mann, 2010). Workflow management systems are driven by business process models. Therefore, it is important to define and streamline business processes in order to improve efficiency and reduce operating cycle times. Ultimately, the success of such modeling efforts lies not only in careful technical design, but also in ensuring the well-formed business processes of such models. Effective business processes modeling involves understanding existing process defects, identifying sources of inefficiency (deadlocks, livelocks, and other anomalies), and redefining processes to increase efficiency or decrease errors. But workflow management systems do not support verification methods for business processes design (van der Aalst, 2011).

The success of workflow management systems and methodologies has been widely publicized, while the more serious failures have not. Mendling et al. (2007), based on more than 2000 process models including well-known sets of models, such as the SAP reference model, report that more than 10 percent of these models are awed.

Workflow nets were introduced by van der Aalst (1997; 1998) and are currently the most widely used model to formally describe workflow processes. Workflow nets are a formal model of business process responsible for the organization of the processing tasks. The existing graphical languages implemented by workflow management systems are typically token-based, and for this reason a transformation to Petri nets is reasonably simple.

Petri nets are a natural technique for formal modeling and analyzing workflow nets because the flow-oriented nature of workflow processes (Desel nad Erwin, 2000;
1.2. Main results. In this paper we propose a trajectory able to reach the sink place of the Petri net. Proceeds at the beginning, the Petri net has always a preset that is, it is possible to show that a process initiated in a Petri net, the computation can always be completed, by the structure of the net, but by operations associated with conditional behavior that will not necessarily satisfy soundness of workflow nets, there exist business processes property. Even if it seems necessary to require the approach showing how the soundness property can be computed over a finite and non-blocking workflow net. We also make emphasis on the reasons why the finite and non-blocking conditions cannot be relaxed. In Section 5 we present two examples which pragmatically illustrate the application of the method. Finally, in Section 6 some concluding remarks and future work are outlined.

1.3. Organization of the paper. The remainder of this paper is organized as follows. We present some of the preliminaries including the mathematical notation and Petri nets basics in Section 2. In Section 3, we motivate the introduction of the soundness workflow verification technique, presenting the basic notion of workflow net and stability followed by the definition of soundness. We also describe and exemplify the finite and non-blocking conditions established for the Petri net. Section 4 outlines the core content of the paper, presenting the basic notions of stability and the main result of the paper about the soundness property. We present a formal approach showing how the soundness property can be computed over a finite and non-blocking workflow net.

2. Preliminaries
In this section, we present some well-established definitions and properties which will be used later (Brams, 1983). Let $\mathbb{N} = \{0,1,2,\ldots\}$, $\mathbb{N}_0^* = \{n_0, n_0 + 1, \ldots, n_0 + k, \ldots\}$, $n_0 \geq 0$, $\mathbb{R} = (-\infty, \infty)$ and $\mathbb{R}_+ = [0, \infty)$.

A (marked) Petri net is a quintuple $PN = (P,Q,F,W,M_0)$, where $P = \{p_1, p_2, \ldots, p_n\}$ 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 which determines a flow relation, $W : F \rightarrow \mathbb{N}_0^*$ is a weight function, $M_0 : P \rightarrow \mathbb{N}$ 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 by $\bullet z := \{y \in P \cup Q \mid (y,z) \in F\}$ the preset of $z$. Analogously, we will denote by $\bullet z = \{y \in P \cup Q \mid (z,y) \in F\}$ 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 that $p \bullet = \emptyset$ (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
\[
W(p,q) = a \text{ or } W(q,p) = b \quad \text{for } a,b \in \mathbb{N}^+_q,
\]
then this is often represented graphically by \( a \) (or \( b \) 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 \( M_k(p_i) \geq W(p_i,q_j) \) for all \( p_i \in P \) such that \( (p_i,q_j) \in F \). 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
\[
M_{k+1}(p_i) = M_k(p_i) + W(q_j,p_i) - W(p_i,q_j).
\]

Let \( A = [a_{ij}] \) denote an \( n \times m \) matrix of integers, called the incidence matrix, where
\[
a_{ij} = a_{ij}^+ - a_{ij}^-
\]
with \( a_{ij}^+ = W(q_j,p_j) \) and \( a_{ij}^- = W(p_j,q_j) \). 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
\[
M_{k+1} = M_k + A^T u_k,
\]
where if at step \( k \), \( a_{ij}^+ < M_k(p_j) \) for all \( p_j \in P \) then \( q_j \in Q \) is enabled, and if this \( q_j \in Q \) fires then its corresponding firing vector \( u_k \) is utilized in the difference equation (3) 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
\[
M' = M + A^T u, \quad u = \sum_{k=0}^{d-1} u_k.
\]

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 (van der Aalst, 2011) if for every two transitions \( q_i, q_j \in Q \), \( q_i \cap q_j \neq \emptyset \) implies \( q_i = q_j \).

Let \( (\mathbb{N}^+_0, d) \) be a metric space where \( d : \mathbb{N}^+_0 \times \mathbb{N}^+_0 \to \mathbb{R}_+ \) is defined by
\[
d(M_1,M_2) = \sum_{i=1}^m \zeta_i | M_1(p_i) - M_2(p_i) |,
\]
where
\[
\zeta_i > 0, \quad i = 1, \ldots, m.
\]

### 3. Motivation

The main point of \( PN \) is its ability to represent mark properties that involve theoretic notions of stability. In this sense, the sink (last place) of \( PN \) is a place whose marking is bounded and it does not change. Therefore, two main concepts must be considered carefully within the notion of stability: cycle and block.

The \( PN \) shown in Fig. 1 represents a cycle. It has the property of stability, because given the incidence matrix
\[
A = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}
\]
and picking the positive vector \( \Phi = [2 \ 2] > 0 \) since \( A \) is already the transpose, we obtain that \( A \Phi^T = [0 \ 0] \leq 0 \) (concluding stability). But the \( PN \) has no final place.

The \( PN \) represented in Fig. 2 represents a block. It has the property of stability, because the incidence matrix
\[
A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 1 \end{bmatrix}
\]
and picking the positive vector such that \( \Phi = [2 \ 1 \ 1 \ 1] > 0 \) since \( A \) is already the transpose, we obtain that \( A\Phi^T = [-1 \ -1 \ -1] \leq 0 \) (concluding stability). But, the sink of the \( PN \) never can be reached.

Loosely speaking, 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, we have the following.

**Definition 1.** 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 (cf. van der Aalst, 1998; 2011).

**Definition 2.** 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 firing sequence leading from state \( M \) to state \( M_o \):

   \[
   \text{for all } M : (M_i \xrightarrow{\sigma} M) \Rightarrow (M \xrightarrow{\sigma} M_o) .
   \]

2. State \( M_o \) is the only state reachable from state \( M_i \) with at least one token in place \( M_o \):

   \[
   \text{for all } M : (M_i \xrightarrow{\sigma} M \land M \geq 0) \Rightarrow (M = M_o) .
   \]

3. There are no dead transitions in \( PN \):

   \[
   \text{for all } q \in Q, \text{ there exist } M, M_i : \quad (M_i \xrightarrow{\sigma} M \xrightarrow{q} M_f) .
   \]

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. Although it is the looked-for soundness of workflow nets, many of the real models with conditional behavior will not satisfy the 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 non-blocking), which is tracked forward, provided 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 3.** Let \( PN \) be a workflow net. \( PN \) is weak 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 \( M_o \):

   \[
   \text{for all } M : (M_i \xrightarrow{\sigma} M) \Rightarrow (M \xrightarrow{\sigma} M_o) .
   \]

2. State \( M(o) \) is the only state reachable from state \( M_i \) with at least one token in place \( M_o \):

   \[
   \text{for all } M : (M_i \xrightarrow{\sigma} M \land M \geq 0) \Rightarrow (M = M_o) .
   \]

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

### 4. Workflow soundness property

Let us consider systems of first ordinary difference equations given by

\[
\begin{align*}
x(n + 1) &= \psi[n, x(n)], \\
x(n_0) &= x_0 .
\end{align*}
\]

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

**Definition 4.** The \( n \)-vector valued function \( \phi(n, n_0, x_0) \) is a solution of \( 5 \) if \( \phi(n_0, n_0, x_0) = x_0 \) and \( \phi(n + 1, n_0, x_0) = \psi(n, \phi(n, n_0, x_0)) \) for all \( n \in \mathbb{N}_+^n \).

**Definition 5.** The system \( 5 \) is said to be practically stable (Lakshmikantham and Martynyuk, 1990; Lakshmikantham et al., 1991) if, given \( (\lambda, \Psi) \) with \( 0 < \lambda < \Psi \), we have that

\[
| x_0 | < \lambda \Rightarrow | x(n, n_0, x_0) | < \Psi , \quad \forall n \in \mathbb{N}_+^n , \quad n_0 \geq 0 .
\]

**Definition 6.** (Lakshmikantham and Martynyuk, 1990; Lakshmikantham et al., 1991) The system \( 5 \) is said to be uniformly practically stable if it is practically stable for every \( n_0 \geq 0 \).

**Definition 7.** A continuous function \( \alpha : [0, \infty) \to [0, \infty) \) is said to belong to the class \( K \) if it is strictly increasing and \( \alpha(0) = 0 \).
Let us consider (Lakshmikantham et al., 1991) the vector function \( v(n, x(n)) : N^n_0 \times \mathbb{R}^n \to \mathbb{R}^n_+ \), and let us define the variation of \( v \) relative to \( \Phi \) by
\[
\Delta v = v(n + 1, x(n + 1)) - v(n, x(n)).
\]
(7)

Then we have the following results due to Lakshmikantham and Martynyuk (1990), Lakshmikantham et al. (1991) as well as Passino et al. (1995).

**Theorem 1.** Let \( v : N^n_0 \times \mathbb{R}^n \to \mathbb{R}^n_+ \) be a continuous function in \( x \), such that for \( \beta, \alpha \in \mathcal{K} \) we have \( \beta(|x|) \leq v(n, x(n)) \leq \alpha(|x|) \) and \( \Delta v(n, x(n)) \leq w(n, v(n, x(n))) \) holds for \( n \in N^n_0 \), \( x(n) \in \mathbb{R}^n \), where \( w : N^n_0 \times \mathbb{R}^n_+ \to \mathbb{R} \) is a continuous function in the second argument. Let us suppose that \( \gamma(n, u) \equiv u + w(n, u) \) is non-decreasing in \( u, 0 < \lambda < \Psi \) are given and finally that \( \alpha(\lambda) < \beta(\Psi) \) is satisfied. Then the stability properties of
\[
u(n + 1) = \gamma(n, u(n)), \quad u(n_0) = u_0 \geq 0
\]
implies the corresponding stability properties of the system \( \Phi \).

We will extend Theorem 1 to the case of several Lyapunov functions. Let us consider a vector Lyapunov function \( v(n, x(n)) : N^n_0 \times \mathbb{R}^n \to \mathbb{R}^n_+ \), and let us define the variation of \( v \) relative to \( \Phi \). Then, we have the following theorem (Lakshmikantham et al., 1991).

**Theorem 2.** Let \( v : N^n_0 \times \mathbb{R}^n \to \mathbb{R}^n_+ \) be a continuous function in \( x \), and define the function \( v_0(n, x(n)) = \sum_{i=1}^{n} v_i(n, x(n)) \) such that it satisfies the estimates
\[
\beta(|x|) \leq v_0(n, x(n)) \leq \alpha(|x|)
\]
(9)
for \( \alpha, \beta \in \mathcal{K} \) and
\[
\Delta v(n, x(n)) \leq w(n, v(n, x(n)))
\]
(10)
for \( n \in N^n_0 \), \( x(n) \in \mathbb{R}^n \), where \( w : N^n_0 \times \mathbb{R}^n_+ \to \mathbb{R}^n \) is a continuous function in the second argument. Assume that \( \gamma(n, u) \equiv qu + w(n, u) \) is non-decreasing in \( u, 0 < \lambda < \Psi \) are given and \( \alpha(\lambda) < \beta(\Psi) \) is satisfied. Then the practical stability properties of
\[
u(n + 1) = \gamma(n, u(n)), \quad u(n_0) = u_0 \geq 0
\]
implies the corresponding practical stability properties of the system \( \Phi \).

Then, we have the following result (Lakshmikantham et al., 1991).

**Corollary 1.** From Theorem 2 we have the following:

1. If \( w(n, e) \equiv 0 \), we obtain uniform practical stability of \( \Phi \), which implies structural stability (Lakshmikantham et al., 1991).

2. If \( w(n, e) = -c(e) \) for \( c \in \mathcal{K} \), then we obtain uniform practical asymptotic stability of \( \Phi \) (cf. Lakshmikantham et al., 1991).

For Petri nets we have the following results of stability (Passino et al., 1995).

**Proposition 1.** Let \( PN \) be a Petri net. Therefore, \( PN \) is uniform practical stable if there exists a \( \Phi \) strictly positive vector such that
\[
\Delta v = u^T A \Phi \leq 0.
\]
(12)
Moreover, \( PN \) is uniform practical asymptotic stability if the following equation holds:
\[
\Delta v = u^T A \Phi \leq -c(e), \quad c \in \mathcal{K}.
\]
(13)
**Proof.** Let us choose as our candidate Lyapunov function \( v(M) = M^T \Phi \) with \( \Phi \) being an \( m \) vector to be chosen. It is simple to verify that \( v \) satisfies all the conditions of Theorem 2. Therefore, the uniform practical asymptotic stability is obtained if there exists a strictly positive vector \( \Phi \) such that Eqn. 13 holds.

**Proposition 2.** Let \( PN \) be a Petri net. Therefore, \( PN \) is uniformly practically stable if there exists a strictly positive \( m \) vector \( \Phi \) such that
\[
\Delta v = u^T A \Phi \leq 0 \iff A \Phi \leq 0.
\]
(14)
**Proof.**

(Necessity) Since \( u^T A \Phi \leq 0 \) holds, for every \( u \) we have that \( A \Phi \leq 0 \).

(Sufficiency) This results from the fact that \( u \) is positive.

**Remark 1.** The "if an only if" relationship of 14 results from the fact that \( u \) is positive.

We have the following theorem that characterizes the soundness property.

**Theorem 3.** Let \( PN \) be a finite and non-blocking workflow net. Then, \( PN \) satisfies the soundness property if there exists a strictly positive \( m \) vector \( \Phi \) such that
\[
\Delta v = u^T A \Phi \leq 0.
\]
**Proof.**

(Necessity) It follows directly from Proposition 1 and Proposition 2.

(Sufficiency) Let us suppose by contradiction that \( u^T A \Phi > 0 \) with \( \Phi \) fixed. From \( M' = M + u^T A \) we have that \( M' \Phi = M \Phi + u^T A \Phi > M \Phi \). Then it is possible to construct an increasing sequence \( M_0 < M_1 < \cdots < M'_n \Phi < \cdots \) which grows up without any bound. Therefore, the \( PN \) is not uniformly practically stable.
Remark 2. The finite and non-blocking conditions over the workflow net cannot be relaxed (see Section 3) and reinforce the definition of the workflow (Definition 3):

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 satisfy the theoretic notion of stability, but it will never reach the sink place of the net.

5. Application examples

The aim of this section is to present application examples represented by a workflow concluding soundness.

Example 1. In the Petri net shown in Fig. 3 only one place is initially marked. Here $t_1$ is enabled and the firing of $t_1$ will result in the state that marks places $p_2$ and $p_4$. In this state, $t_2$, $t_3$ and $t_4$ are enabled. If $t_2$ fires, $t_4$ becomes disabled, but $t_3$ remains enabled. Similarly, if $t_3$ fires, $t_4$ becomes disabled, but $t_2$ remains enabled, etc. The incidence matrix of the workflow net shown in Fig. 3 is given by

$$A = \begin{bmatrix}
-1 & 1 & 0 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 \\
0 & -1 & 1 & -1 & 1 & 0 \\
0 & 0 & -1 & 0 & -1 & 1
\end{bmatrix},$$

and picking the positive vector

$$\Phi = \begin{bmatrix} 4 & 2 & 1 & 1 & 1 \end{bmatrix} > 0$$

since $A$ is already the transpose, we obtain that

$$A\Phi^T = \begin{bmatrix} -1 & -1 & 0 & -1 & -1 \end{bmatrix} \leq 0,$$

concluding soundness (stability).

If we remove transition $t_4$, the resulting net is a free-choice Petri net. These types of Petri nets are interesting from the viewpoint of analysis (van der Aalst, 2011): (i) liveness and boundedness can be decided in polynomial time for free-choice nets (this is not the case for non-free-choice Petri nets) and (ii) they always satisfy the soundness properties.

Example 2. Let us consider an insurance broker agency. As a broker, the agency sells policies for different companies. The main products are life and automobile policies. For selling and advertising, the insurance company obtains detailed information from potential customers, as well as private and governmental agencies. This information is distributed between the company’s agents who contact potential clients via phone and try to set up a conference call. However, they also have their own sources of information. At the interview, the agent examines the client’s current insurance coverage and tries to find an opportunity for a policy that will best fit the customer’s needs.

Before obtaining an insurance policy, the new client must undergo an identity investigation. In the case of a life insurance, the client has, in addition, to approve a physical examination test in an accredited hospital. If the investigation is positive, both the parts sign a policy and keep a copy of the contract. If, during the investigation, irregularities are found, the agent is informed and meets the client in order to find new options.

The insurance policy is in effect when the client makes the first insurance premium payment. Every policy carries with a schedule of premiums, which varies with the type and coverage. Each policy provides a commission for the agency. The commission varies with the insurance company, policy type and coverage. The insurance company management defines the commission policy, which varies from agency to agency. The agency splits the commission received for each policy with the agent who sold it. The rate depends on the seniority of the agent. Once a policy has been sold, the agency submits premium bills to the client, collects payment and sends the payment, minus the commission, to the insurance company.

If a client fails to pay premiums, the agent who sold the policy is informed, so that they can contact the client. Claims can be made on insurance policies as specified in the policy itself. Clients or beneficiaries contact the agent to file such claims. Life insurance claims may be made by the beneficiaries on the death of the insured. In both the cases, the insurance company sends an adjuster to legitimate the claim and arrange the final insurance details. For an automobile insurance policy, claims are made when the car is involved in an accident, damaged or stolen. For simplification, we will consider just the
An analytical method for well-formed workflow/Petri net verification of classical soundness

6. Conclusion and future work

Reasoning about the correctness of a workflow model without any domain knowledge corresponds to the

organizational strategy of the insurance company.

The insurance broker agency business process is represented in Fig. 4 by a free-choice PN. (It is important to note that the PN represented in Fig. 4 is a simplification of the workflow explained in the text description of the broker agency routines.) Now, the incidence matrix $A$ of the workflow net shown in Fig. 4 is given by

$$
\begin{bmatrix}
-1 & 1 & 0 & 1 & 0 \\
0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 \\
0 & 0 & -1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 1 \\
\end{bmatrix},
$$

and picking the positive vector

$$
\Phi = [3 \ 2 \ 1 \ 2 \ 2 \ 2 \ 2 \ 1] > 0, 
$$

since $A$ is already the transpose, we obtain that

$$
A\Phi^T = [0 \ -1 \ -1 \ -1 \ 0 \ 0 \ 0 \ -1] \leq 0, \n$$

concluding soundness (stability).

It is important to note that the key contribution of the paper is the analytical method itself. The definition of soundness is introduced because the proposed method only satisfies part of the soundness property (van der Aalst, 2011). In this sense, the proposed analytical method is a step forward in checking the soundness of workflow nets.

Without doubt there are more than a few theoretical challenges that need to be considered in future research in Lyapunov-based theory for solving the soundness verification problem. This paper has interesting implications for using more sophisticated definitions of Petri nets, because the Lyapunov method introduces new concepts in the Petri nets area. In this work, we consider dynamical systems governed by ordinary difference equations described by Petri nets. Then, an important emerging open research challenge is the use of Lyapunov theory to produce a trajectory tracking function (Lyapunov-like function) as a solution to the difference equation (constructed with respect the constraints imposed by the system). Then the Lyapunov-like function will compute the trajectory of the token over the Petri net, converging naturally into the sink place (Clempner, 2005).
References


An analytical method for well-formed workflow/Petri net verification of classical soundness


Julio Clempner has more than ten years of experience in the field of management consulting. Currently, he specializes in the application of high technology. He has also conducted projects related to IT government, project management, analysis and design of software (RUP, UML), development of software (add-hoc and products), information technology strategic planning, balanced scorecard, evaluation of software and business process reengineering (SIX-SIGMA) for infusing advanced computer technologies into diverse lines of businesses. Dr Julio Clempner is now involved in providing e-business solutions, including e-business strategy, management of technology, operations strategy, project management, continuous quality improvement, and managing total quality transformation. Dr. Clempner’s research interests are focused on justifying and introducing the Lyapunov equilibrium point in shortest-path decision processes and shortest-path game theory, which has led to several streams of research. Dr. Julio Clempner is a member of the Mexican National System of Researchers (SNI).

Received: 4 July 2013
Revised: 30 November 2013
Re-revised: 24 February 2014