Resource Workflow Nets: A Petri Net Formalism for Workflow Modelling

Oana Otilia Prisecaru

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

Abstract. A workflow is the automation of a business process that takes place inside one organization. While most of the formal approaches to workflow modelling consider only the process perspective, we propose a Petri net model which integrates both the process and the resource perspective. The paper introduces a special class of nested Petri nets, resource workflow nets (RWFN-nets), which unifies the two perspectives into a single model. Unlike other models, RWFN-nets permit a clear distinction between the perspectives, modelling efficiently their interaction, and ensure the flexibility of the system. The paper also defines a notion of behavioural correctness for RWFN-nets, soundness, and proves this property is decidable.

1 Introduction

Over the last few years workflow technology has developed rapidly and has been increasingly used in many large organizations. A workflow is a complex process, consisting of activities organized in order to accomplish some goal. 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 population in which the workflow is executed (the resources) and the existing roles (resource classes based on organizational or functional aspects).

A formal method which has been successfully used for workflow modelling is Petri nets. Most of the current researches have been focused on the modelling of the process perspective of workflows. A Petri net model for workflows, which includes resources, can be found in [5, 6] where special places are used for representing resources in the process perspective. While the resource perspective is represented in a simplistic manner, this approach defines and studies a soundness notion for workflows. A more detailed view on the resource perspective is offered in papers like [11, 12], where coloured Petri nets are used in order to model a work distribution system, but no correctness notion is discussed.

The main problem with the approaches described above is that they either model the resource perspective in a simplistic manner, or they fail to solve verification problems for workflows. Also, there is an unclear mixture of perspectives, which can make workflow specifications difficult to understand, analyze and work with.

In order to tackle these problems, this paper proposes a special class of nested Petri nets, which will be used for the integrated modelling of the process and of the resource perspective of workflows, permitting a clear distinction between them. Nested Petri nets, which were introduced in [8], are a special class of the Petri net model, in which tokens may be nets themselves (object-nets). *Resource workflow nets (RWFN-nets)* are introduced as a special case of two-level nested Petri nets, in which the two perspectives are modelled as two separate object-nets: one object-net is a Petri net which models the resource perspective and the other is a Petri net which models the process perspective. The dynamic behaviour of the RWFN-net ensures the collaboration between perspectives.

For the modelling of the process perspective we will use *extended workflow nets*, a slightly modified version of workflow nets, introduced in [1]. In order to model the resource perspective we introduce *resource nets*, a Petri net model which will be able to describe the existing resources and roles, the allocation of resources to specific roles (according to predefined rules) and the release of resources from roles. The two object-nets synchronize whenever a task from the workflow net uses a role of the resource net and they behave independently otherwise. The paper also introduces a notion of behavioural correctness for RWFN-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 modelling the process perspective of workflows (for details the reader is referred to [1]). We assume the reader is familiar with the Petri net terminology and notation.

In the process perspective, the workflow processes are instantiated for a specific case (or workflow instance). In [1] a special class of Petri nets is introduced for modelling the process perspective: *workflow nets (WF-nets)*. 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* $o\bullet = \emptyset$*. (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 Petri net (and 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 remain of the paper is organized as follows: Section 2 introduces the definition of resource nets, Section 3 defines RWFN-nets, Section 4 defines and studies the soundness property for RWFN-nets, Section 5 presents a short example of a RWFN-net and Section 6 concludes the paper.

2 The Modelling of the Resource Perspective Using Petri Nets

This section introduces a Petri net model for the resource perspective. This perspective centers on the modelling of resources and their interaction with the process perspective. There is a limited number of resources available for executing the tasks of the workflow. 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 way (pattern) of representing and using resources is called ”role-based allocation” ([7, 10, 12]).

A *role*, also referred to as a resource class, 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 ([7]).

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 ($res(r) \subseteq RT$).

Given the elements above, 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:
  - $P_{RT} = RT$, $P_{ROLE} = RO$.
  - $P' = \{R_{ki} | Role_i \in RO, Type_k \in res(Role_i)\}$.
- $T_{RN} = \{assign_{ki}, release_{ik} | Role_i \in RO, Type_k \in res(Role_i)\}$.
- $F_{RN} = \{(Type_k, assign_{ki}), (assign_{ki}, Role_i), (assign_{ki}, R_{ki}), (R_{ki}, release_{ik}), (Role_i, release_{ik}), (release_{ik}, Type_k) | Role_i \in RO, Type_k \in res(Role_i)\}$.

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 (see Fig. 1): a place R_{ki}, which will be used for the proper release of resources; 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 the initial marking of the net, in every place $Type_i$, there will be a number of tokens equal to the number of resources of that type.
One can notice that the Petri net model we propose abstracts from the interaction with the process perspective. In the next section, for every task in the workflow that needs the role $Role_i$ for its execution, a new transition will be added to the resource net (transition use_i in Fig. 1).

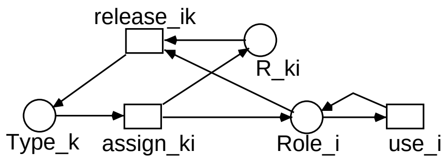

Fig. 1. A resource workflow net in its initial marking.

3 The Definition of Resource Workflow Nets

This section introduces a special class of nested Petri nets, which will model a workflow which incorporates both the process perspective and the resource perspective.

We define, first, *extended workflow nets*, an extension of the WF-nets defined in Sect. 1.

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

Nested Petri nets are high level Petri nets which can have as tokens ordinary Petri nets. A nested Petri net consists of a system net (a high level Petri net with expressions on arcs) and object Petri nets.

Definition 2. *A Resource Workflow Net is a two-level nested Petri net* $RWFN = (Var, Lab, (WF', i), (RN, m_0), SN, \Lambda, Role)$ *such that:*

1. $Var = \{x, y\}$ *a set of variables.*
2. Lab *- a set of net labels.*
3. $(WF', i), (RN, m_0)$ *are the token (object) nets:*
   – (WF', i) *is an extended workflow net with its initial marking.*
   – (RN, m_0) *is a resource net with its initial marking.*
4. $SN = (N, W, M_0)$ *- the system net of RWFN, such that:*
   – $N = (P, T, F)$ *is a high level Petri net such that*
     • $P = \{I, p, O\}$*, where O is a place such that* $O\bullet = \emptyset$ *and I is a place such that* $\bullet I = \emptyset$.
     • $T = \{end\}$.
     • $F = \{(I, end), (p, end), (end, O)\}$.
   – M_0 *is the initial marking of the net, in which there exists a single atomic token in place I and the place* p *contains the pair* $((WF', i), (RN, m_0))$.
   – W *is the arc labelling function:* $W(I, end) = 1$, $W(p, end) = (x, y)$, $W(end, O) = 1$.
5. Λ *is a partial function which assigns to certain transitions from the nets* WF', RN, SN*, a label from the set* Lab*, and* $\Lambda(end) = e$, $\Lambda(t') = \overline{e}$.
6. $Role$ *is a partial function which assign to every labelled transition* t *from* WF' *(*$t \neq t'$*) a role from* RN *such that: if* $\Lambda(t) = l$ *and* $Role(t) = Role_i$ *then there exists a transition* t^* *in* RN *with* $\Lambda(t^*) = l$ *and* $(t^*, Role_i), (Role_i, t^*)$ *are arcs in* RN.

There are only two object-nets in a RWFN-net: (WF', i) is an extended workflow Petri net and (RN, m_0) is the resource net which describes the resource perspective. Variables x and y will be assigned a certain value at runtime: each variable can take as value an object-net in a certain marking. The system net, SN, is a high level Petri net. Tokens in SN can be atomic tokens, without inner structure, or net-tokens (the two object-nets). W is a function that assigns to each arc in SN an expression. In RWFN-nets, an expression can be either the pair (x, y) or the constant 1. Λ is a partial function that labels transitions from the two object-nets and the transition of the system net. The labelled transitions from WF (the underlying WF-net of WF') represent the tasks that need roles from the resource net. $Role$ is a partial function which assigns to every task (labelled transition) t in WF, a role $Role_i$ from the resource net. This function designates the role that can execute this task.

A workflow is modelled using RWFN-nets in the following manner: first, the process perspective (the tasks that need to be executed and their order of execution) is modelled using an extended WF-net. The resource perspective is modelled separately using a resource net. For each task that needs a certain role for its execution, a new transition is connected with the place corresponding to that role, in the resource net. The task and the added transition have the same label. A simple example of a RWFN-net is presented in Fig. 2.

We denote by A_{net} the net tokens of the RWFN-net:
$A_{net} = \{(WF', m_1), (RN, m_2)$ / m_1 is a marking of WF', m_2 is a marking of $RN\}$.

A marking of a RWFN-net is a function such that: $M(I)$ and $M(O)$ are natural numbers, $M(p) \in A_{net} \times A_{net}$. We write M as a vector $M = (M(I), M(p), M(O))$.

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

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

The firing of end produces a new marking, M': $M[end[b]\rangle M'$, such that, for every place q: $M'(q) = (M(q) - W(q,t)(b)) \cup W(t,q)(b)$.

There are three types of steps in a RWFN-net:

Definition 5. *A vertical synchronization step:*
If transition end is enabled in a marking M w.r.t. a binding b and t' is enabled in the object-net WF', then the simultaneous firing of end and t' is a vertical syncronization step.
This step removes the two object-nets from p. In the resulting marking, M', there is only one atomic token in place O. We write $M[end[b]; t'\rangle M'$.

Definition 6. *An object - autonomous step:*
Let M be a marking of a RWFN-net and (α_1, α_2) a pair of tokens from p. Let α_i be one of the two object-nets ($i \in \{1, 2\}$) $\alpha_i = (WF', m)$ or $\alpha_i = (RN, m)$. 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 RWFN-net obtained from the old marking M by replacing in the tuple (α_1, α_2) from $M(p)$ the net token α_i with the net token α'_i, where $\alpha'_i = (WF', m')$ or $\alpha'_i = (RN, m')$. We write: $M[;t\rangle M'$.

M' is a marking of the RWFN-net obtained from M by the firing of a local transition in one of the object-nets. We notice that none of the object nets are moved from the place p.

Definition 7. *A horizontal synchronization step:*
Let M be a marking of RWFN and (α_1, α_2) a tuple of net-tokens from p. Assume, for instance, that $\alpha_1 = (WF', m_1)$, $\alpha_2 = (RN, m_2)$. Let t_1 be a transition in WF such that $m_1[t_1\rangle m'_1$ (by means of classical Petri nets) and $\Lambda(t_1) = l$. Let t_2 be a transition in RN such that $m_2[t_2\rangle m'_2$ (by means of classical Petri nets) and $\Lambda(t_2) = l$, $l \in Lab$. The synchronous firing of t_1 and t_2 is called a horizontal synchronization step.
The resulting marking, M', is obtained from M by replacing the net tuple (α_1, α_2) from place p with tuple (α'_1, α'_2), where $\alpha'_1 = (WF', m'_1)$, $\alpha'_2 = (RN, m'_2)$. We write: $M[t_1; t_2\rangle M'$.

M' is the marking obtained by the simultaneous firing of t_1 in (WF', m_1) and t_2 in (RN, m_2), while both object-nets remain in the same place of SN.

A *firing sequence* from M to M' is a sequence of steps $Y_1, \ldots, Y_n$ such that $M[Y_1\rangle M_1[\ldots[Y_n\rangle M'$. If there is a firing sequence from M to M' we can write $M[*\rangle M'$ or $M \xrightarrow{*} M'$. We say M' is reachable from M and write $M' \in [M\rangle$.

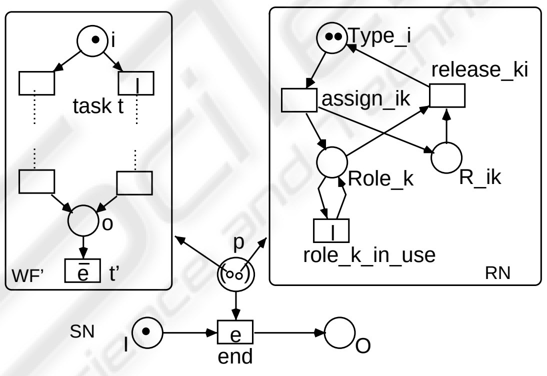

Fig. 2. A resource workflow net in its initial marking.

4 The Soundness of Resource Workflow Nets

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

A notion of soundness was defined for WF-nets, expressing the minimal conditions a correct workflow should satisfy ([1, 4]). An extended workflow net WF' is sound if the underlying WF-net is sound.

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): $(\forall m)((i[*\rangle m) \implies (m[*\rangle o))$.*
2. *Marking o is the only marking reachable from state i with at least one token in place o: $(\forall m)((i[*\rangle m) \wedge m \geq o) \implies (m = o))$.*
3. *There are no dead transitions in WF: $(\forall t \in T)(\exists m, m')(i[*\rangle m[t\rangle m')$.*

We will consider the final state for a RWFN-net, a marking M_f, in which there is only one atomic token in place O: $M_f = (0, 0, 1)$. A RWFN-net is sound if: (1) WF' is sound and (2) for any reachable marking of the RWFN-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 a RWFN-net as follows:

Definition 9. *A RWFN-net $RWFN$ is sound if and only if:*

1. *(WF', i) is a sound workflow net.*
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: $(\forall M)((M_0[*\rangle M) \implies (M[*\rangle M_f))$.*

First, we consider the workflow is sound if the WF-net describing the process is sound (abstracting from resources). The final marking of the RWFN-net is reached if and only if the vertical synchronization step fires. This implies that transition t' is enabled in WF', which happens if and only if the final marking of the WF-net has been reached. Thus, the second condition from the soundness definition basically states that the workflow is sound if the termination condition still holds in the WF-net, when the firing of tasks is restricted by the resource perspective.

The notion of soundness for RWFN-nets is weaker than the notion of soundness defined in [1], as it does not impose the absence of dead steps in the RWFN-net. If a RWFN-net is sound, one can easily see that marking M_f is the only marking reachable from M_0 with at least one token in place O. This property is similar to condition (2) in Def. 8.

In order to decide whether the soundness property defined is decidable, we introduce a partial order on the markings of the RWFN - net (see [8]):

Definition 10. *Let $RWFN$ be a RWFN-net, M_1 and M_2 markings of $RWFN$. $M_1 \preceq M_2$ 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 $\alpha = (\alpha_1, \alpha_2) \in M_1(p)$ and for $J_p(\alpha) = \alpha' = (\alpha'_1, \alpha'_2)$ we have for $i \in \{1, 2\}$ either $\alpha_i = \alpha'_i$ or $\alpha_i = (EN, m)$ and $\alpha'_i = (EN, m')$ $(EN \in \{WF', RN\})$ and for all the places q of EN: $m(q) \leq m'(q)$.*

Definition 11. *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 [8, 9] that the inevitability problem is decidable for nested Petri nets.

Theorem 1. *Let $RWFN$ be a RWFN-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. If $M[*\rangle M_f$ in $RWFN$, we can consider $M' = M_f$.
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 $Y = (end[b]; t')$ fires in $RWFN$. 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 RWFN - nets.*

Proof. Let $RWFN$ be a RWFN-net. Using the definition of soundness and Theorem 1, $RWFN$ is sound if and only if: (1) WF is sound and (2) for any reachable marking in $RWFN$, $M \in [M_0\rangle$, 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 WF-nets 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 An Example of a Resource Workflow Net

The example in Fig. 3 presents a RWFN-net modelling a workflow which processes credit requests in a bank. We assume there are two types of resources (clerks and economists) and two possible roles (secretary and credit officer). A *secretary* role can be performed by a clerk and a *credit officer* role can be performed by an economist. The specification for the resource perspective is:
RT = {clerks,economists}, RO = {secretary,credit_officer}, res(secretary)={clerks}, res(credit_officer)={economists}. In the process perspective, described by the extended workflow net WF', when a request for a credit appears, the first transition to fire is *register_request*. A *secretary* role is needed for the execution of this task. After this, one of the transitions *aprove_credit* or *deny_credit* can fire (a *credit_officer* role is needed for their execution). Finally, transition *send_answer* fires (a *secretary* role sends the answer to the client). The function $Role$ is defined as follows: *Role(register_request)=secretary, Role(aprove_credit)=credit_officer, Role(deny_credit)=credit_officer, Role(send_answer)=secretary*. From the specification and from the $Role$ function results a resource net RN (Fig. 3). We consider the following marking m_0 for RN: $m_0 = 1'clerks + 1'economists$. In the initial marking of the net, $M_0(I) = 1$, $M_0(p) = ((WF', i), (RN, m_0))$. Several object-autonomous steps (the firing of unlabelled transitions from RN) are possible in M_0. Let assume that transition $assign\_secretary$ fires in the resource net RN. This is a role-allocation transition in the resource net. The new marking of the RWFN-net is $M_1 = (1, ((WF', i), (RN, m_1)), 0)$, where $m_1 = 1'secretary + 1'R_1 + 1'economists$. When a request appears, the first task to execute is *register_request*, which can only fire simultaneously with transition *use_secretary* in RN. In marking M_1 of the RWFN-net, the synchronization step described above can

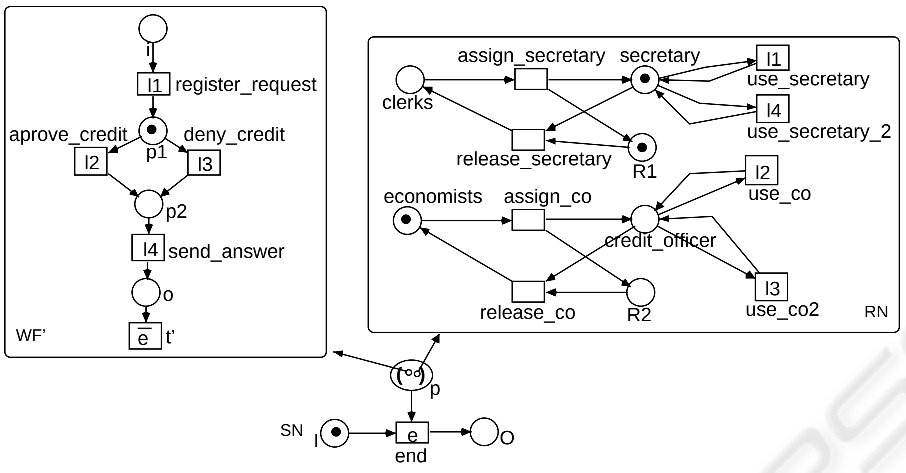

Fig. 3. A resource workflow net in its initial marking.

fire: transition *register_request* is enabled in (WF', i) and transition *use_secretary* is enabled in (RN, m_1). After the firing of this step, a new marking, M_2 (see Fig. 3), results in the resource workflow net: $M_2 = (1, ((WF', m_1'), (RN, m_2)), 0)$, where $m_1' = 1'p_1$ and m_2 is the marking of RN obtained from marking m_1 by the firing of *use_secretary* ($m_2 = m_1$). The system remains blocked (no other task in the process is executed) unless the resource allocation system allocates resources (economists) for the *credit officer* role. Transition *assign_co* can fire independently in RN. The resulting marking in the RWFN-net is $M_3 = (1, ((WF', m_1'), (RN, m_3)), 0)$, where $m_3 = 1'secretary +$ $1'$*credit_officer* $+ 1'R_2 + 1'R_1$. In M_3, transition *aprove_credit* in WF' can fire simultaneously with transition *use_co* in RN. After the firing of this synchronization step, the resulting marking of the RWFN-net is $M_4 = (1, ((WF', m_2'), (RN, m_4)), 0)$ where $m_2' = 1'p_2$ and $m_4 = m_3$. Next, transition *send_answer* from WF' can fire simultaneously with transition *use_secretary_2*. After the firing of this synchronization step, the resulting marking of the RWFN-net is $M_5 = (1, ((WF, m_3'), (RN, m_5)), 0)$ where $m_5 = m_4$ and $m_3' = 1'o$. One can notice that the vertical synchronization step $Y = (end; t')$ is enabled in marking M_5 with the binding b: $b(x) = (WF, m_3'), b(y) = (RN, m_5)$ and the firing of this step removes the two net tokens from place p. The RWFN-net has reached a final state and terminated correctly. This RWFN-net is a sound RWFN-net.

6 Conclusion

This paper presented a special class of nested Petri nets used to model both the resource perspective and the process perspective for a workflow. The two perspectives are represented as two independent object-nets which synchronize whenever a task from the workflow net uses a role of the resource net. The advantage of this approach is that it integrates both perspectives but it keeps a clear difference between them: unlike other approaches that use Petri nets, resources and roles are not represented in the same Petri net as the process. This allows a clear distinction between the two perspectives and a flexible workflow system: changes in the two perspectives can be done easily, with minimal changes in the other perspective. A notion of soundness was introduced and we proved this property is decidable for RWFN-nets. Future work aims at defining RWFN-nets which will model workflows that process batches of cases, instead of one case in isolation.

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: *Structural Characterization of Sound Workflows nets*, Computing Science Reports 96-23, Eindhoven University of Technology, Eindhoven, 1996.
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: *Verifications 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.
5. K. Barkaoui, L. Petrucci: *Structural analysis of workflow nets with shared resources*. Workflow management: Net-based Concepts, Models, Techniques and Tools (WFM'98), volume 98/7 of Computing science reports, pp. 82-95, Eindhoven University of Technology, 1998.
6. K. van Hee, N. Sidorova, M. Voorhoeve: *Resource-Constrained Workflow Nets*. Lindemann, Burkhard, Czaja, Skowron, Schlingloff, Suraj (Eds.): Proceedings of the International Workshop on Concurrency, Specification and Programming (CS & P) 2004, pp. 166-177, Informatik-Berichte der Humboldt-Universitat zu Berlin, Nr. 170, September 2004.
7. 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.
8. I.A. Lomazova: *Nested Petri Nets - a Formalism for Specification and Verification of Multi - Agent Distributed Systems*. Fundamenta Informaticae 43 pp. 195-214, 2000.
9. I. A. Lomazova, Ph. Schnoebelen: *Some Decidability Results for Nested Petri Nets*. Ershov Memorial Conference: pp. 208-220, 1999.
10. M. zur Mühlen. *Resource modeling in workflow applications*. Proceedings of the 1999 Workflow Management Conference (November 1999), pp. 137-153, 1999.
11. 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
12. 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.
13. WFMC, *Workflow Management Coalition Terminology and Glossary (WFMC-TC-1011)*, Technical Report, The Workflow Management Coalition, Brussels 1999.