A Petri Net based Approach to Modelling Resource Constrained Interorganizational Workflows

Oana Prisecaru

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

Abstract. Interorganizational workflows represent a 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, *resource constrained interorganizational workflow nets*. This approach will allow the specification of the participating workflows and of the communication structure between them, permitting a clear distinction between these components. In our model, the resources from one workflow can be represented explicitly and shared with other component workflows.

1 Introduction

A workflow is the automation of a business process that takes place inside one organization. A workflow is structured into several perspectives, among which we mention: *the process perspective* - specifies which tasks need to be executed and in what order; *the resource perspective* - specifies the resources in the organization and the existing roles (resource classes based on organizational or functional aspects). 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*. There have been developed several specification languages for interorganizational workflows, based on XML and Web services: WSFL, BPEL4WS, XLANG, WSCL, etc ([8]). These languages lack formal semantics and analytical power (they cannot be used to study behavioural properties of interorganizational workflows). In order to solve these problems, several formalisms have been proposed for specifying interorganizational workflows: Communicating Finite Automata ([6]), Category theory ([7]), Process algebra and Petri nets. Petri nets represent a well-known formal method, successfully used as a modelling technique for workflows (see [1, 2]), due to their graphical representation, their formal semantics and expressiveness. Also, there are many analysis techniques and tools used for investigating the properties of Petri nets. Petri nets have also been used for modelling interorganizational workflows: in [3], IOWF-nets are defined for modelling loosely coupled interorganizational workflows.

Prisecaru O. (2008).
A Petri Net based Approach to Modelling Resource Constrained Interorganizational Workflows.
In *Proceedings of the 6th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, pages 29-38
DOI: 10.5220/0001736700290038

[15] 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 [5] uses Documentary Petri Nets, a variant of high-level Petri nets, to model and enact trade procedures. The P2P approach from [4], based on Petri nets, 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. These approaches do not take into consideration the resources involved in the execution of the local workflows.

This paper presents a new approach on the modelling of interorganizational workflows, based on nested Petri nets. Nested Petri nets ([10]) are Petri nets 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. *Resource constrained interorganizational workflow nets (RIWF-nets)* are introduced as a special case of nested Petri nets, in which the process and the resource perspective of the local workflows, as well as the communication mechanisms between all the local workflows are modelled as distinct object-nets. Our model permits the sharing of certain resources from one organization with other participating workflows. This approach offers a clear distinction between all the local workflows and the communication structure, ensuring a modular view over the interorganizational workflow. The paper introduces a notion of behavioural correctness for RIWF-nets, *soundness*, and proves this property is decidable.

In what follows we will give the basic terminology and notation concerning workflow nets. We assume the reader is familiar with the Petri net terminology and notation. In [1] *workflow nets (WF-nets)* are introduced for modelling the process perspective: a WF-net specifies the procedure that handles a single case (workflow instance) at a time. A WF-net is a Petri net with two special places: a source place, i, and a sink place, o. In a WF-net 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* $o\bullet = \emptyset$*. (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 $m : P \to \mathbb{N}$ (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 denoted by i. The marking $1'o$, represents the end of the procedure that handles the case (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 RIWF-net, Section 3 introduces RIWF-nets, Section 4 defines and studies the soundness property for RIWF-nets and Section 5 presents the concluding remarks.

2 An Introductory Example

In what follows we will present an introductory example of an interorganizational workflow, modelled by a resource constrained interorganizational workflow net (a RIWF-net). Our interorganizational workflow consists of two loosely coupled workflows. In the resource perspective of the first workflow, there are two types of resources, *clerks* and *economists*, which will execute some of the tasks of the workflow. In order to ensure the flexibility of the system, resources will be assigned different roles, according to their capabilities (a resource can play different roles at different moments of time). The possible roles the resources can take are *secretary* and *manager*. The tasks of the workflow will be executed by appropriate roles (and not directly by resources). This way of using resources is called *role-based allocation*. The specification for the resource perspective consists in the set of resource types (RT), the set of roles (RO) and a function, res, which describes, for each role, the set of resource types that can be mapped onto that role. In our example, a *secretary* role can be performed by a clerk, while a *manager* role can be performed by an economist. Resources can be allocated dynamically to certain roles. The specification for the resource perspective in our example is $< RT, RO, res >$, where *RT* = {*clerks,economists*}, *RO* = {*secretary, manager*}, *res(secretary)*={*clerks*}, *res(manager)*={*economists*}. The resource perspective is described by the object-net RN_1 (a Petri net, called *resource net*) in Fig.1. Every element from RT and RO is described by a place in RN_1. The transitions *assign_secretary* and *assign_manager* allow the system to assign resources to certain roles, according to the function res. The dual transitions, *release_secretary* and *release_manager* are used to release the resources from roles. In the process perspective, (described by the extended WF-net WF'_1 in Fig.1), task t_1 needs a role *secretary* for its execution, while t_3 needs a role *manager* for its execution . t'_1 is a special transition which empties the place o_1.

The specification for the resource perspective of the second workflow is $< RT, RO, res >$: *RT* = {*work-rs*}, *RO* = {*administrator, supervisor*}, *res(administrator)*= {*workers*}, *res(supervisor)*={*workers*}. In the process perspective, task t_4 needs an *administrator* role for its execution, while task t_5 needs a *supervisor* role.

In the interorganizational workflow, the workflow processes need to interact at certain points, according to a certain communication structure. There are two ways of interaction: asynchronous communication and synchronous communication. In our case, in order to describe the asynchronous communication, we define a partial order on tasks: $AC = \{(t_1, t_4)\}$ (i.e. task t_1 in WF'_1 must fire before t_4 in WF'_2). Task t_3 in WF'_1 and task t_6 in WF'_2 must fire synchronously (there is a synchronous communication between the two workflows, through these transitions). We define the set of synchronous communication elements: $SC = \{\{t_3, t_6\}\}$. The RIWF-net used for modelling this interorganizational workflow is a nested Petri net which consists of a system net, SN and of five object- nets. SN is a Petri net with expressions on arcs, whose places can contain atomic tokens or net-tokens (object-nets). Thus, in the initial marking of the net, there is an atomic token in place I and all the object-nets reside in place p: (WF'_1, i_1), $(RN_1, r_{10}), r_{10} = 1'clerks + 1'economists$, (WF'_2, i_2), $(RN_2, r_{20}), r_{20} = 2'workers$, $(C, 0)$. Some of the transitions of the RIWF-net are labelled using a partial function, Λ. The transitions from AC will be assigned asynchronous communication labels: $\Lambda(t_1) = l_1, \Lambda(t_4) = l_2$. The transitions from SC will

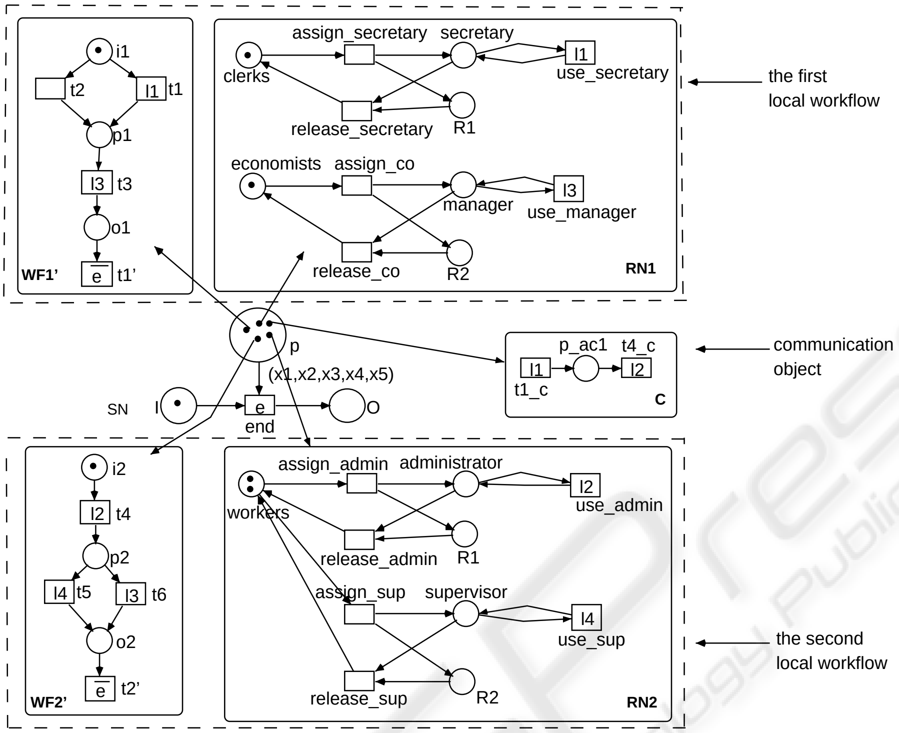

Fig. 1. A resource constrained interorganizational workflow net in its initial marking.

be assigned the same synchronous communication label: $\Lambda(t_3) = \Lambda(t_6) = l_3$. 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 object-net C describes the asynchronous communication between the local workflows. C is obtained from AC as follows: the set of places is $P_C = \{p_{ac_1}\}$, where $ac_1 = (t_1, t_4)$. The transitions (T_C) correspond to the transitions involved in asynchronous communication: $T_C = \{t_{1c}, t_{4c}\}$. Since $ac_1 = (t_1, t_4) \in AC$ then we will add the arcs (t_{1c}, p_{ac_1}) and (p_{ac_1}, t_{4c}). We have: $\Lambda(t_{1c}) = \Lambda(t_1) = l_1$ and $\Lambda(t_{4c}) = \Lambda(t_4) = l_2$.

In nested Petri nets, there are several firing rules ([10]): an unlabelled transition from an object-net can fire if the transition is enabled in the object-net (this is 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. The simultaneous firing of these transitions is called an horizontal synchronization step. A labelled transition enabled in SN should fire simultaneously with the transitions from the object-nets which have a complementary label (this is a vertical synchronization step). In our example, the transition end from SN should fire simultaneously with the transitions labelled with $\overline{e}$ in the object-nets.

In our example, t_1 in WF'_1, $use\_secretary$ in RN_1 and t_{1c} in C should fire at the same time, because they have the same label, l_1. But *use_secretary* is not enabled in RN_1 (i.e. there does not exist a *secretary* role available yet), so, although t_1 is enabled in WF'_1, it cannot fire yet. Transition t_4 is enabled in (WF'_2, i_2). t_4 can only fire at the same time with t_{4c} in C, but t_{4c} is not enabled in C, because t_{1c} (and t_1) has not fired yet. This behaviour is consistent with the restrictions specified in AC: t_4 will fire after the firing of t_1. The unlabelled transition *assign_secretary* is enabled in RN_1. The firing of this transition represents an object-autonomous step in $RIWF$ and produces a token in place $secretary$. In the resulting marking, the horizontal synchronization step $(; t_1, use\_secretary, t_{1c})$ can fire and it produces a new marking M_1 such that place I contains an atomic token and place p contains the object-nets with their corresponding new markings: $(WF'_1, m1_1)$ (with $m1_1 = 1'p_1$), (WF'_2, i_2), (C, mc_1) (with $mc_1 = 1'p_{ac_1}$), (RN_1, r_{11}) $(r_{11} = 1'economists + 1'secretary + 1'R_1)$, (RN_2, r_{20}). One can notice that t_3 in WF'_1, t_6 in WF'_2 and $use\_manager$ in RN_1 have the same label, so they can only fire at the same time. Thus, t_3 and t_6 fire synchronously, as specified by SC and t_3 fires only if there is a *manager* role available for its execution. t_6 does not need a role for its execution (there does not exist a transition labelled with l_3 in RN_2). The vertical synchronization step $(end; t'_1, t'_2)$ can only fire if t'_1 is enabled in WF'_1 and t'_2 is enabled in WF'_2. The firing of this step removes the atomic token from I, the object-nets from p and adds an atomic token to place O.

3 Definition of Resource Constrained Interorganizational Workflow Nets

In this section we first present a Petri net model for the resource perspective of a workflow, following the approach we used in [13]. A task that needs to be executed for a specific case is called a work item. Each work item should be performed by a resource suited for its execution. In order to facilitate the better allocation of resources to work items, resources are grouped into roles. Thus, instead of assigning work items directly to resources, work items will be assigned to certain roles. This pattern of representing and using resources is called "role-based allocation" ([9, 11, 14]). A *role* is a group of resources with similar characteristics. We consider that each resource has a general type. A resource can have more roles (at different moments in time) and each role can be performed by several resources of different types ([9]).

In our model, for each role one must specify the set of resource types that can be mapped onto that role. Based on these rules (which are specified at design time), the system will be able to allocate dynamically resources to the appropriate roles. Thus, a specification for the resource perspective consists in the following elements: a set of resource basic types: $RT = \{Type_1, \ldots, Type_n\}$. For each type $Type_i, i \in \{1, 2, \ldots, n\}$ there is a number n_i of resources of that type; a set of roles, $RO = \{Role_1, Role_2, \ldots, Role_m\}$; for each role $r \in RO$, $res(r)$ represents the resource types which can be assigned to the role ($res(r) \subseteq RT$).

A resource net $RN = (P_{RN}, T_{RN}, F_{RN})$ can be defined as follows:
- $P_{RN} = P_{RT} \cup P_{ROLE} \cup P'$ where $P_{RT} = RT$, $P_{ROLE} = RO$ and $P' = \{R_{ki} | Role_i \in RO, Type_k \in res(Role_i)\}$.
- $T_{RN} = \{assign_{ki}, release_{ik} | Role_i \in RO, Type_k \in res(Role_i)\}$.
- $F_{RN} = \{(Type_k, assign_{ki}), (assign_{ki}, Role_i), (assign_{ki}, R_{ki}), (R_{ki}, release_{ik}), (Role_i, release_{ik}), (release_{ik}, Type_k) | Role_i \in RO, Type_k \in res(Role_i)\}$.

In the resource net, P_{RT} corresponds to the set of resource types and P_{ROLE} corresponds to the set of roles. For each role $Role_i$ and for each resource type $Type_k \in res(Role_i)$ the following elements are added to the net : a place R_{ki}, a transition $assign_{ki}$ which moves a resource from $Type_k$ to role $Role_i$; a transition $release_{ik}$ which releases the resources of type $Type_k$, assigned to $Role_i$, when they are not needed any longer.

In what follows we will define a model, based on nested Petri nets, for loosely coupled interorganizational workflows. We will assume there are n local workflows which behave independently, but need to interact at certain points, according to a communication structure. There are two types of communication: asynchronous communication (corresponding to the exchange of messages) and synchronous communication.

We define, first, *extended workflow nets*, an extension of the WF-nets, which will be used for modelling the local workflows from the interorganizational workflow.

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 $F' = F \cup \{(o, t')\}$.*

Resource constrained interorganizational workflow nets (RIWF-nets) are defined as a special class of nested Petri nets.

Definition 2. *A resource constrained interorganizational workflow net RIWF is a nested Petri net: $RIWF = (Var, Lab, AC, SC, (C, 0), WF, RN, SN, \Lambda, Role)$ such that:*

1. *Var is a set of variables.*
2. *$Lab = Lab_{AC} \cup Lab_{SC} \cup Lab_{Res} \cup \{e, \overline{e}\}$ is a set of labels.*
3. *$WF = \{(WF'_1, i_1), \ldots, (WF'_n, i_n)\ \}$ is a set of extended WF-nets.*
4. *$RN = \{(RN_1, r_{10}), \ldots (RN_m, r_{m0})\}$ is the set of resource nets.*
5. *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 $i \neq j$.*
6. *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 $i \neq j$.*
7. *$C = (P_C, T_C, F_C)$ is the communication object:*
   – $P_C = \{p_{ac} | ac \in AC\}$.
   – $T_C = \{t_c | \exists (t', t) \in AC \vee (t, t') \in AC\}$.
   – $F_C = \{(p, t) \in P_C \times T^\circ | p = (t', t) \in AC\} \cup \{(t, p) \in T^\circ \times P_C | p = (t, t') \in AC\}$
8. *$SN = (N, W, M_0)$ is the system net of RIWF, such that:*
   - $N = (P_N, T_N, F_N)$ is a high level Petri net: $P_N = \{I, p, O\}$, $T_N = \{end\}$, $F_N = \{(I, end), (p, end), (end, O)\}$.
   - W is the arc labelling function: $W(I, end) = 1, W(p, end) = (x_1, x_2, \ldots, x_{n+m+1})$, $W(end, O) = 1$.
   - M_0 is the initial marking of the net: $M_0(I) = 1$, $M_0(p) = ((WF'_1, i_1), \ldots, (WF'_n, i_n), (RN_1, r_{10}), \ldots (RN_m, r_{m0}), (C, 0))$ and $M_0(O) = 0$.
9. *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* $\Lambda(end) = e$.
– $\forall t, t' \in T_i(i \in \{1, \ldots, n\}) : \Lambda(t) \neq \Lambda(t')$.

10. $Role$ *is a partial function which assigns to a labelled transition t (*$\Lambda(t) \in Lab_{Res}$*) from* $WF'_i \in WF$ *(*$t \neq t'$*) a role from a resource net* $RN_j \in RN$ *such that: if* $\Lambda(t) = l$ *and* $Role(t) = Role_k$ *then there exists a transition* t^* *in* RN_j *with* $\Lambda(t^*) = l$ *and* $(t^*, Role_k), (Role_k, t^*) \in F_{RN_j}$.

In a RIWF-net there are n extended WF-nets modelling the local workflows, m resource nets and a communication object, C. The set of all the object-nets is denoted by Obj. Var is the set of variables in the net, which 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 asynchronous communication elements, the labels from Lab_{SC} are used for synchronous communication elements and the labels from Lab_{Res} are used for labelling tasks in resource nets. The three sets are not necessary disjoint. AC represents the asynchronous communication relation: if $(t, t') \in AC$, then t must execute before t'. SC is the set of synchronous communication elements: if $x \in SC$, then, all the transitions from x have to execute at the same time. C is an object-net which describes the asynchronous communication between the local workflows: if $ac = (t, t') \in AC$, then there is a corresponding place p_{ac} in P_C, two transitions $t_c, t'_c \in T_C$ and two arcs $(t_c, p_{ac}), (p_{ac}, t'_c) \in F_C$. SN is a Petri net in which the tokens are either atomic tokens (without inner structure) or net-tokens. W is a function that assigns to each arc in SN an expression (a tuple of variables or the constant 1). Λ is a partial function which labels transitions from $RIWF$. If $x \in SC$, 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 C such that: $\Lambda(t) = \Lambda(t_c) = l, l \in Lab_{AC}$. $Role$ is a partial function which specifies the roles needed for executing certain tasks. Some tasks do not need roles for their execution. In our model, a task from a workflow can be executed by a role belonging to a different workflow. Also, as the number of resource nets may differ from the number of workflow nets, different workflows can share the same resource perspective.

We denote by A_{net} the net tokens of the RIWF-net: $A_{net} = \{(EN, m) \mathbin{/} m$ is a marking of EN, $EN \in Obj\}$. Then, a marking of a RIWF-net is a function such that: $M(I) \in \mathbb{N}$, $M(O) \in \mathbb{N}$ and $M(p) \in A_{net}^{n+m+1}$. We write M as a vector $M = (M(I), M(p), M(O))$.

A *binding* (of transition *end*) is a function $b : Var \to A_{net}$. If $expr$ is an expression, $expr(b)$ denotes the evaluation of $expr$ in binding b.

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

There are several types of steps, defining the behaviour of nested Petri nets (see [10]). In the case of RIWF-nets, we only focus on vertical synchronization, object-autonomous steps and horizontal synchronization.

There is only one vertical synchronization step in our case: 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 $M' = (0, 0, 1)$.

An object - autonomous step, in our case, represents the firing of an unlabelled transition in an object-net from the place p.

A horizontal synchronization step represents the simultaneous firing of the transitions with the same labels from object-nets: Let M be a marking of $RIWF$ and $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{m+n+1})$ the tuple of net-tokens from p. Assume $t_1, \ldots t_s$ is the set of all the transitions with the same label $l \neq \overline{e}$, $\Lambda(t_1) = \Lambda(t_2) = \ldots = \Lambda(t_s) = l$, such that: every transition t_j $(j \in \{1, \ldots, s\})$ is enabled in a net-token $\alpha_{k_j} = (EN_j, m_j)$ $(\{k_1, \ldots k_s\} \subseteq \{1, \ldots, m+n+1\}$, $EN_j \in Obj$) 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 α from place p with the tuple $(\alpha'_1, \alpha'_2, \ldots, \alpha'_{m+n+1})$, where $\alpha'_{k_j} = (EN_j, m'_j), \forall j \in \{1, \ldots, s\}$ and $\alpha'_i = \alpha_i, \forall i \in \{1, \ldots, m+n+1\} \setminus \{k_1, \ldots k_s\}$. We write: $M[; t_1, \ldots, t_s\rangle M'$.

4 The Soundness Property for Resource Constrained Interorganizational Workflow Nets

In this section we will introduce a notion of soundness for RIWF-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 ($(\forall m)((i[*\rangle m) \implies (m[*\rangle o))$), any case must terminate correctly $(\forall m)((i[*\rangle m) \wedge m \geq o) \implies (m = o))$, and every task should contribute to at least one possible execution of the workflow $(\forall t \in T)(\exists m, m')(i[*\rangle m[t\rangle m')$.

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

An extended workflow net WF' is sound if its underlying net,WF, is sound.

In an interorganizational workflow, although the local workflows are sound, we can have synchronization errors and interlockings. We will define a notion of soundness for interorganizational workflows. The final state for a RIWF-net is a marking M_f, in which there is only one atomic token in place O: $M_f = (0, 0, 1)$. A RIWF-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, there is a firing sequence that leads to M_f. We can define formally the notion of soundness for a RIWF-net as follows:

Definition 3. *A RIWF-net is sound if and only if:*

1. (WF'_j, i_j) *is a sound extended workflow net,* $\forall j \in \{1, \ldots, n\}$.
2. $(\forall M)((M_0[*\rangle M) \implies (M[*\rangle M_f))$.

The second condition from the 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 and the resources involved.

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

Definition 4. *Let $RIWF$ be a RIWF-net, M_1 and M_2 markings of $RIWF$. $M_1 \preceq M_2$ 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 : $M_1(p) \rightarrow M_2(p)$, such that for $\alpha = (\alpha_1, \ldots, \alpha_{n+m+1}) \in M_1(p)$ and for $J_p(\alpha) = \alpha' = (\alpha'_1, \ldots \alpha'_{n+m+1})$ we have for $i \in \{1, \ldots, n+m+1\}$ either $\alpha_i = \alpha'_i$ or $\alpha_i = (EN, m)$ and $\alpha'_i = (EN, m')$ $(EN \in Obj)$ and for all the places q of EN: $m(q) \leq m'(q)$.*

Let $RIWF$ be a RIWF-net and M and M' two markings of $RIWF$. The marking M covers M' (w.r.t. the partial ordering $\preceq$) if $M' \preceq M$.

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 [10] that the inevitability problem is decidable for nested Petri nets.

Theorem 1. *Let $RIWF$ be a RIWF-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 $RIWF$. 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 $M'(I) = 0$. 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 $RIWF$. 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 RIWF - nets.*

Proof: Let $RIWF$ be a RIWF-net. $RIWF$ is sound if and only if: (1) WF'_i are sound, $\forall i \in \{1, \ldots, n\}$ and (2) for any reachable marking in $RIWF$, $M \in [M_0\rangle$, 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 $Q = \{(1, 0, 0)\}$.

5 Conclusions

The approach we propose in this paper has several advantages: one can have a modular view on the interorganizational workflow; steps in RIWF- nets can easily express the synchronous and the asynchronous communication; RIWF-nets represent a flexible model for interorganizational workflows, because any component can be modified easily, with minimal changes to the other components. A notion of soundness was introduced for RIWF-nets and we proved this property is decidable. Future work aims at defining a specification language based on XML, which will then be translated into RIWF-nets, in order to check the soundness and other behavioural properties of the interorganizational workflow. We intend to develop a tool based on this language and on RIWF-nets, for executing interorganizational workflows. We will also study the case 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: *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.
3. 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.
4. 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.
5. 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): pp. 6183, 1998.
6. T.L. Casavant, J.G. Kuhl. *A Communicating Finite Automata Approach to Modeling Distributed Computation and its Application to Distributed Decision-Making*. IEEE trans. on computers, May 1990,Vol. 39, No. 5, pp. 628-639.
7. Jiang Guo. *Using Category Theory to Model Software Component Dependencies*. Proc. of 9th Annual IEEE Intl. Conf. and Workshop on the Engineering of Computer-Based Systems(ECBS 2002) ,Lund, Sweden, IEEE Computer Society Press, April 08 - 11, 2002, pp. 185-194.
8. G. Kramler , W. Retschitzegger. *Specification of Interorganizational Workflows - A Comparison of Approaches*. Vienna, Austria: Institute of Software Technology and Interactive Systems, Business Informatics Group, Vienna University of Technology, 2002. Technical Report, 08/02.
9. A. Kumar, W.M.P. van der Aalst, H.M.W. Verbeek: *Dynamic Work Distribution in Workflow Management Systems: How to balance quality and performance?*. Journal of Management Information Systems, 18(3), pp.157-193, 2002.
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. M. zur Mühlen. *Resource modeling in workflow applications*. Proceedings of the 1999 Workflow Management Conference (November 1999), pp. 137-153, 1999.
12. M. Netjes, W.M.P. van der Aalst, H.A. Reijers. *Analysis of resource-constrained processes with colored petri nets*. In K. Jensen, Proceedings of the 6th Workshop and Tutorial on Practical Use of Coloured Petri nets and the CPN Tools (CPN'05), pp. 251-265, 2005
13. O. Prisecaru: *Resource Workflow Nets:a Petri Net Formalism for Workflow Modelling*. In Proc. of MSVVEIS 2007, pp. 11-21, Portugal, 12-13 June 2007
14. I.T.P. Vanderfeesten, W.M.P. van der Aalst, H.A. Reijers. *Modelling a product based workflow system in CPN Tools*. In K. Jensen, Proceedings of the 6th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pp. 99-118, Aarhus, Denmark: University of Denmark, 2005.
15. H. M. W. Verbeek, A. Hirnschall, Wil M. P. van der Aalst. *XRL/Flower: Supporting Interorganizational 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.