Interorganizational Workflow Nets: a Petri Net Based Approach for Modelling and Analyzing Interorganizational Workflows

Oana Prisecaru and Toader Jucan

Faculty of Computer Science, "Al. I. Cuza" University,
Gen. Berthlot St, No 16,
740083 Iasi, Romania
{otto,jucan}@infoiasi.ro
http://www.infoiasi.ro

Abstract. Interorganizational workflows represent a new technique that offers companies a solution for managing business processes that involve more than one organization. In this paper, an interorganizational workflow will be modelled using a special class of nested Petri nets, interorganizational workflow nets (IWF-nets). This approach will allow the specification of the local workflows in the organizations involved and of the communication structure between them, permitting a clear distinction between these components. The paper defines a notion of behavioural correctness (soundness) and proves this property is decidable for IWF-nets.

Key words: interorganizational workflows, workflow modelling, Petri nets, soundness

1 Introduction

A workflow is the automation of a business process that takes place inside one organization. Workflow management deals with controlling, monitoring, optimizing and supporting workflows.

Due to the rise of virtual organizations, electronic commerce and international companies, many existent business processes involve more than one organization. These workflows , distributed over a number of different organizations, are referred to as *interorganizational workflows.*

A formal method which has been successfully used for workflow modelling is Petri nets. Petri nets are a graphical and mathematical tool for modelling concurrent/distributed systems, which permit the explicit representation of the states and transitions of a system. Petri nets are a suitable modelling technique for workflows (see [2, 3]), due to several reasons: Petri nets are a graphical and intuitive language, they have a formal semantics, they are expressive, there are many analysis techniques for investigating the properties of Petri nets.

Petri nets have also been proposed for modelling interorganizational workflows: in [4], IOWF-nets are defined for modelling loosely coupled interorganizational workflows. An IOWF-net describes the local workflows and the coordination structure used for their interaction. [12] describes a XML-based language, called XRL, for the specification of interorganizational workflows. XRL semantic is expressed in terms of Petri nets. The approach in [6] uses Documentary Petri Nets, a variant of high-level Petri nets, to model and enact trade procedures. Another approach based on Petri nets is the P2P approach from [5], which uses inheritance to align local workflows.

A common problem in these approaches is the mixture between the different components of the interorganizational workflow, which makes the model difficult to understand and analyze. Also, the interoperability between the constituent workflows either is not represented explicitly in the model, or it lacks clarity.

In order to tackle these problems, 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 loosely coupled interorganizational workflows: there are n local workflow processes which can behave independently, but need to interact at certain points in order to accomplish a global business goal. The interaction is made through asynchronous or synchronous communication. *Interorganizational workflow nets (IWF-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 [2]. The communication mechanisms between the local workflows are also described using an object-net. The dynamic behaviour and the synchronization steps of the IWF-net ensure the collaboration between the constituent workflows.

This approach offers a clear distinction between all the local workflows and the communication structure, which is represented separately from the local workflows. Thus, IWF-nets ensure a modular view over the components of an interorganizational workflow. The proposed model provides a high degree of flexibility: any local workflow can be modified without interfering with the other local workflows and the communication structure can be changed without affecting the local workflows. Also, our solution, based on nested Petri nets, preserves privacy and autonomy of the local workflows: the workflow inter-visibility is reduced, since the local workflows make public only the labels (identifiers) of those tasks involved in cooperation. The paper introduces a notion of behavioural correctness for IWF-nets, *soundness*, and proves this property is decidable.

The rest of the paper is organized as follows: Section 2 introduces the basic terminology and notations related to Petri nets and workflows, Section 3 presents an informal definition of IWF-nets and a small introductory example of an IWF-net, Section 4 introduces the formal definition of IWF-nets, Section 5 defines and studies the soundness property for IWF-nets and Section 6 presents the concluding remarks.

2 Preliminaries

This section introduces the terminology and notations related to Petri nets and workflows. For details, the reader is referred to [11] and [1, 2].

The classical Petri net is a directed bipartite graph with two node types called places and transitions. The nodes are connected via directed arcs. Places are represented by circles and transitions by rectangles.

A Petri net is a triple $PN = (P, T, F)$*, where* P *is a finite set of places,* T *is a finite set of transitions* $(P \cap T = \emptyset)$*,* $F \subseteq (P \times T) \cup (T \times P)$ *is a set of arcs (flow relation).*

For $x \in P \cup T$, the *preset* of x is $\bullet x = \{y|(y, x) \in F\}$ and the *postset* of x is $x\bullet = \{y|(x, y) \in F\}$. At any time a place contains zero or more tokens, drawn as black dots. The state (or the marking), is the distribution of tokens over places: $M : P \to \mathbb{N}$ ($\mathbb{N}$ denotes the set of natural numbers). We will represent a state as follows: if $P = \{p_1, p_2, p_3\}$, then $1'p_1 + 2'p_2$ is the state with one token in place p_1, two tokens in p_2 and no tokens in p_3. Transitions change the state of the net according to the following firing rule: a transition t is said to be enabled (in a marking M) iff each place p in the preset of t contains at least one token in the marking M: $M(p) \geq 1, \forall p \in \bullet t$. If an enabled transition t fires, it changes the marking M into M': $M'(p) = M(p) - 1, \forall p \in \bullet t$ and $M'(p) = M(p) + 1, \forall p \in t\bullet$. We write $M[t\rangle M'$.

We have the following notation: $M_1[\sigma\rangle M_n$: the firing sequence $\sigma = t_1 t_2 \dots t_n \in T^*$ leads from state M_1 to M_n (i.e. there exist $M_2, \dots, M_{n-1}$ such that $M_1[t_1\rangle M_2[t_2\rangle \dots [t_n\rangle M_n$). If $\sigma = \lambda$, then $M[\sigma\rangle M$. A state M_n is called reachable from M_1 ($M_1[*\rangle M_n$) iff there is a firing sequence σ such that $M_1[\sigma\rangle M_n$. We denote by $[M\rangle$ the set of markings reachable from M. If $N = (P, T, F)$ is a Petri net and M_0 is the initial marking of N, the set of reachable markings of N is denoted by $[M_0\rangle$.

Let N be a Petri net and M_0 its initial marking. A transition t is not dead in (N, M_0) if $\exists M \in [M_0\rangle$ such that $M[t\rangle$.

There are several extensions of Petri nets, which are obtained from the classical model by adding expressions on arcs, tokens with a complex structure, modified firing rules, time concepts, etc. *Nested Petri nets* are an extension of the Petri net model, in which tokens can be Petri nets themselves (see [10]). A nested Petri net consists of a Petri net SN, called system net, several Petri nets called object-nets, a set of labels for the transitions ($Lab = Lab_v \cup Lab_h$) and a set of expressions used for labelling the arcs of SN. Tokens in SN can be either atomic tokens or net tokens (i.e. object-nets in a certain marking). In nested Petri nets, there are several firing rules ([9, 10]): an unlabelled transition from an object-net can fire if the transition is enabled in the object-net, according to the firing rule from ordinary Petri nets. The firing of such a transition is called an object-autonomous step. If two transitions with adjacent labels (from Lab_h) belonging to two object-nets (which reside in the same place of SN) are enabled in those object-nets, then they should fire synchronously. The simultaneous firing of these transitions is called an horizontal synchronization step. A transition with a label l, enabled in SN, should fire simultaneously with the transitions from the object-nets which have an adjacent label $\bar{l}$ ($l, \bar{l} \in Lab_v$). This is a vertical synchronization step. For the formal definition of nested Petri nets, the reader is referred to [10].

A workflow can be seen as a collection of tasks organized in order to accomplish some goal. The order of tasks is specified through different constructors, which permit the control of the flow of execution, e.g. sequence, choice, parallelism, and synchronization (see [13]). A task is considered to be an atomic piece of work. The workflow processes are instantiated for a specific case, thus, a case is also named a workflow instance. For example, a case can be a specific request for a lone, a tax declaration, an order.

One of the most convenient ways of specifying workflows is through the use of Petri nets: tasks can be modelled by transitions, the causal dependencies between tasks can be expressed using arcs and places. A place can model a pre or post-condition for a task. The state of the process can be explicitly modelled using the state of the Petri net.

In [2], *workflow nets (WF-nets)* are introduced for modelling workflows: a WF-net will specify the procedure that handles a single case at a time. A WF-net is a Petri net which has two special places: one source place, i, and one sink place, o. The marking in which there is only one token in the source place represents the beginning of the life-cycle of a case (and the initial marking of the net, denoted by i). The marking in which there is only one token in the sink place, represents the end of the procedure that handles the case (and the final marking of the net, denoted by o). An additional requirement is that there should not be conditions and tasks that do not contribute to the processing of the case. 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* $o\bullet = \emptyset$ *and (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.*

3 Interorganizational Workflow Nets: an Introductory Example

In this section we will present an informal definition of interorganizational workflow nets and then we will apply their key features to a simple example of an interorganizational workflow.

The interorganizational workflows we are interested in are based on a special type of interoperability between processes : the global workflow consists of loosely coupled workflow processes which operate independently, but need to communicate and synchronize their activity at certain points in order to accomplish correctly the global workflow process. There are two ways of interaction between processes: asynchronous communication (corresponding to the exchange of messages) and synchronous communication. Thus, the interorganizational workflow consists of private workflows and a communication structure.

In order to model interorganizational workflows, we use an approach based on nested nets: *interorganizational workflow nets (IWF-nets)* are nested Petri nets, extended with two special sets used for describing the communication structure between the local workflows, SC and AC. 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 $x \in SC$, then, all the transitions from x have to execute at the same time. Also, the IWF-net has two special sets of labels used for synchronizing the transitions involved in the elements of AC and SC: the labels in Lab_{AC} are used for the transitions which appear in the elements of AC and the labels from Lab_{SC} are used for the transitions which appear in the elements of SC. In an IWF-net, the system net SN comprises three places (I, p, O) and a transition (end). Tokens in SN can be atomic tokens (without inner structure) or net-tokens (object-nets). The arcs of SN are labelled with expressions: the arc (p, end) is labelled with the expression $(x_1, \ldots, x_n, x_{n+1})$ (where $x_1, \ldots, x_n, x_{n+1}$ are variables), while the rest of the arcs are labelled with the constant 1 (which is not represented explicitly on arcs). In an IWF-net, there are $n+1$ object-nets: n object-nets (extended WF-nets) representing the local workflows involved in the interorganizational workflow and one object-net, C, which describes a part of the communication structure (the asynchronous communication). The structure of C results from AC: if $ac = (t, t') \in AC$, then, in C there is a place p_{ac}, a transition t_c (corresponding to t), a transition t'_c (corresponding to t'), one arc from t_c to p_{ac} and one arc from p_{ac} to t'_c. In an IWF-net there is a partial function, Λ, which labels transitions from the object-nets and the transition end from SN. If $x \in SC$ is a set of transitions which must fire synchronously, then all the transitions from x 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 with the same label as t: $\Lambda(t) = \Lambda(t_c) = l, l \in Lab_{AC}$.

In our example, there are two local workflow processes: the first process contains the tasks t_1, t_2, t_3 and t_4, while the second process contains the tasks t_5, t_6, t_7 and t_8. The two workflow processes are modelled by two extended workflow nets, WF'_1 and WF'_2 (see Fig. 1). These nets are workflow nets, extended with transitions which empty the sink places of the workflow nets (transition t'_1 in WF'_1 and t'_2 in WF'_2). The initial marking of WF'_1 is i_1 and the initial marking of WF'_2 is i_2.

In the interorganizational workflow, the two workflow processes interact as follows: task t_1 in WF'_1 must fire before the tasks t_5 and t_6 in WF'_2, task t_6 in WF'_2 must fire before task t_3 in WF'_1. We define, thus, a partial order on tasks, describing the asynchronous communication between the two workflow processes: $AC = \{(t_1, t_5), (t_1, t_6), (t_6, t_3)\}$. Also, task t_4 in WF'_1 and task t_8 in WF'_2 must fire synchronously (there is a synchronous communication between the two local workflows, through these transitions). We will define the set of synchronous communication elements: $SC = \{\{t_4, t_8\}\}$. The IWF-net used for modelling this interorganizational workflow consists of the system net SN and of three object nets, WF'_1, WF'_2 and C. In the initial marking of the net (Fig.

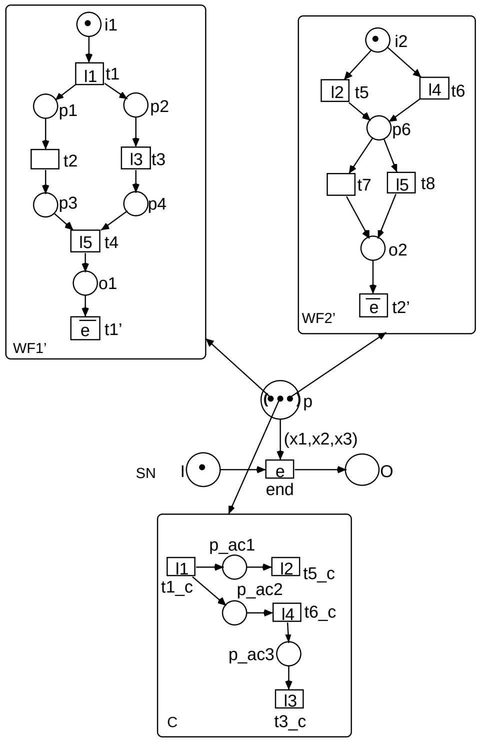

Fig. 1. An example of an IWF-net

1), there is an atomic token in place I and all the object-nets reside in place p. The initial marking of C is 0 (there are no tokens in the places of C). The arc (p, end) is assigned the expression (x_1, x_2, x_3), while the rest of the arcs are assigned the constant expression 1.

The object-net C results from the asynchronous communication relation. The set of places is $P_C = \{p_{ac_1}, p_{ac_2}, p_{ac_3}\}$, where $ac_1 = (t_1, t_5)$, $ac_2 = (t_1, t_6)$, $ac_3 = (t_6, t_3)$. The transitions in T_C correspond to the transitions involved in asynchronous communication: $T_C = \{t_{1c}, t_{3c}, t_{5c}, t_{6c}\}$. The arcs of C are obtained using the sets P_C, T_C and AC: for instance, since $ac_1 = (t_1, t_5) \in AC, p_{ac_1} \in P_C, t_{1c}, t_{5c} \in T_C$, then we will add the arcs (t_{1c}, p_{ac_1}) and (p_{ac_1}, t_{5c}).

Some of the transitions of the IWF-net are labelled. The sets of labels are: $Lab_{AC} = \{l_1, l_2, l_3, l_4\}$, $Lab_{SC} = \{l_5\}$ and a set $\{e, \overline{e}\}$. The transitions involved in the asynchronous communication elements will be assigned asynchronous communication labels: in WF_1', $\Lambda(t_1) = l_1, \Lambda(t_3) = l_3$ and in WF_2', $\Lambda(t_5) = l_2$ and $\Lambda(t_6) = l_4$. The transitions from the synchronous communication element will be assigned the same synchronous communication label: $\Lambda(t_4) = \Lambda(t_8) = l_5$. We also have $\Lambda(t_1') = \overline{e}$ in WF_1', $\Lambda(t_2') = \overline{e}$ in WF_2' and $\Lambda(end) = e$ in SN. The labels for the transitions in C are assigned as follows: since $(t_1, t_5) \in AC$, then $\Lambda(t_{1_c}) = \Lambda(t_1) = l_1$ and $\Lambda(t_{5c}) = \Lambda(t_5) = l_2$. $(t_1, t_6) \in AC$, then $\Lambda(t_{6c}) = \Lambda(t_6) = l_4$. $(t_6, t_3) \in AC$, then $\Lambda(t_{3c}) = \Lambda(t_3) = l_3$.

The firing rules in an IWF-net are the same as those from nested Petri nets. The only difference is that we allow the horizontal synchronization of transitions belonging to several object-nets (not just from two object-nets, as defined in [9, 10]). In an IWF-net, an unlabelled transition from an extended WF-net can fire if the transition is enabled in that extended WF-net (an object-autonomous step). Also, if several labelled transitions, with the same label, from some object-nets are enabled in those object-nets, then they should fire synchronously (an horizontal synchronization step). Finally, if the transition *end* is enabled in SN, then it should fire simultaneously with the transitions from the object-nets labelled with $\overline{e}$, if these transitions are enabled in their corresponding object-nets (the vertical synchronization step).

In the example in Fig. 1, transition t_1 is enabled in (WF_1', i_1) and the transitions t_5 and t_6 are enabled in (WF_2', i_2). But transition t_5 should fire at the same time with transition t_{5c} in the object-net C (because $\Lambda(t_5) = \Lambda(t_{5c})$) and transition t_6 should fire at the same time with transition t_{6c} in the object-net C (because $\Lambda(t_6) = \Lambda(t_{6c})$). The transitions t_{5c} and t_{6c} are not enabled in $(C, 0)$. So, transitions t_5 and t_6 cannot fire yet. This behaviour is consistent with the restrictions imposed by the asynchronous communication relation: $(t_1, t_5), (t_1, t_6) \in AC$ means that t_1 should fire before t_5 and t_6. Since transition t_1 is enabled in (WF_1', i_1), transition t_{1c} is enabled in $(C, 0)$, then the horizontal synchronization step $(; t_1, t_{1c})$ is enabled in marking M_0. In the resulting marking, M_1, place I contains an atomic token, place O contains no tokens, and place p contains three object-nets with their corresponding new markings: WF_1' (with the marking $m1_1 = 1'p_1 + 1'p_2$, i.e. p_1 and p_2 have one token), WF_2' (in its initial marking, i_2) and C (with the marking $mc_1 = 1'p_{ac_1} + 1'p_{ac_2}$). We write: $M_1 = (1, ((WF_1', m1_1), (WF_2', i_2), (C, mc_1)), 0)$. In $(WF_1', m1_1)$ the unlabelled transition t_2 is enabled. The firing of this local transition is an object-autonomous step and the resulted marking is $M_2 = (1, ((WF_1', m1_2), (WF_2', i_2), (C, mc_1)), 0)$, where $m1_2 = 1'p_2 + 1'p_3$. One can notice that only the marking of the object-net WF_1' has changed. Transition t_3 in WF_1' can only fire synchronously with transition t_{3c} in C, but transition t_{3c} is not enabled in (C, mc_1). The horizontal synchronization steps $(; t_5, t_{5c})$ and $(; t_6, t_{6c})$ are enabled in M_2. If $(; t_6, t_{6c})$ fires, it produces the marking $M_3 = (1, ((WF_1', m1_2), (WF_2', m2_1), (C, mc_2)), 0)$, where $m2_1 = 1'p_6$ and $mc_2 = 1'p_{ac_1} + 1'p_{ac_3}$. If we assign to x_1 the object-net $(WF_1', m1_4)$, to x_2 the object-net $(WF_2', m2_2)$ and to x_3 the object-net (C, mc_3), then transition *end* is enabled in M_5: place I contains an atomic token, place p contains three object-nets. Since t_1' is enabled in $(WF_1', m1_4)$ and t_2' is enabled in $(WF_2', m2_2)$, then the vertical synchronization step $(end; t_1', t_2')$ is enabled in M_5. The firing of this step removes the atomic token from I, the object-nets from p and adds an atomic token to place O.

4 Definition of Interorganizational Workflow Nets

This section introduces the definition of interorganizational workflow nets, a model based on nested Petri nets, for loosely coupled interorganizational workflows.

We will assume that there are n local workflows belonging to n business partners. Each business partner has total control over his own workflow process and the local workflows interact at certain points, according to a communication structure. There are two types of communication: asynchronous communication (corresponding to the exchange of messages between workflows) and synchronous communication (which forces the local workflows to execute specific tasks at the same time) .

We define, first, *extended workflow nets*, an extension of the WF-nets defined in Sect. 2.

Definition 1. *Let $WF = (P, T, F)$ be a WF-net. The extended WF-net is $WF' = (P, T', F')$, where $T' = T \cup \{t'\}$ and $F' = F \cup \{(o, t')\}$*

WF is called the underlying net of WF'. One can notice that the set of reachable markings of WF' includes the set of reachable markings of WF. The final marking of an extended workflow net is the empty marking (the marking in which all the places are empty), denoted by 0. This marking is reachable only from the final marking of WF, o, by firing the transition t'.

Extended WF-nets will be used for modelling the local workflows from the interorganizational workflow.

Interorganizational workflow nets (IWF-nets) are defined as a special class of nested Petri nets. IWF-nets are 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.

Definition 2. *An interorganizational workflow net IWF is a nested Petri net:*
$IWF = (Var, Lab, (WF'_1, i_1), \ldots, (WF'_n, i_n), AC,$
$SC, (C, 0), SN, \Lambda)$ *such that:*

1. $Var = \{x_1, x_2, \ldots, x_n, x_{n+1}\}$ *is a set of variables.*
2. $Lab = Lab_{AC} \cup Lab_{SC} \cup \{e, \overline{e}\}$ *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 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:*
   – $\forall x, y \in SC : x \cap y = \emptyset$.
   – *if $t \in T_i, t' \in T_j, t, t' \in x, x \in SC$, then $i \neq j$.*
6. *$C = (P_C, T_C, F_C)$ is the communication object:*
   – $P_C = \{p_{ac} | ac \in AC\}$.
   – $T_C = \{t_c | \exists (t', t) \in AC \vee (t, t') \in AC\}$.

– $F_C = \{(p,t) \in P_C \times T^\circ | p = (t',t) \in AC\} \cup \{(t,p) \in T^\circ \times P_C | p = (t,t') \in AC\}$

7. $SN = (N, W, M_0)$ *is the system net of IWF, such that:*
   – $N = (P_N, T_N, F_N)$ *is a Petri net:*
     - $P_N = \{I, p, O\}$, *where* O *is a place such that* $O\bullet = \emptyset$ *and* I *is a place such that* $\bullet I = \emptyset$.
     - $T_N = \{end\}$.
     - $F_N = \{(I, end), (p, end), (end, O)\}$.
   – W *is the arc labelling function:* $W(I, end) = 1$, $W(p, end) = (x_1, x_2, \ldots, x_{n+1})$, $W(end, O) = 1$.
   – M_0 *is the initial marking of the net:* $M_0(I) = 1$, $M_0(p) = ((WF'_1, i_1), \ldots, (WF'_n, i_n), (C, 0))$ *and* $M_0(O) = 0$.
8. Λ *is a partial labelling function such that:*
   – $\forall x \in SC, \forall t, t' \in x, \Lambda(t) = \Lambda(t') = l, l \in Lab_{SC}$.
   – *if* $t \in T^\circ$ *such that* $(t,t') \in AC$ *or* $(t',t) \in AC$, *then there exists* $t_c \in T_C : \Lambda(t_c) = \Lambda(t) = l, l \in Lab_{AC}$.
   – $\Lambda(t'_i) = \overline{e}, \forall i \in \{1, \ldots n\}$ *and* $\Lambda(end) = e$.
   – $\forall t, t' \in T_i (i \in \{1, \ldots, n\}) : \Lambda(t) \neq \Lambda(t')$.

In an IWF-net there are n object-nets (extended WF-nets) representing the local workflows corresponding to the n business partners involved in the interorganizational workflow.

Var is the set of variables in the net. Variables x_i, $i \in \{1, \ldots, n\}$ will take as value an object WF-net in a certain marking. Variable x_{n+1} will take as value the object-net C in a certain marking.

Lab is a set of labels: the labels in Lab_{AC} are used for asynchronous communication elements and the labels from Lab_{SC} are used for synchronous communication elements. Lab_{AC} and Lab_{SC} are not necessary disjoint. The label $\overline{e}$ is used for labelling the transition t'_i from WF'_i, $\forall i \in \{1, \ldots, n\}$. The label e is used for the transition end from SN.

AC defines a partial order on the transitions of the extended workflow nets, representing the asynchronous communication: if $(t,t') \in AC$, t must execute before t' (i.e. t' waits for a message which is sent after the firing of t). Also, t and t' should not belong to the same set of transitions T_i ($i \in \{1, \ldots n\}$): if t and t' belong to the same local workflow, there exist other mechanisms for synchronizing these transitions and it is not necessary to specify an asynchronous communication relation between them.

SC represents the set of synchronous communication elements: if $x \in SC$, then, all the transitions from x have to execute at the same time. x should not contain two transitions from the same set of transitions T_i: the local transitions from the same workflow should not be synchronized using SC. All the sets in SC should be disjoint.

C is an object-net which describes the asynchronous communication between workflows. C can be constructed automatically using the elements of AC: for every asynchronous communication element $ac \in AC$, there is a corresponding place p_{ac} in P_C. For every transition $t \in T^\circ$ involved in an asynchronous communication element, there is a transition $t_c \in T_C$. Also, if $ac = (t,t') \in AC$,

$t_c \in T_C$ is the transition which corresponds to t, $t'_c \in T_C$ is the transition which corresponds to t', p_{ac} is the place which corresponds to ac, then there exist two arcs $(t_c, p_{ac}), (p_{ac}, t'_c) \in F_C$. The initial marking of C is the empty marking, denoted by 0.

W is a function that assigns to each arc in SN an expression. In IWF-nets, an expression can be either the pair $(x_1, \ldots, x_n, x_{n+1})$ or the constant 1.

Λ is a partial function which assigns labels from Lab to certain transitions. All the transitions from a synchronous communication element $x \in SC$ have the same label $l \in Lab_{SC}$. If t is a transition from an extended WF-net which appears in an asynchronous communication element $ac \in AC$, then $\Lambda(t) \in Lab_{AC}$. Since t belongs to an asynchronous communication element, there is a corresponding transition, t_c, in the object-net C. t and t_c have the same label: $\Lambda(t) = \Lambda(t_c) = l$. In any extended WF-net, there are no transitions with the same label.

We denote by A_{net} the net tokens of the IWF-net:
$A_{net} = \{(EN, m) \ / \ m$ is a marking of EN, $EN \in \{WF'_1, \ldots, WF'_n, C\}\}$.

A marking of an IWF-net is a function such that: $M(I) \in \mathbb{N}$, $M(O) \in \mathbb{N}$ and $M(p) \in A_{net}^{n+1}$. We write M as a vector $M = (M(I), M(p), M(O))$.

Definition 3. *A binding (of transition* end*) is a function* $b : Var \rightarrow A_{net}$.

If *expr* is an expression, $expr(b)$ denotes the evaluation of *expr* in binding b. $expr(b)$ is obtained from *expr* by replacing every variable $z \in Var$ from *expr* with $b(z)$. If *expr* is a constant expression (an expression without variables), then $expr(b) = expr$.

Definition 4. *Transition* end *from the system net* SN *of an IWF-net is enabled in a marking* M *w.r.t. a binding* b *if and only if:*
$\forall q \in \bullet end : W(q, end)(b) = M(q)$*, where* $W(q, end)(b)$ *is the arc expression of the arc* (q, end) *evaluated in binding* b.

Transition *end* is enabled in a marking M w.r.t. the binding b if $M(I) = 1$ and the expression from the arc (p, end) evaluates to the same tuple of tokens from place p: $W(p, end)(b) = M(p)$.

There are several types of steps, defining the behaviour of nested Petri nets (see [9, 10]). In the case of IWF-nets, which are a special class of two-level nested Petri nets, these steps are:

Definition 5. *A vertical synchronization step:*
If transition end *is enabled in a marking* M *w.r.t. a binding* b *and every transition* t'_i $(\Lambda(t'_i) = \overline{e})$ *is enabled in the object-net* $b(x_i) = (WF'_i, m_i), \forall i \in \{1, \ldots, n\}$*, then the simultaneous firing of* end *and* $t'_1, \ldots, t'_n$ *is a vertical synchronization step.*

The firing of the vertical synchronization step $(end; t'_1, \ldots, t'_n)$ *in marking* M *produces the marking* $M' = (0, 0, 1)$.

This step removes the object-nets from p and the atomic token from I. In the resulting marking, M', there is only one atomic token in place O.

Definition 6. *An object - autonomous step:*
Let M be a marking of an IWF-net and $(\alpha_1, \alpha_2, \ldots, \alpha_{n+1})$ a tuple of tokens from p. Let α_i be one of the object-nets ($i \in \{1, \ldots, n\}$) $\alpha_i = (WF'_i, m)$. Let t be a transition in α_i such that t is enabled in marking m, $\Lambda(t)$ is undefined and $m[t\rangle m'$ (i.e. the firing of t, by the firing rule from the classical Petri nets, produces a new marking m' in α_i).

Let M' be a marking of the IWF-net obtained from the old marking M by replacing, in the tuple $(\alpha_1, \alpha_2, \ldots, \alpha_{n+1})$ from $M(p)$, the net token α_i with the net token α'_i, where $\alpha'_i = (WF'_i, m')$. We write: $M[;t\rangle M'$.

An object-autonomous step is the firing of a local unlabelled transition in one of the local workflows. None of the object-nets are moved from the place p.

Definition 7. *A horizontal synchronization step:*
Let M be a marking of IWF and $(\alpha_1, \alpha_2, \ldots, \alpha_{n+1})$ a tuple of net-tokens from p. Assume $t_1, \ldots t_s \in T^\circ$ is the set of all the transitions with the same label $l \neq e$, $\Lambda(t_1) = \Lambda(t_2) = \ldots = \Lambda(t_s) = l$, such that: every transition t_j ($j \in \{1, \ldots, s\}$) is enabled in a net-token $\alpha_{k_j} = (EN_j, m_j)$ ($\{k_1, \ldots k_s\} \subseteq \{1, \ldots, n+1\}$, $EN_j \in \{WF'_1, \ldots, WF'_n, C\}$) and $m_j[t_j\rangle m'_j$ (by means of classical Petri nets). The synchronous firing of $t_1, \ldots, t_s$ is called a horizontal synchronization step.

The resulting marking, M', is obtained from M by replacing the tuple $(\alpha_1, \alpha_2, \ldots, \alpha_{n+1})$ from place p with the tuple $(\alpha'_1, \alpha'_2, \ldots, \alpha'_{n+1})$, where $\alpha'_{k_j} = (EN_j, m'_j), \forall j \in \{1, \ldots, s\}$ and $\alpha'_i = \alpha_i, \forall i \in \{1, \ldots, n+1\} \setminus \{k_1, \ldots k_s\}$. We write: $M[;t_1, \ldots, t_s\rangle M'$.

If in a marking M of an IWF-net, all the transitions $t_1, \ldots, t_s$, with the same label $l \neq e$, are enabled in the object-nets $\alpha_{k_1} = (EN_1, m_1), \ldots, \alpha_{k_s} = (EN_s, m_s)$ from $M(p)$, then the simultaneous firing of these transitions is a horizontal synchronization step.

The definitions of the vertical synchronization step and of the object-autonomous step are the definitions from [10], adapted for the structure of IWF-nets. The definition of the horizontal synchronization step for IWF-nets extends the definition from [10], allowing the horizontal synchronization of transitions from several object-nets. This extension does not change the properties of nested Petri nets.

5 The Soundness Property for Interorganizational Workflow Nets

In this section we will introduce a notion of soundness for IWF-nets.

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 complete a case, any case must terminate correctly, and every task should contribute to at least one possible execution of the workflow. In a WF-net, completion of a case is signalled by a token in its sink place. Thus, the completion (or termination) option means that it must always be possible to produce a token for this place. Correct termination means that, as soon as a token is produced for the sink place, all other places must be empty. The last requirement means that a WF-net should not have any dead transitions. The formal definition of soundness (from [1]) is:

Definition 8. *A workflow net WF = (P, T, F) is sound iff:*

1. *For every marking m reachable from the initial marking i, there exists a firing sequence leading from m to the final marking o (termination condition):* $(\forall m)((i[*\rangle m) \Longrightarrow (m[*\rangle o))$.
2. *Marking o is the only marking reachable from state i with at least one token in place o:* $(\forall m)((i[*\rangle m) \wedge m \geq o) \Longrightarrow (m = o))$.
3. *There are no dead transitions in WF:* $(\forall t \in T)(\exists m, m')(i[*\rangle m[t\rangle m')$.

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

Definition 9. *Let WF' be an extended workflow net and WF its underlying workflow net. WF' is sound if WF is sound.*

In an interorganizational workflow, although the local workflows are sound, we can have synchronization errors and it is possible that communication elements introduce interlocking. In the interorganizational workflow in Fig. 1, the two local workflows are sound. But if we consider the execution sequence $M_0[Y_1\rangle M_1[Y_2\rangle M_2$, where $Y_1 = (;t_1, t_{1c})$, $Y_2 = (;t_5, t_{5c})$, then, transition t_3 will never fire in WF_1', so the local workflow WF_1' does not terminate (transition t_4 will never be enabled).

We will define a notion of soundness for interorganizational workflows. We will consider the final state for an IWF-net, a marking M_f, in which there is only one atomic token in place O: $M_f = (0, 0, 1)$. An IWF-net is sound if: (1) every extended WF-net WF_i' $(i \in \{1, \ldots, n\})$ is sound and (2) for any reachable marking of the IWF-net, $M \in [M_0\rangle$, there is a firing sequence that leads to M_f.

We can define formally the notion of soundness for an IWF-net as follows:

Definition 10. *An interorganizational workflow net* $IWF = (Var, Lab, (WF_1', i_1), \ldots, (WF_n', i_n), AC, SC, (C, 0), SN, \Lambda)$ *is sound if and only if:*

1. (WF_j', i_j) *is a sound extended workflow net,* $\forall j \in \{1, \ldots, n\}$.
2. *For every marking M reachable from the initial marking M_0, there exists a firing sequence leading from M to the final marking M_f:* $(\forall M)((M_0[*\rangle M) \Longrightarrow (M[*\rangle M_f))$.

First, we consider the interorganizational workflow is sound if the extended WF-nets describing the local workflows are sound. The final marking of the IWF-net is reached if and only if the vertical synchronization step fires. This implies that all the transitions t_i' are enabled in WF_i' $(i \in \{1, \ldots, n\})$, which happens if and only if the final markings in the extended WF-nets have been reached. Thus, the second condition from the soundness definition basically states that the interorganizational workflow is sound if the termination condition still holds for every WF-net, when the firing of tasks is restricted by the communication structure.

The notion of soundness for IWF-nets is weaker than the notion of soundness defined in [2], as it does not impose the absence of dead steps in the IWF-net.

The notion of soundness for IWF-nets does not impose a condition similar to condition (2) from Def. 8, because such a condition always holds for sound IWF-nets:

Lemma 1. *Let IWF be a sound IWF-net. Marking M_f is the only marking reachable from M_0 with at least one token in place O:* $(\forall M)((M_0[*\rangle M) \wedge M \geq M_f) \Longrightarrow (M = M_f))$

Proof. Assume $M \in [M_0\rangle$ and $M \geq M_f$. If $M \geq M_f$, then $M(O) \geq 1$. But tokens can be added to place O only when transition *end* fires in IWF. This transition can only fire once and it empties place I and place p. No other steps can fire after this. So, the only reachable marking with $M(O) \geq 1$ is M_f.

In order to decide whether the soundness property defined is decidable, we introduce a partial order on the markings of the IWF - net (see [10]):

Definition 11. *Let IWF be an IWF-net, M_1 and M_2 markings of IWF. $M_1 \preceq M_2$ if and only if $M_1(I) \leq M_2(I)$, $M_1(O) \leq M_2(O)$ and there is an embedding $J_p : M_1(p) \to M_2(p)$, such that for $\alpha = (\alpha_1, \ldots, \alpha_{n+1}) \in M_1(p)$ and for $J_p(\alpha) = \alpha' = (\alpha'_1, \ldots \alpha'_{n+1})$ we have for $i \in \{1, \ldots, n+1\}$ either $\alpha_i = \alpha'_i$ or $\alpha_i = (EN, m)$ and $\alpha'_i = (EN, m')$ ($EN \in \{WF'_1, \ldots, WF'_n, C\}$) and for all the places q of EN: $m(q) \leq m'(q)$.*

Definition 12. *Let IWF be an IWF-net and M and M' two markings of IWF. The marking M covers M' (w.r.t. the partial ordering $\preceq$) if $M' \preceq M$.*

Definition 13. *Given a set of markings $Q = \{q_1, q_2, \ldots, q_n\}$ and an initial marking M,* the inevitability problem *is to decide whether all computations starting from M eventually visit a marking not covering (w.r.t. the partial ordering $\preceq$) one of the markings from Q.*

It was proven in [9, 10] that the inevitability problem is decidable for nested Petri nets.

Theorem 1. *Let IWF be an IWF-net and $M \in [M_0\rangle$. There is a firing sequence $M[*\rangle M_f$ if and only if there is a firing sequence $M[*\rangle M'$ and M' does not cover (w.r.t. $\preceq$) the marking* $(1, 0, 0)$.

Proof:
($\Longrightarrow$) Assume $M[*\rangle M_f$ in IWF. Since M_f does not cover the marking $(1, 0, 0)$, we can consider $M' = M_f$.
($\Longleftarrow$) We assume there exists a firing sequence from marking M to a marking M' which does not cover the marking $(1, 0, 0)$. If M' does not cover $(1, 0, 0)$, then $M'(I) = 0$ (there are no tokens in place I). Marking M' is reachable from M_0

(because $M_0[*\rangle M[*\rangle M'$). $M'(I) = 0$ if and only if the vertical synchronization step $Y = (end[b]; t'_1, \ldots, t'_n)$ fires in IWF. The firing of this step always leads to the marking M_f (so, $M' = M_f$). This implies there is a firing sequence such that $M[*\rangle M_f$.

Theorem 2. *The soundness problem is decidable for IWF - nets.*

Proof: Let IWF be an IWF-net. Using the definition of soundness and Theorem 1, IWF is sound if and only if: (1) WF'_i are sound, $\forall i \in \{1, \ldots, n\}$ and (2) for any reachable marking in IWF, $M \in [M_0\rangle$, there exists a firing sequence $M[*\rangle M'$ such that M' does not cover (w.r.t. $\preceq$) the marking $(1, 0, 0)$. The soundness of the extended WF-nets is decidable (because the soundness for WF-net is decidable) and condition (2) is equivalent to the inevitability problem, if we consider the marking M and the set of markings $Q = \{(1, 0, 0)\}$.

6 Conclusions

In this paper we introduced a new approach on the modelling of interorganizational workflows, based on nested Petri nets: the local workflows and the communication structure are modelled as object-nets. The local workflows have an independent behaviour and they can interact, according to the given communication structure, using the synchronization mechanisms offered by nested Petri nets. This approach has several advantages: one can have a modular view on the interorganizational workflow, because the local workflows and the communication structure are distinct elements in IWF-nets; steps in IWF- nets can easily express the synchronous and the asynchronous communication; IWF-nets represent a flexible model for interorganizational workflows, because any component can be modified easily, with minimal changes to the other components. This is an important requirement for interorganizational workflows, since the communication structure between the local workflows can change in time. Also, our contribution preserves the privacy of the local workflows: the local workflows only share two sets of labels used for the synchronous and the asynchronous communication. A notion of soundness was introduced for IWF-nets: an IWF-net is sound if and only if all the extended workflow nets which describe the local workflows are sound and for any reachable marking of the IWF-net, there is a firing sequence that leads to the final marking. We proved this property is decidable for IWF-nets. Future work aims at defining IWF-nets which will model interorganizational workflows in which every local workflow processes batches of cases, instead of one case in isolation.

References

1. W. M. P. van der Aalst. *Verification of Workflow Nets.* P. Azema and G. Balbo, editors, Application and Theory of Petri nets 1997, volume 1248 of Lecture Notes in Computer Science, pp. 407-426, SpringerVerlag, Berlin, 1997.

2. 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.
3. W.M.P. van der Aalst. *Three Good Reasons for Using a Petri-net-based Workflow Management System.* Wakayama et al., editors. Information and Process Integration in Enterprises: Rethinking Documents. Volume 428 of The Kluwer International Series in Engineering and Computer Science. Boston, Kluwer Academic Publicers, pp.161-182, 1998.
4. 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.
5. W.M.P. van der Aalst and M. Weske. *The P2P approach to Interorganizational Workflows* In K.R. Dittrich, A. Geppert, and M.C. Norrie, editors, Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01), volume 2068 of Lecture Notes in Computer Science, pages 140156. Springer-Verlag, Berlin, 2001.
6. R.W.H. Bons, R.M. Lee, and R.W. Wagenaar. *Designing trustworthy interorganizational trade procedures for open electronic commerce.* International Journal of Electronic Commerce, 2(3):6183, 1998.
7. I. Chebbi, S. Dustdar, S. Tata. *The view-based approach to dynamic interorganizational workflow cooperation.* In Data & Knowledge Engineering, Volume 56 , Issue 2, pp. 139-173, 2006.
8. O. Captarencu, T. Jucan. *Interorganizational Workflows - an approach based on Level Petri Nets.* Scientific Annals of "Al. I. Cuza" University, Tome 13: pp. 17-38, 2003.
9. I. A. Lomazova, Ph. Schnoebelen. *Some Decidability Results for Nested Petri Nets.* Ershov Memorial Conference: pp. 208-220, 1999.
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. W. Reisig. *Petri Nets. An Introduction.* In EATCSMonographs on Theoretical Computer Science, Springer-Verlag, 1985.
12. H. M. W. Verbeek, A. Hirnschall, Wil M. P. van der Aalst. *XRL/Flower: Supporting Inter-organizational Workflows Using XML/Petri-Net Technology.* Proceedings of Web Services, E-Business, and the Semantic Web, WES 2002, Toronto, Canada, May 2002,pp. 93-108.
13. WFMC. *Workflow Management Coalition Terminology and Glossary (WFMC-TC-1011).* Technical Report, The Workflow Management Coalition, Brussels 1999.