A Petri Net Based Approach for Modelling and Analyzing Interorganizational Workflows with Dynamic Structure

Oana Otilia Prisecaru

Faculty of Computer Science, ”Al. I. Cuza” University
Gen. Berthlot St, No 16, 740083 Iasi, Romania

Abstract. Interorganizational workflows represent a special type of workflows that involve more than one organization. In this paper, an interorganizational workflow will be modelled using a special class of nested Petri nets, dynamic interorganizational workflow nets (DIWF-nets). DIWF-nets can model interorganizational workflows in which some of the local workflows can be removed, during the execution of the workflow, due to exceptional situations. Our approach permits a clear distinction between the component workflows and the communication structure. The paper defines a notion of behavioural correctness (soundness) and proves this property is decidable for DIWF-nets.

1 Introduction

A workflow is an operational description of a business process that takes place inside one organization. Due to the rise of virtual organizations, electronic commerce and international companies, many existent business processes involve more than one organization. These workflows are referred to as *interorganizational workflows*. There have been developed several specification languages for interorganizational workflows, based on XML and Web services: WSFL, BPEL, BPEL4Chor, XLANG, WSCDL, etc [6]. These languages lack formal semantics and analytical power. In order to solve these problems, several formalisms have been proposed for specifying interorganizational workflows: Communicating Finite Automata, Category theory, Process algebra and Petri nets. Petri nets represent a well-known formal method, successfully used as a modelling technique for workflows see [1], due to their graphical representation, their formal semantics and expressiveness. Petri nets have also been used for modelling interorganizational workflows [2, 4, 3, 8, 5]. In the existing approaches, there is not a clear distinction between the component workflows and the communication structure, which makes the models difficult to understand and work with. Also, the structure of the interorganizational workflow is considered to be static (i.e. the number of component workflows involved is fixed), but this does not always happen in real situations.

This paper presents a new approach on the modelling of interorganizational workflows, based on nested Petri nets. Nested Petri nets [10] are a special class of the Petri net model, in which tokens may be Petri nets (object-nets). The paper deals with

Otilia Prisecaru O. (2009).
A Petri Net Based Approach for Modelling and Analyzing Interorganizational Workflows with Dynamic Structure.
In *Proceedings of the 7th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, pages 23-32
DOI: 10.5220/0002200700230032

loosely coupled interorganizational workflows: the component workflows behave independently, but need to interact in order to accomplish a global business goal. The interaction is made through asynchronous or synchronous communication. *Dynamic interorganizational workflow nets (DIWF-nets)* are introduced as a special case of nested Petri nets, in which every local workflow is modelled as a distinct object-net. For the modelling of a local workflow we use *extended workflow nets*, a version of the workflow nets introduced in [1]. The communication mechanisms between the local workflows are also described using an object-net. Thus, our approach offers a modular view over the components of an interorganizational workflow. In our model the structure of the interorganizational workflow can change during its execution, as the local workflows can be dynamically removed at certain points. The paper introduces a notion of behavioural correctness for DIWF-nets, *soundness*, and proves this property is decidable.

In what follows we will give the basic terminology and notation concerning workflow nets, a Petri net formalism which has been used for the modelling of workflows [1]. We assume the reader is familiar with the Petri net terminology and notation details can be found in [12].

A workflow net (WF-net) is a Petri net with two special places: a source place, i, and a sink place, o. In a WF-net, all places and transitions should be on a path from i to o. The two conditions are expressed formally as follows:

A Petri net PN=(P,T,F) is a WF-net iff: (1) PN has a source place i and a sink place o such that $^\bullet i = \emptyset$ *and* $o^\bullet = \emptyset$. *(2) If we add a new transition* t^* *to PN such that* $^\bullet t^* = \{o\}$ *and* $t^{*\bullet} = \{i\}$*, then the resulted Petri net is strongly connected.*

A marking of a WF-net is a multiset $m : P \rightarrow \mathbb{N}$ (where $\mathbb{N}$ denotes the set of natural numbers). We write $m = 1'p_1 + 2'p_2$ for a marking m with $m(p_1) = 1, m(p_2) = 2$ and $m(p) = 0, \forall p \in P - \{p_1, p_2\}$. The marking $1'i$ represents the initial marking of the net, and it is also denoted by i. The marking $1'o$, represents the end of the workflow process (and the final marking of the net, denoted by o).

The rest of the paper is organized as follows: Section 2 presents an introductory example of a DIWF-net, Section 3 introduces DIWF-nets, Section 4 defines and studies the soundness property for DIWF-nets, Section 5 presents some of the related work and Section 6 presents the concluding remarks.

2 Dynamic Interorganizational Workflow Nets: An Introductory Example

In this section we present an example of a DIWF-net, modelling an interorganizational workflow which consists of two workflows. The workflows are modelled by two extended workflow nets, WF_1' and WF_2' (see Fig. 1(a)). These nets are WF-nets, extended with special transitions: *exit* in WF_1' terminates abnormally the workflow execution. t_1' and t_2' empty the sink places of the two WF-nets.

The two workflows interact as follows: task t_1 in WF_1' must fire before t_4 in WF_2' (i.e. there is an asynchronous communication between the two workflows) and task t_2 in WF_1' and t_5 in WF_2' must fire synchronously (i.e. there is a synchronous communication between the local workflows, through these transitions).

The asynchronous communication is described using a partial order on tasks: $AC =$

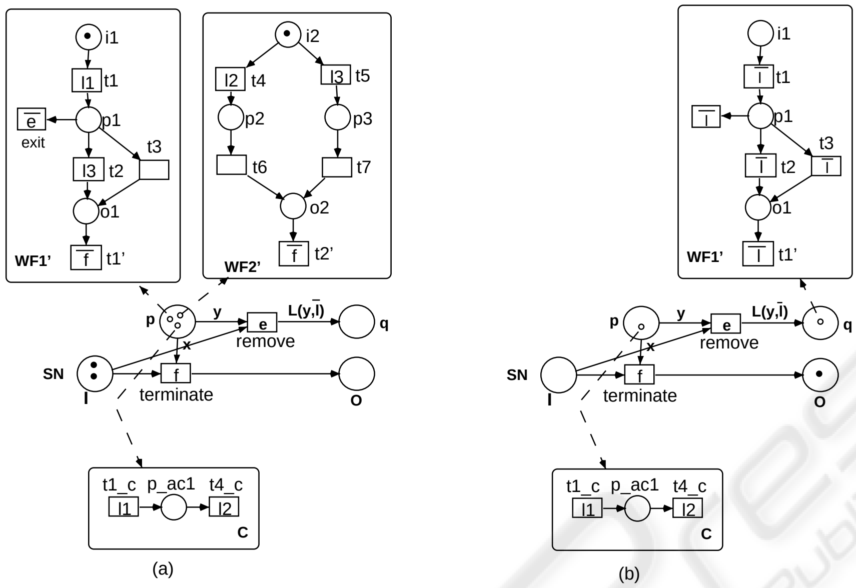

Fig. 1. An example of a DIWF-net: (a) in its initial marking and (b) in a final marking.

$\{(t_1, t_4)\}$. The synchronous communication is specified using the set of synchronous communication elements: $SC = \{\{t_2, t_5\}\}$. The DIWF-net is a nested Petri net which consists of a system net, SN and of three object-nets, WF_1', WF_2' and C. The initial marking of the DIWF-net is depicted in Fig.1(a). The object-net C describes the asynchronous communication between the local workflows. The set of places is $P_C = \{p_{ac_1}\}$, where $ac_1 = (t_1, t_4)$. The transitions in T_C correspond to the transitions involved in AC: $T_C = \{t_{1c}, t_{4c}\}$. The initial marking of C is 0.Some of the transitions of the DIWF-net are labelled (using a partial function, Λ). The transitions involved in AC and their corresponding transitions from C will be assigned the same labels: $\Lambda(t_1) = \Lambda(t_{1c}) = l_1$ and $\Lambda(t_4) = \Lambda(t_{4c}) = l_2$. The transitions which appear in SC will be assigned the same label: $\Lambda(t_2) = \Lambda(t_5) = l_3$. We write a marking of a DIWF-net as a vector $M = (M(I), M(p), M(q), M(O))$.In DIWF-nets, there are several firing rules: an unlabelled transition from an object-net can fire if the transition is enabled in the object-net (an object-autonomous step). Also, if all the transitions with the same label, from object-nets residing in the same place of SN, are enabled in those object-nets, then they should fire synchronously (a horizontal synchronization step). Finally, a labelled transition enabled in SN should fire simultaneously with the transitions from the object-nets ”involved” in this firing, which have a complementary label (this is a vertical synchronization step).

In the example in Fig. 1(a), t_1 is enabled in (WF_1', i_1) and t_4 and t_5 are enabled in (WF_2', i_2). But t_4 should fire at the same time with t_{4c} in the object-net C. Since t_{4c} is not enabled in $(C, 0)$, t_4 cannot fire yet. Thus, t_1 always fires before t_4, as specified by AC. Also, t_5 should fire at the same time with t_2 from WF_1'. Since t_1 is enabled in (WF_1', i_1) and t_{1c} is enabled in $(C, 0)$, then the horizontal synchronization step denoted by $(;t_1, t_{1c})$ is enabled in marking M_0. The resulting marking is denoted as a tuple: $M_1 = (2, \{(WF_1', m1_1), (WF_2', i_2), (C, mc_1)\}, 0, 0)$. If we bind the variable y to the net-token $(WF_1', m1_1)$ from p, the transition $remove$ from SN is enabled in M_1 with this binding (i.e. the firing of this transition can remove the net-token $(WF_1', m1_1)$ from place p). $remove$ from SN should fire synchronously with $exit$ from WF_1' (which is labelled by $\overline{e}$). The simultaneous firing of $exit$ and $remove$ is a vertical synchronization step, denoted by $(remove; exit)$. If this step fires, then the first workflow is removed. The resulted marking is $M_2 = (1, \{(WF_2, i_2), (C, mc_1)\}, \{(WF_1', m1_1')\}, 0)$, and all the transitions in WF_1' in q are labelled by $\overline{l}$. If we consider the firing of the sequence of steps (t_4, t_{4c}), (t_6), it results the marking $M_4 = (1, \{(WF_2', o_2), (C, 0)\}, \{(WF_1', m1_1')\}, 0)$. The vertical synchronization step $(terminate; t_2')$ is enabled in M_4 (if we bind x to (WF_2', o_2)). The resulting marking is $M_5 = (0, \{(C, 0)\}, \{(WF_1', m1_1')\}, 1)$ (Fig. 1(b)). The transitions in WF_1' are all re-labelled with a label $\overline{l}$, which prevents them from firing.

3 Definition of Dynamic Interorganizational Workflow Nets

In what follows, we will assume there are n local workflows which behave independently, but need to interact at certain points using asynchronous communication (which corresponds to the exchange of messages) and synchronous communication (which forces the local workflows to execute specific tasks at the same time). We will consider the situation in which a local workflow can interrupt its normal execution at a certain point, due to the occurrence of an error. At this point, the workflow will be removed from the interorganizational workflow. We will assume that at least one workflow, whose executions is critical, cannot interrupt abnormally its execution.

In order to model a workflow which can terminate abnormally its execution, we define *extended workflow nets* (extended WF-nets), an extension of the WF-nets defined in [1]. These Petri nets are WF-nets which can be extended with two transitions: one of the transitions empties the sink place of the workflow, while the other transition interrupts the normal execution of the workflow (this transition is optional).

Definition 1. *Let $WF = (P, T, F)$ be a WF-net. The extended WF-net is $WF' = (P, T', F')$, where:*
- $T' = T \cup \{t'\} \cup T_e$, $T_e \subseteq \{\text{exit}\}$ such that, if $\text{exit} \in T'$ then ${}^\bullet\text{exit} \neq \emptyset$.
- $F' = F \cup \{(o, t')\} \cup F_e$, where $F_e \subseteq P \times \{\text{exit}\}$ ($F_e = \emptyset$, if $\text{exit} \notin T'$)

WF is called the underlying net of WF'.
Dynamic interorganizational workflow nets (DIWF-nets) are defined as nested nets with a particular structure, extended with two sets (AC and SC), used for describing the communication between the local workflows, and a special labelling system. We also use a special expression, $L(y, \overline{l})$, for labelling an arc of SN.

Definition 2. *A dynamic interorganizational workflow net DIWF is a nested Petri net: $DIWF = (Var, Lab, (WF_1', i_1), \ldots, (WF_n', i_n), AC, SC, (C, 0), SN, \Lambda)$ such that:*

1. *$Var = \{x, y\}$ is a set of variables.*
2. *$Lab = Lab_{AC} \cup Lab_{SC} \cup \{e, \overline{e}, f, \overline{f}\}$ is a set of labels.*

3. *$(WF'_1, i_1), \ldots, (WF'_n, i_n)$ are extended WF-nets, with the corresponding initial markings $i_1, i_2, \ldots, i_n$.*
4. *AC is the asynchronous communication relation: $AC \subseteq T^\circ \times T^\circ$, where $T^\circ = \cup_{k\in\{1,\ldots,n\}} T_k$, T_k is the set of transitions from the underlying WF-net of WF'_k. If $(t, t') \in AC, t \in T_i, t' \in T_j$, then $i \neq j$.*
5. *SC is the set of synchronous communication elements: $SC \subseteq P(T^\circ)$ and:*
   - $\forall u, v \in SC : u \cap v = \emptyset$.
   - if $t \in T_i, t' \in T_j, t, t' \in u, u \in SC$, then $i \neq j$.
6. *$C = (P_C, T_C, F_C)$ is the communication object:*
   - $P_C = \{p_{ac} | ac \in AC\}$.
   - $T_C = \{t_c | \exists t \in T^\circ : (t', t) \in AC \vee (t, t') \in AC\}$.
   - $F_C = \{(p_{ac}, t) \in P_C \times T^\circ | ac = (t', t) \in AC\} \cup \{(t, p_{ac}) \in T^\circ \times P_C | ac = (t, t') \in AC\}$
7. *$SN = (N, W, M_0)$ is the system net of DIWF, such that:*
   - $N = (P_N, T_N, F_N)$ is a high level Petri net: $P_N = \{I, p, q, O\}$, where O is a place such that $O^\bullet = \emptyset$ and I is a place such that ${}^\bullet I = \emptyset$; $T_N = \{terminate, remove\}$; $F_N = \{(I, terminate), (p, terminate),$
   $(terminate, O), (p, remove), (remove, q), (I, remove)\}$.
   - W is the arc labelling function: $W((p, terminate)) = x$, $W((p, remove)) = y$,
   $W((remove, q)) = L(y, \overline{l})$ and $W(a) = 1$ for the rest of the arcs.
   - M_0 is the initial marking of the net: $M_0(I) = n$, $M_0(p) = \{(WF'_1, i_1), \ldots,$
   $(WF'_n, i_n), (C, 0)\}$, $M_0(q) = 0$ and $M_0(O) = 0$
   . - Λ is a partial labelling function such that:
   - $\forall u \in SC, \forall t, t' \in u, \Lambda(t) = \Lambda(t') = l, l \in Lab_{SC}$.
   - if $t \in T^\circ$ such that $(t, t') \in AC$ or $(t', t) \in AC$, then there exists $t_c \in T_C : \Lambda(t_c) = \Lambda(t) = l, l \in Lab_{AC}$.
   - $\Lambda(t'_i) = \overline{f}, \forall i \in \{1, \ldots n\}$ and $\Lambda(terminate) = f$.
   - $\Lambda(remove) = e$ and, if $exit_i \in T'_i$, then $\Lambda(exit_i) = \overline{e}$ ($i \in \{1, \ldots n\}$).
   - $\forall t, t' \in T_i (i \in \{1, \ldots, n\}) : \Lambda(t) \neq \Lambda(t')$.

In a DIWF-net there are n object-nets (extended WF-nets) representing the local workflows. We denote by t'_i the transition which empties the output place o_i in an extended WF-net WF_i. Var is the set of variables in the net. Variables x and y will take as value an object-net in a certain marking. Lab is a set of labels: the labels in Lab_{AC} are used for the elements of AC and the labels from Lab_{SC} are used for the elements of SC. Lab_{AC} and Lab_{SC} are not necessary disjoint. The label $\overline{f}$ is used for labelling the transition t'_i from WF'_i, $\forall i \in \{1, \ldots, n\}$. AC represents the asynchronous communication relation: if $(t, t') \in AC$, then, the transition t must execute before the transition t'. SC represents the set of synchronous communication elements: if $u \in SC$, then, all the transitions from u have to execute at the same time. C is an object-net which describes the asynchronous communication: if $ac = (t, t') \in AC$, then there is a corresponding place p_{ac} in P_C. For every transition $t \in T^\circ$ involved in an element of AC, there is a transition $t_c \in T_C$. Also, if $ac = (t, t') \in AC$, then there exist two arcs $(t_c, p_{ac}), (p_{ac}, t'_c) \in F_C$. In DIWF-nets, the expressions on arcs can be either variables (x or y), the constant 1 or the function $L(y, \overline{l})$. Λ is a partial function which labels transitions of the DIWF-net. If $u \in SC$, then all the transitions from u have the same label $l \in Lab_{SC}$. For every transition t involved in an asynchronous communication element, there is a transition t_c in the object-net C and $\Lambda(t) = \Lambda(t_c) = l, l \in Lab_{AC}$.

We denote by A_{net} the net tokens of the DIWF-net: $A_{net} = \{(EN, m)$ / m is

a marking of EN, $EN \in \{WF'_1, \ldots, WF'_n, C\}\}$. L is a function such that $L : A_{net} \times Lab_v \to A_{net}$, which relabels all the transitions of $(EN, m) \in A_{net}$ with $l \in Lab_v$.
A marking M of a DIWF-net is a function such that: $M(I) \in \mathbb{N}$, $M(O) \in \mathbb{N}$ and $M(p), M(q) \subseteq A_{net}$. We write M as a vector $M = (M(I), M(p), M(q), M(O))$.

If $t \in T_{SN}$, we denote by $Var(t)$ the set of variables which appear in the expressions from the arcs adjacent to t. A *binding* (of a transition $t \in T_{SN}$) is a function $b : Var(t) \to A_{net}$. We have that $b(L(y, \bar{l})) = L(b(y), \bar{l})$.

In a DIWF-net, a transition *t* from SN is enabled in a marking M w.r.t. a binding b iff: (1) $W(p, t)(b) \subseteq M(p)$ (where $W(p, t)(b)$ is the arc expression of the arc (p, t) evaluated in binding b) and (2) $1 \leq M(I)$.

There are several types of steps, defining the behaviour of nested Petri nets see [10]. In the case of DIWF-nets, there are two vertical synchronization steps:
-If transition *terminate* is enabled in a marking M w.r.t. a binding b and the transition t'_i is enabled in the object-net $b(x) = (WF'_i, m_i)$, $(WF'_i, m_i) \in M(p)$, then the simultaneous firing of *terminate* and t'_i is a vertical synchronization step, denoted by $(terminate[b]; t'_i)$. The firing of $(terminate[b]; t'_i)$ removes the object-net (WF'_i, m_i) from p and an atomic token from I and adds one atomic token to place O.
-If transition *remove* is enabled in a marking M w.r.t. a binding b and the transition $exit_i$ $(\Lambda(exit_i) = \bar{e})$ is enabled in the object-net $b(y) = (WF'_i, m_i)$, $(WF'_i, m_i) \in M(p)$, then the simultaneous firing of *remove* and $exit_i$ is a vertical synchronization step. The firing of $(remove[b]; exit_i)$ removes the net-token (WF'_i, m_i) from p and adds the net-token $b(L(WF'_i, m_i), \bar{l}) = (\overline{WF'_i}, m_i)$ to the place q, where $\overline{WF'_i}$ is obtained from WF'_i by labelling all the transitions with the label $\bar{l}$. We also write WF'_i instead of $\overline{WF'_i}$ ($\overline{WF'_i}$ only appears in place q).

The definition of the horizontal synchronization step is different from the one in [10], allowing the synchronization of arbitrarily many transitions from several object-nets. This change does not affect the general properties of nested nets:

Let M be a marking of $DIWF$ and $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ the set of net-tokens from p ($k \leq n + 1$). Assume $t_1, \ldots t_s \in T^\circ$ is the set of all the transitions with the same label $l \neq \bar{e}, \bar{f}$, $\Lambda(t_1) = \Lambda(t_2) = \ldots = \Lambda(t_s) = l$, such that: every transition t_j ($j \in \{1, \ldots, s\}$) is enabled in a net-token $\alpha_{k_j} = (EN_j, m_j) \in M(p)$ and $m_j[t_j\rangle m'_j$. The synchronous firing of $t_1, \ldots, t_s$ is called an horizontal synchronization step. The resulting marking, M', is obtained from M by replacing the set $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ from place p with $\{\alpha'_1, \alpha'_2, \ldots, \alpha'_k\}$, where $\alpha'_{k_j} = (EN_j, m'_j), \forall j \in \{1, \ldots, s\}$ and $\alpha'_i = \alpha_i, \forall i \in \{1, \ldots, k\} \setminus \{k_1, \ldots k_s\}$. We write: $M[; t_1, \ldots, t_s\rangle M'$.

4 The Soundness Property for Dynamic Interorganizational Workflow Nets

In this section we will introduce a notion of soundness for DIWF-nets. In order to prove the decidability of soundness we will use some results regarding well-structured transitions systems [7].
A quasi ordering is any reflexive and transitive relation $\leq$. We let $x < y$ denote $x \leq y \not\leq x$. A partial ordering is an asymmetric quasi-ordering. A *well-quasi-ordering*

is any quasi-ordering $\leq$ (over some set X) such that, for any infinite sequence $x_0, x_1, \ldots$ in X, there exists indexes $i < j$ such that $x_i \leq x_j$.

A transition system (TS) is a structure $TS = \langle S, \rightarrow \rangle$ such that S is a set of states and $\rightarrow \subseteq S \times S$ is a transition relation. If $s \in S$, $Succ^*(s) = \{s' \in S | s \xrightarrow{*} s'\}$.

A *well-structured transition system* is a transition system $WSTS = \langle S, \rightarrow, \leq \rangle$ such that: $\leq \subseteq S \times S$ is a well-quasi-ordering and $\leq$ is (upward) compatible with $\rightarrow$, i.e. for all $s_1, t_1, s_2 \in S$ with $s_1 \leq t_1$ and $s_1 \rightarrow s_2$, there exists a sequence $t_1 \xrightarrow{*} t_2$ such that $s_2 \leq t_2$. $WSTS$ has *strict compatibility* iff for all $s_1 < t_1$ and $s_1 \rightarrow s_2$, there exists a sequence $t_1 \xrightarrow{*} t_2$ with $s_2 < t_2$.

A WSTS is bounded from s if $Succ^*(s)$ is finite. It was proven in [7] that boundness is decidable for WSTS's with strict compatibility.

A notion of soundness was defined for WF-nets, expressing the minimal conditions a correct workflow should satisfy [1]: a workflow must always be able to terminate $((\forall m)((i[*\rangle m) \Longrightarrow (m[*\rangle o)))$, the worklfow must terminate correctly $(\forall m)((i[*\rangle m) \wedge m \geq o) \Longrightarrow (m = o))$, and there do not exist dead tasks $(\forall t \in T)(\exists m, m')(i[*\rangle m[t\rangle m')$.

It was proven see [1] that the soundness property is decidable for WF-nets.

Definition 3. *Let WF' be an extended workflow net and WF its underlying WF-net. WF' is sound if: (1) WF is sound and (2) if* exit $\in T'$*, then transition* exit *is not dead.*

In an interorganizational workflow, although the local workflows are sound, we can have synchronization errors and deadlocks. A correct interorganizational workflow should satisfy the following conditions: every local workflow should be sound; for any reachable marking M in $DIWF$, even if some local workflows have been removed, there is an execution sequence from M such that the remaining workflows will still be able to terminate correctly their execution. We will aslo require that the component workflows should not be allowed to send an infinite number of messages to the other workflows and that the $DIWF$ should be quasi-live (i.e. every step can fire in a certain reachable marking).

If M is a marking in a DIWF-net, a final marking corresponding to M is a marking $(0, \{(C, m)\}, M(q), k)$ ($k = |M(p)| - 1$, $n = M_0(I)$). In such a marking, k is the number of workflows which terminated correctly their execution. All the atomic tokens have been removed from I (by firing $remove$ and $terminate$). We denote the set of final markings corresponding to M by $\mathcal{M}_f(M)$. $\mathcal{Y}$ will denote the set of steps in a DIWF-net.$[M\rangle$ denotes the set of markings reachable from M.

We can define formally the notion of soundness for a DIWF-net as follows:

Definition 4. *A DIWF-net $DIWF$ is sound if and only if:*

1. (WF'_j, i_j) *is a sound extended workflow net,* $\forall j \in \{1, \ldots, n\}$.
2. $DIWF$ *is quasi-live:* $(\forall Y \in \mathcal{Y})$ $(\exists M \in [M_0\rangle : M[Y\rangle)$.
3. *For every marking M reachable from the initial marking M_0, there exists a firing sequence leading from M to a final marking M_f:*
   $(\forall M)((M_0[*\rangle M) \Longrightarrow (M[*\rangle M_f, M_f \in \mathcal{M}_f(M))$.
4. *The communication net is bounded:* $\forall M \in [M_0\rangle$, $(C, m) \in M(p)$, *then* $\exists n \in \mathbb{N} : m(p_{ac}) \leq n, \forall p_{ac} \in P_C$.

A partial order on the markings of nested Petri nets was defined in [10]. In the case of DIWF-nets we have that $M_1 \preceq M_2$ if and only if $M_1(I) \leq M_2(I)$, $M_1(O) \leq M_2(O)$ and there exists an embedding $J_p : M_1(s) \to M_2(s)$ ($s \in \{p, q\}$), such that for any $\alpha_k \in M_1(p)$ ($k \leq n+1$), $J_p(\alpha_k) = \alpha'_k$ such that: either $\alpha_k = \alpha'_k$ or $\alpha_k = (EN, m)$ and $\alpha'_k = (EN, m')$ ($EN \in \{WF'_1, \ldots, WF'_n, C\}$) and $m \leq m'$.

One can notice that in a DIWF-net, for any reachable marking $M \in [M_0\rangle$, it holds: (1) $M_0(I) + 1 = |M(q)| + M(O) + |M(p)|$ and (2) $M(I) + 1 = |M(p)|$. Using these observations, the following lemma can be easily proven (we omit the proof here):

Lemma 1. *Let $DIWF$ be a DIWF-net and the extended WF-nets WF'_j are sound, for all $j \in \{1, \ldots, n\}$. Assume $M_1, M_2 \in [M_0\rangle$ such that $M_2 \succ M_1$. Then: (1) $M_1(I) = M_2(I)$, $|M_1(q)| = |M_2(q)|$, $M_1(O) = M_2(O)$ and (2) for every $(WF'_j, m_j) \in M_1(s)$, $(WF'_j, m'_j) \in M_2(s)$ ($s \in \{p, q\}$) and $m'_j \geq m_j$. If $M_2 \in [M_1\rangle$, then $m' > m$.*

A DIWF-net is bounded if $[M_0\rangle$ is finite. We will prove that, in the case that all the component WF-nets are sound, boundness is decidable for DIWF-nets.

Theorem 1. *Let $DIWF$ be a DIWF-net such that all the component extended WF-nets are sound. Then, $WSTS = \langle [M_0\rangle, [\rangle, \preceq \rangle$ is a well-structured transition system with strict compatibility.*

Proof. Assume $M_1, M_2 \in [M_0\rangle$, $M_2 \succ M_1$. If $Y \in \mathcal{Y}$ such that $M_1[Y\rangle M'_1$, we will prove that $M_2[Y\rangle M'_2$ and $M'_2 \succ M'_1$.

If Y is an object-autonomous step, an horizontal synchronization step or $Y = (remove; exit_j)$, the proof uses the fact that the order $\leq$ on the markings of an ordinary Petri net is strictly monotonic.

If $Y = (terminate, t'_j)$ is enabled in M_1. t_j is enabled in a net-token $(WF'_j, m_j) \in M_1(p)$. We will show that $m_j = m'_j$. Because $M_1 \prec M_2$, for every $(WF'_i, m_i) \in M_1(r)$ and $(WF'_i, m'_i) \in M_2(r)$ ($r \in \{p, q\}$), $m'_i \geq m_i$. Also, $m' \geq m$. At least one of these inequalities is strict. We also have $M_1(I) = M_2(I)$, $|M_1(q)| = |M_2(q)|$, $M_1(O) = M_2(O)$. In $M_2(p)$, there is a net-token (WF'_j, m'_j) such that $m'_j \geq m_j$. Hence, t'_j is also enabled in (WF'_j, m'_j), and the step Y is enabled in M_2. Because $M_1, M_2 \in [M_0\rangle$, then m_j, m'_j are reachable markings in WF'_j. Because t_j is enabled in m_j and m'_j, then $m_j(o_j) \geq 1$ and $m'_j(o_j) \geq 1$. But WF'_j is sound, hence the only reachable marking which contains a token in the place o_j is the final marking, o_j. So, $m_j = m'_j = o_j$. Because $M_1 \prec M_2$, either there exists at least a net token $(WF'_s, m_s) \in M_1(r)$ such that $(WF'_s, m'_s) \in M_2(r)$ ($r \in \{p, q\}$) and $m'_s > m_s$ (with $s \neq j$), or $m > m'$. $M_1[Y\rangle M'_1$ and M'_1 and M_1 differ only in the marking of p and O: $M'_1(p) = M_1(p) \setminus \{(WF_j, m_j)\}$, $M'_1(O) = M_1(O) + 1$. $M_2[Y\rangle M'_2$ and M'_2 and M_2 differ only in the marking of p and O: $M'_2(p) = M_2(p) \setminus \{(WF_j, m_j)\}$, $M'_2(O) = M_2(O) + 1$. Because there exists $s \neq j$ such that $m'_s > m_s$, or $m > m'$, it results that $M'_2 \succ M'_1$.

Consequence 1 *Boundness is decidable for DIWF-nets, if all the component WF-nets are sound.*

Theorem 2. *Assume $DIWF$ is a DIWF-net, such that all the component extended WF-nets are sound. Then, $DIWF$ is bounded if and only if $\exists n \in N$ such that $\forall M \in [M_0\rangle$, $(C, m) \in M(p), \forall p_{ac} \in P_C$: $m(p_{ac}) \leq n$.*

Proof. $(\Rightarrow)$ If $DIWF$ is bounded, then the places of any object-net are bounded, in any reachable marking of $DIWF$. $(\Leftarrow)$ If we assume that $DIWF$ is unbounded, using lemma 1 (2), we can obtain an infinite number of reachable markings for C. Hence, there is a place of C with an infinite number of tokens. Contradiction.

Theorem 3. *Soundness is decidable for DIWF-nets.*

Proof. The condition (1) in the definition of soundness is decidable, because soundness is decidable for extended WF-nets and the number of extended WF-nets is finite. Condition (2) is decidable, because the coverability problem is decidable in nested Petri nets [9] and the quasi-liveness is equivalent to the coverability problem. If we assume that all the extended WF-nets are sound, the boundness problem is decidable. If $DIWF$ is unbounded, it results that the last condition in the definition of soundness does not hold and thus the DIWF-net is not sound. If $DIWF$ is bounded, the last condition in the definition of soundness holds. It also results that $[M_0\rangle$ is finite and the reachability problem is decidable. Thus, the third condition in the definition of soundness is also decidable: given a reachable marking M, we can decide whether a marking $M_f \in \mathcal{M}_f(M)$ is reachable from M ($\mathcal{M}_f$ is a finite set if $DIWF$ is bounded).

5 Related Work

BPEL4Chor is a choreography language based on BPEL which allows the specification of interorganizational workflows. [8] proposes a translation from BPEL4Chor to Open Workflow Nets, in order to allow the verification of BPEL4Chor. This approach does not take into consideration the situation in which the component workflows are dynamically removed. In IOWF-nets defined in [2], the component workflows are all represented into the same ”flat” Petri net and the structure of the interorganizational workflow is fixed. [3] proposes a method of designing correct interorganizational workflows in a top-down way: first a contract is used to specify the way the workflows interact. Then the private component workflows are build such that each workflow accords with the contract and the overall interorganizational workflow terminates properly. A similar approach is used in [4], where a shared public workflow-net is used for the specification of the communication structure. A notion of projection inheritance is used for the private workflows, instead of the notion of accordance from [3]. The approaches in [4, 3] ensure the privacy of the workflows and offer a modular view over the interorganizational workflow, but they work with a fixed number of component workflows and they do not offer a model for executing the interorganizational workflow. The approach in [5] uses nets in nets for modelling workflows and interorganizational workflows focusing on the concept of mobility and on the notion of inheritance. This approach does not define a notion of behavioural correctness for interorganizational workflows. In [11], we proposed IWF-nets for modelling interorganizational workflows in a modular way. In this paper we extended that approach, which only considered a fixed structure of the interorganizational workflow.

6 Conclusions

In this paper we introduced a new approach on the modelling of interorganizational workflows, based on nested Petri nets. Our approach offers a modular view on the interorganizational workflow, because the local workflows and the communication structure are distinct elements in DIWF-nets; steps in DIWF- nets can easily express the synchronous and the asynchronous communication; our approach permits the modelling of a situation which can often occur in practice: some local workflows can be dynamically removed from the interorganizational workflow during its execution. A notion of soundness was introduced for DIWF-nets and we proved this property is decidable for DIWF-nets. Future work aims to extend DIWF-nets in order to allow the dynamic creation of workflows and also to define and study the soundness property for this extension.

References

1. W. M. P. van der Aalst. *The Application of Petri nets to Workflow Management*. The journal of Circuits, Systems and Computers, 8(1): pp. 21- 66, Eindhoven University of Technology, The Netherlands, 1998.
2. W.M.P. van der Aalst. *Loosely coupled interorganizational workflows: Modeling and analyzing workflows crossing organizational boundaries*. Information and Management, vol.37, no.2, pp.6775, 2000.
3. W. M. P. van der Aalst, N. Lohmann, P. Massuthe, C. Stahl, and K. Wolf. *From Public Views to Private Views – Correctness-by-Design for Services*. In Proc. of WS-FM 2007, Brisbane, Australia, September 28-29, 2007,vol. 4937 of LNCS, pp. 139-153, 2008. Springer-Verlag.
4. W.M.P. van der Aalst and M. Weske. *The P2P approach to Interorganizational Workflows* Proc. of CAiSE'01, vol. 2068 of LNCS, pp. 140156. Springer-Verlag, Berlin, 2001.
5. W.M.P. van der Aalst, D. Moldt, R. Valk and F. Wienberg. *Enacting Interorganizational Workflows Using Nets in Nets*. In Proc. of Workflow Management '99, pp. 117-136, 1999.
6. G. Kramler , W. Retschitzegger. *Specification of Interorganizational Workflows - A Comparison of Approaches*. Institute of Software Technology and Interactive Systems, Business Informatics Group, Vienna University of Technology, 2002. Technical Report, 08/02.
7. A. Finkel and Ph. Schnoebelen. *Well-Structured Transition Systems Everywhere!*. Theoretical Computer Science 256(1-2), pp. 63-92, 2001.
8. N. Lohmann, O. Kopp, F. Leymann, and W. Reisig. *Analyzing BPEL4Chor: Verification and Participant Synthesis*. Proc. of WS-FM 2007 Brisbane, Australia, September 28-29, 2007, vol. 4937 of LNCS, pp. 46-60, 2008. Springer-Verlag.
9. I.A. Lomazova: *Nested Petri nets: modeling and analysis of distributed systems with object structure*. Nauchny Mir, Moscow, 2004 (in Russian).
10. I.A. Lomazova. *Nested Petri Nets - a Formalism for Specification and Verification of Multi - Agent Distributed Systems*. Fundamenta Informaticae 43 pp. 195-214, 2000.
11. O. Prisecaru and T.Jucan. *Interorganizational Workflow Nets: a Petri Net Based Approach for Modelling and Analyzing Interorganizational Workflows*. In: Proc. of EOMAS-2008, Montpellier, France, June 16-15, 2008, CEUR Workshop Proceedings, ISSN 1613-0073.
12. W. Reisig. *Petri Nets. An Introduction*. In EATCSMonographs on Theoretical Computer Science, Springer-Verlag, 1985.