

The Analysis of Resource Constrained Workflows using Petri Nets

Oana Otilia Prisecaru

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

Abstract. A workflow describes a complex process that takes place inside an organization. A workflow can be structured into several perspectives. In order to model both the process and the resource perspective of workflows, a Petri net model based on nested Petri nets has been proposed: resource workflow nets (RWF-nets). Unlike other models, RWF-nets permit a clear distinction between the perspectives, modelling efficiently their interaction, and ensure the flexibility of the system. A case (or workflow instance) is the subject of the operations in the workflow. RWF-nets permit the handling of one case at a time. This paper extends the definition of RWF-nets in order to allow the handling of multiple cases at a time, defines a notion of behavioural correctness for RWF-nets, k -soundness, and proves the decidability of this property for a special class of RWF-nets.

1 Introduction

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 workflow management system (WFMS) is a software system that supports the modelling and execution of workflows. WFMS's can use different modelling languages for the definition of workflows. A formal method which has been successfully used for workflow modelling is Petri nets. Most of the current research has focused on the modelling of the process perspective of workflows. A Petri net model for workflows, which includes resources, can be found in [3, 4] where special places are used for representing resources in the process perspective. While no allocation mechanisms are represented, this approach defines and studies a soundness notion for workflows. A more detailed view on the resource perspective is offered in papers like [10, 14], where coloured Petri nets are used in order to model a work distribution system. The approach in [7] allows the modelling of resources and the handling of several cases; In [13], resource-extended stochastic workflow nets allow the performance analysis of the workflows. None of these approaches study the logical correctness of the workflows. Thus, the existing approaches 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, in [11, 12], we proposed a special class of nested Petri nets - Resource Workflow Nets (RWF-nets), for the integrated modelling of the process and of the resource perspective of workflows. Nested Petri nets ([8]) are a special class of the Petri net model, in which tokens may be nets themselves (object-nets). RWF-nets are defined 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 process perspective is modelled using *extended workflow nets*, an extension of workflow nets, introduced in [1]. The resource perspective is modelled using *resource nets*, a Petri net model which describes 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. A RWF-net describes the handling of one case at a time, where a case is the subject of the operations in the workflow. A notion of behavioural correctness was defined and proved decidable for RWF-nets.

In workflow management systems several instances of a workflow (cases) are executed simultaneously, hence it is important to verify if the soundness criteria are also met in this situation. In this paper we extend the definition of RWF-nets in order to allow the handling of several cases at a time and we define the notion of k -soundness in order to describe the correct behaviour of RWF-nets for this situation. We prove that in the case the resource net is live, the k -soundness of the RWF-net is equivalent with the k -soundness of the extended workflow net and thus, decidable.

The remainder of the paper is organized as follows: Section 2 introduces the definition of resource nets and RWF-nets, Section 3 defines and studies the k -soundness property for RWF-nets and Section 4 concludes the paper.

2 The Modelling of the Resource Perspective using Petri Nets

2.1 Preliminaries

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. A workflow net (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 are k tokens in the source place represents the beginning of the processing for k cases (the initial marking of the net, denoted by $i.k$). The marking in which there are k tokens in the sink place, represents the end of the processing for the k cases (and the final marking of the net, denoted by $o.k$). An additional requirement is that every element of the workflow net should be on a path from i to o .

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 resulting Petri net is strongly connected.

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

We will present the Petri net model used for describing the resource perspective, defined in [11, 12]. The resource perspective defines the existing resources and their interaction with the process perspective. 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" ([6, 9, 14]).

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 ([6]).

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: $RT = \{Type_1, \dots, Type_n\}$. For each type $Type_i, i \in \{1, 2, \dots, n\}$ there is a number n_i of resources of that type.
- A set of roles, $RO = \{Role_1, Role_2, \dots, Role_m\}$.
- 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: $P_{RT} = RT, P_{ROLE} = RO$ and $P' = \{R_{ki} | Role_i \in RO, Type_k \in res(Role_i)\}$.
- $T_{RN} = \{assign_{ki}, release_{ik} | Role_i \in RO, Type_k \in res(Role_i)\} \cup T_{Rem}$, where T_{Rem} is a set of transitions which can remove resources.
- $F_{RN} = \{(Type_k, assign_{ki}), (assign_{ki}, Role_i), (assign_{ki}, R_{ki}), (R_{ki}, release_{ik}), (Role_i, release_{ik}), (release_{ik}, Type_k) | Role_i \in RO, Type_k \in res(Role_i)\} \cup F_{Rem}$, where $F_{Rem} \subseteq P_{RT} \times T_{Rem}$.

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. The transitions in the set T_{Rem} can model the situation in which certain resources become permanently unavailable in the workflow.

One can notice that the Petri net model we propose abstracts from the interaction with the process perspective. In the Petri net model which integrates both perspectives,

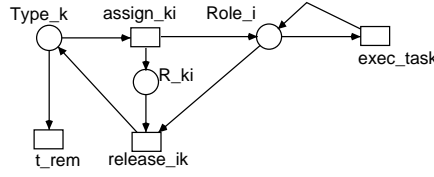


Fig. 1. A resource workflow net in its initial marking.

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 $exec_task$ in Fig. 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 $F' = F \cup \{(o, t')\}$.

2.2 Resource Workflow Nets

In what follows we will define *resource workflow nets* (RWF-nets). We will extend our approach from [11, 12], in order to allow the handling of several cases simultaneously in the workflow. RWF-nets will be defined as a special class of nested Petri nets, in which there exist only two object-nets, together with a function $Role$. Nested Petri nets are 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 1. A *Resource Workflow Net* is a two-level nested Petri net together with a function $Role$: $RWFN = (Var, Lab, (WF', i.k), (RN, rm_0), SN, \Lambda, Role)$:

1. $Var = \{x, y\}$ is a set of variables.
2. $Lab = Lab_h \cup Lab_v$ is a set of net labels such that $Lab_v = \{e, \bar{e}\}$.
3. $(WF', i.k), (RN, rm_0)$ are object-nets: $(WF', i.k)$ is an extended WF-net with its initial marking and (RN, m_0) is an extended resource net with its initial marking.
4. $SN = (N, W, M_0^k)$ is the system net of RWFN, such that:
 - $N = (P_N, T_N, F_N)$ is a high level Petri net such that:
 - $P_N = \{I, p, O\}$, where O is a place such that $O \bullet = \emptyset$ and I is a place such that $\bullet I = \emptyset$.
 - $T_N = \{end\}$.
 - $F_N = \{(I, end), (p, end), (end, p), (end, O)\}$.
 - M_0^k is the initial marking of the net, in which there exist k atomic tokens in place I , place p contains the pair $((WF', i.k), (RN, m_0))$ and place O is empty.
 - W is the arc labelling function: $W(I, end) = 1$, $W(p, end) = W(end, p) = (x, y)$, $W(end, O) = 1$.
5. Λ is a partial function which assigns to certain transitions from the nets WF' , RN , SN , a label from the set Lab , and: $\Lambda(end) = e$, $\Lambda(t') = \bar{e}$. For any transition t in WF' , $t \neq t'$, such that $\Lambda(t)$ is defined, there is a transition $t_r \in T_{RN}$ such that $\Lambda(t) = \Lambda(t_r) \in Lab_h$.
6. $Role$ is a partial function which assigns 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 t^* in RN with $\Lambda(t^*) = l$ and $(t^*, Role_i), (Role_i, t^*)$ are arcs in RN .

There are two object-nets in a RWFN-net: $(WF', i.k)$ is an extended WF-net which models the process perspective and (RN, rm_0) describes the resource perspective: RN is a resource net to which some labelled transitions are added in order to ensure the interaction with the process perspective. Variables x and y will be assigned certain values at runtime: the possible values for these variables are the object-nets (in certain markings). x has the net type WF' and y has the net type RN . In SN , the places I and O will hold atomic tokens, while p will hold a pair of net-tokens. There is only one constant, 1, for the arcs (I, end) , (end, O) , which is interpreted as an atomic token. $Role$ is a partial function which assigns to every task (labelled transition) t in WF , a role $Role_i$ from RN . This function designates the role that can execute this task. Λ is a partial function which labels the transitions of the object-nets and of the system-net. If t is a labelled transition in WF' , $t \neq t'$ and t needs a role $Role_i$ for its execution (i.e. $Role(t) = Role_i$), then there exists a corresponding transition in RN , connected to $Role_i$ with the same label as t . Also, t' and end have complementary labels.

A workflow is modelled using RWF-nets in the following manner: first, the process perspective 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.

We denote by A_{net} the set of net-tokens (marked object-nets): $A_{net} = \{(EN, m) \mid EN \in \{WF', RN\}, m \text{ is a marking of } EN\}$. A marking of a resource workflow net can be represented as a vector $M = (M(I), M(p), M(O))$, where $M(I), M(O) \in \mathbb{N}$ and $M(p) \in \mathbb{N}^{A_{net}}$. A binding b for the transition end is a function $b : \{x, y\} \rightarrow A_{net}$. b assigns net-tokens to the variables of the net. Transition end from the system net SN of a RWF-net is enabled in a marking M w.r.t. a binding b if I contains at least an atomic token and $W(p, end)(b) = (x, y)(b) = (b(x), b(y)) \in M(p)$.

The steps that can occur in resource workflow nets are those defined for two-level nested nets ([8]): The firing of an unlabelled transition, which is enabled in the marking of RN or of WF' , represents an object-autonomous step; A labelled transition enabled in the marking of WF' should fire at the same time with a transition with the same label enabled in the marking of RN . The simultaneous firing of these two transitions represents a horizontal synchronization step. In the resulting marking of the RWF-net, both the marking of WF' and RN will change; If end is enabled in SN w.r.t. a binding b and t' is enabled in WF' , the synchronous firing of end and t' represents a vertical synchronization step. The firing of the vertical synchronization step in a certain binding b ($b(x) = (WF', m), b(y) = (RN, rm)$) removes the pair of net-tokens $((WF', m), (RN, rm))$ from place p and then adds back to p the pair of net-tokens : $((WF', m'), (RN, rm))$, where m' is the marking obtained in WF' by firing the transition t' . An atomic token will be added to place O .

The set of all steps in a RWF-net is denoted by \mathcal{Y} .

The example in Fig. 2 presents a RWF-net modelling a workflow which processes admission applications for a college. There are two types of resources (assistants and professors) and two possible roles: secretary (S) and commission member (CM). A S role can be performed by an assistant and a CM role can be performed by a professor. The specification for the resource perspective is: $RT = \{\text{assistants, professors}\}$,

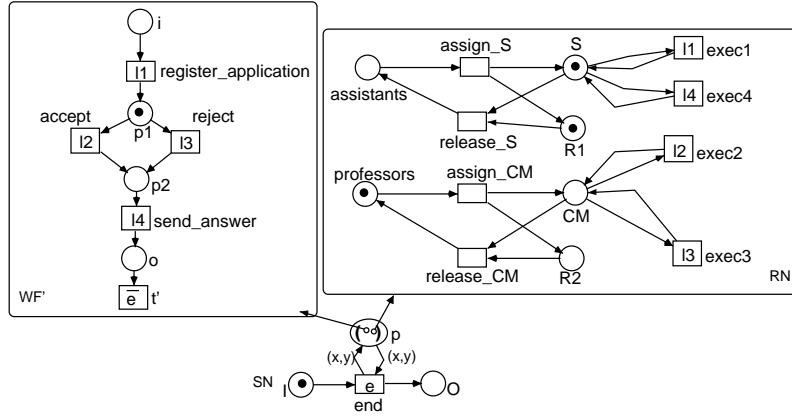


Fig. 2. A resource workflow net in its initial marking.

$RO = \{S, CM\}$, $res(S) = \{\text{assistants}\}$, $res(CM) = \{\text{professors}\}$. The function *Role* describes which roles must execute the tasks of the workflow and it is defined as follows: $Role(\text{register_application}) = S$, $Role(\text{accept}) = CM$, $Role(\text{reject}) = CM$, $Role(\text{send_answer}) = S$. The marking m_0 for *RN* is: $m_0 = 1'assistants + 1'professors$. In the initial marking of the RWF-net, $M_0(I) = 1$, $M_0(p) = ((WF', i.1), (RN, m_0))$. In M_0 , transition *register_application* from *WF'* cannot fire, although it is enabled in $(WF', i.1)$: it can only fire simultaneously with transition *exec1* in *RN*, which is not enabled. Assume the object-autonomous step (*assign_S*) fires first. The new marking of the RWF-net is $M_1 = (1, ((WF', i.1), (RN, m_1)), 0)$, where $m_1 = 1'S + 1'R_1 + 1'professors$. The step (*register_application, exec1*) is enabled in M_1 , producing a new marking (see Fig. 2) $M_2 = (1, ((WF', m'_1), (RN, m_2)), 0)$, where $m'_1 = 1'p_1$ and $m_2 = m_1$. The system remains blocked (no other task in the process is executed) until resources (assistants) are allocated for the *CM* role. Transition *assign_CM* can fire independently in *RN*. The resulting marking is $M_3 = (1, ((WF', m'_1), (RN, m_3)), 0)$, where $m_3 = 1'S + 1'CM + 1'R_2 + 1'R_1$. In M_3 , the step (*accept, exec2*) can fire and the resulting marking is $M_4 = (1, ((WF', m'_2), (RN, m_4)), 0)$ where $m'_2 = 1'p_2$ and $m_4 = m_3$. Next, the step (*send_answer, exec4*) can fire producing the marking $M_5 = (1, ((WF', m'_3), (RN, m_5)), 0)$ where $m_5 = m_4$ and $m'_3 = 1'o$. The vertical synchronization step $Y = (\text{end}; t')$ 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 produces the marking $(0, ((WF', 0), (RN, m_5)), 1)$ which corresponds to the correct processing of the case.

3 The k -Soundness of Resource Workflow Nets

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

A notion of k -soundness was defined for WF-nets, expressing the minimal conditions a correct workflow should satisfy ([3, 5]). We consider that an extended workflow net *WF'* is sound if the underlying WF-net is sound.

A workflow net $WF = (P, T, F)$ is k -sound iff: (1) for every marking m reachable from the initial marking $i.k$, there exists a firing sequence leading from m to the

final marking $o.k$ (termination condition): $(\forall m)((i[*]m) \implies (m[*]o))$; (2) All the transitions in WF are quasi-live: $(\forall t \in T)(\exists m, m')(i.k[*]m[t]m')$.

It was proven ([5]) that, if WF is k -sound: $(m \in [i.k]) \wedge m \geq o.k \implies (m = o.k)$.

The following result can be easily proven:

Lemma 1. *Let $RWFN$ be a RWF-net and $M \in [M_0^k]$ a reachable marking. Then $M = (k_1, ((WF', m), (RN, rm)), k_2)$, where m is a reachable marking in WF' and rm is a reachable marking in RN .*

Let $RWFN$ be a RWF-net. If the initial marking is M_0^k , the set of final markings for $RWFN$ is: $\mathcal{M}_f^k = \{(0, ((WF', 0), (RN, rm)), k) \mid rm \text{ is a reachable marking of } RN\}$.

We will consider that a RWF-net is k -sound if: (1) WF' is k -sound, (2) for any reachable marking of the RWF-net, $M \in [M_0^k]$, there is a firing sequence that leads to a final marking M_f (the termination property) and (3) all the steps in the RWF-net are quasi-live.

Definition 2. *A RWF-net $RWFN$ is k -sound if and only if:*

1. $(WF', i.k)$ is a k -sound extended workflow net.
2. For every marking M reachable from the initial marking M_0^k , there exists a firing sequence leading from M to a final marking M_f :
 $(\forall M)((M_0^k[*]M) \implies (M[*]M_f, M_f \in \mathcal{M}_f^k))$.
3. $RWFN$ is quasi-live: $\forall Y \in \mathcal{Y}$, there exists a marking $M \in [M_0]$ such that $M[Y]$.

First, we consider the workflow is k -sound if the WF-net describing the process is k -sound (abstracting from resources). A final marking of the RWF-net is reached if the vertical synchronization step fires k times. This implies that transition t' can fire k times in WF' , which happens if and only if the final marking of the WF-net can be reached. Thus, condition (2) states that the workflow is k -sound if the termination condition still holds in the WF-net, when the firing of tasks is restricted by the resource perspective.

Lemma 2. *Let $RWFN$ be a RWF-net such that WF' is k -sound. The markings in \mathcal{M}_f^k are the only markings reachable from M_0^k which contain k tokens in place O :
 $(\forall M)((M_0^k[*]M) \wedge M(O) = k) \implies (M \in \mathcal{M}_f^k)$.*

Proof. Let $M \in [M_0^k]$ with $M(O) = k$. Tokens can be added to place O only by the firing of the vertical synchronization step ($end; t'$). In order to produce k tokens in place O , this step has to fire k times. Since $M_0[*]M$, then $m \in [i.k]_{WF'}$ and $rm \in [rm_0]$ (Lemma 1). Thus, $\exists \sigma \in T'^*$ such that $i.k[\sigma]m$. t' has k occurrences in σ . Since $o \bullet = \{t'\}$, the order in which the transitions t' appears in σ is not important. Thus, we can write $\sigma = \sigma'(t')^k$, $\sigma' \in T'^*$. Hence, $i.k[\sigma']m'[(t')^k]m$. $m'(o) = k$. Since WF is k -sound, the only reachable marking with k tokens in o is $o.k$. If $o.k[(t')^k]m$, then $m = 0$. Also, because $M(I) + M(O) = k$ and $M(O) = k$, it results that $M(I) = 0$. Thus, $M \in \mathcal{M}_f^k$.

Lemma 3. *Let $RWFN = (Var, Lab, (WF', i.k), (RN, rm_0), SN, \Lambda, Role)$ be a RWF-net such that WF' is k -sound and (RN, rm_0) is live. Then, for every reachable marking $m \in [i.k]$ in WF' , there is a reachable marking $M \in [M_0^k]$ in $RWFN$ such that $M = (k_1, ((WF', m), (RN, rm)), k_2)$, where $rm \in [rm_0]$.*

Proof. Let $m \in [i.k]$ be a reachable marking in WF' . Then, there is a sequence of transitions in WF' , $\sigma \in T'^*$, such that $i.k[\sigma]m$. Let $|\sigma| = n$. We will prove, by induction on n , that there exists a sequence of steps in $RWFN$, $\sigma' \in \mathcal{Y}^*$, such that $M_0^k[\sigma']M$ and $M = (k_1, ((WF', m), (RN, rm)), k_2)$, where rm is a reachable marking in RN .

Base. If $n = 0$, then $m = i.k$. If we consider σ' the empty sequence of steps in $RWFN$, then $M_0^k[\sigma']M_0^k$, $M_0^k = (k, ((WF', i.k), (RN, rm_0)), 0)$.

Step. Assume the property holds for n and we prove it for $n + 1$: If $i.k[t_1 \dots t_n]m'[t_{n+1}]m$, for the sequence $i.k[t_1 \dots t_n]m'$, there is a sequence of steps in $RWFN$ such that $M_0^k[Y_1 \dots Y_s]M'$, where $M' = (k'_1, ((WF', m'), (RN, rm')), k'_2)$ and $rm' \in [rm_0]$.

(1) If t_{n+1} is an unlabelled transition in WF' : the result follows easily, considering then the object-autonomous step $Y = (; t_{n+1})$ in $RWFN$ from the marking M' .

(2) If $t_{n+1} \neq t'$ is a labelled transition in WF' , then there exists t'_{n+1} in RN such that $\lambda(t_{n+1}) = \lambda(t'_{n+1})$. If t'_{n+1} is not enabled in rm' in RN , because t'_{n+1} is live, there exists $rm'' \in [rm']$ such that: $rm'[t'_1 \dots t'_m]rm''[t'_{n+1}]rm$. We can assume there are no labelled transitions in this sequence (the labelled transitions do not have any effect on markings so they can be removed from the sequence). We can consider the following steps: $Y'_1 = (; t'_1), \dots, Y'_m = (; t'_m)$. Thus, $M'[Y'_1 \dots Y'_m]M''$, where $M'' = (k'_1, ((WF', m'), (RN, rm'')), k'_2)$. $Y = (t_{n+1}; t'_{n+1})$ is enabled in M'' : $M_0^k[Y_1 \dots Y_s]M'[Y'_1 \dots Y'_m]M''[Y]M$, where $M = (k'_1, ((WF', m), (RN, rm)), k'_2)$ and $rm \in [rm_0]$.

(3) If $t_{n+1} = t'$, t' is enabled in marking m' in WF' . We prove that the vertical synchronization step $Y = (end[b]; t')$ is enabled in M' with the binding $b(x) = (WF', m')$, $b(y) = (RN, rm')$. In order to show that transition end is enabled in M' , we will prove that $M'(I) > 0$. Assume $M'(I) = 0$. Then, it results that $M'(O) = k$. From Lemma 2, it results that $m' = 0$, which contradicts the fact that t' is enabled in WF' . Thus, $M_0^k[Y_1 \dots Y_s](k'_1, ((WF', m'), (RN, rm'')), k'_2)[(end[b]; t')]$ $(k'_1 - 1, ((WF', m), (RN, rm')), k'_2 + 1)$.

Let $(WF', i.k)$ be a k -sound extended WF-net together with its initial marking. It can be easily proven that all the transitions in WF' are quasi-live.

Lemma 4. *Let $RWFN$ be a RWF-net such that WF' is k -sound and (RN, rm_0) is live. All the steps in $RWFN$ are quasi-live: $\forall Y \in \mathcal{Y}, \exists M \in [M_0^k]$ such that $M[Y]$.*

Proof. WF' is k -sound, so all the transitions in WF' are quasi-live. (RN, m_0) is live hence, all the transitions in RN are quasi-live. Let Y be a step in $RWFN$. We consider the following cases:

(1) Y is an object-autonomous step: then $Y = (; t)$ and t is an unlabelled transition from WF' or RN . If t is an unlabelled transition from RN : since t is live in RN , there is $rm \in [rm_0]$ such that: $rm_0[t_1 \dots t_m]rm[t]$. We can consider that t_1, \dots, t_m are unlabelled in RN . The following object-autonomous steps can fire in $RWFN$ from the initial marking: $Y_1 = (; t_1), \dots, Y_m = (; t_m)$: $M_0^k[Y_1 \dots Y_m]M$, where $M = (k, ((WF', i.k), (RN, rm)), 0)$. The step $Y = (; t)$ is enabled in M , because $rm[t]$ in RN . If t is an unlabelled transition from WF' : similar to the previous case.

(2) Y is a horizontal synchronization step: similar to (1).

(3) If Y is the vertical synchronization step: $Y = (end; t')$. t' is quasi-live in WF' , so there is $m \in [i.k]$ such that $m[t']$. For $m \in [i.k]$, there exists $M \in [M_0^k]$, $M = (k_1, ((WF', m), (RN, rm)), k_2)$. Also, $M(I) \neq 0$ (otherwise, from Lemma 2, $M \in \mathcal{M}_f^k$ and $m = 0$). Now, if we consider the binding b with $b(x) = (WF', m)$ and $b(y) = (RN, rm)$, then end is enabled in marking M w.r.t. binding b and $M[end(b); t']$.

The following theorem shows that in a RWF-net in which the resource net RN is live, the k -soundness is equivalent with the k -soundness of its workflow net, which is a decidable property ([3, 4]):

Theorem 1. *Let $RWFN = (Var, Lab, (WF', i), (RN, rm_0), SN, \Lambda, Role)$ be a RWF-net such that (RN, rm_0) is a live resource net. Then, $RWFN$ is k -sound if and only if WF' is k -sound.*

Proof. (\implies) If $RWFN$ is k -sound, WF' is k -sound.

(\impliedby) If WF' is k -sound, the first condition in the definition of soundness for RWF-net takes place. Also, $RWFN$ is quasi-live: because (RN, rm_0) is live and WF' is k -sound, using Lemma 4, it results that $RWFN$ is quasi-live.

We will show that the second condition in the definition of RWF-nets also takes place: Assume $M \in [M_0^k]$, $M = (k_1, ((WF', m), (RN, rm)), k_2)$. There exists $\tau \in \mathcal{Y}$ such that $M_0[\tau]M$. k_2 is the number of occurrences of the vertical step $(end; t')$ in τ . $k_1 + k_2 = k$. We can write $\tau = \tau'(end; t')^{k_2}$ and $M_0[\tau']M'[(end; t')^{k_2}]M$. $M' = (k, ((WF', m'), (RN, rm)), 0)$ and $m = m' - o.k_2$. Using Lemma 1, it results that there exists a transition sequence σ in WF' such that $i.k[\sigma]m'$ (σ does not contain the transition t'). Because WF' is k -sound, there is a transition sequence σ' such that $m'[\sigma']o.k$. σ' does not contain t' and σ' can also fire from $m = m' - o.k_2$ (o does not have output transitions in σ'). $m[\sigma'](o.k - o.k_2)$. Let $\sigma' = t_1 \dots t_k$ and $t_i \neq t'$. We show there exists a sequence of steps such that $M' = (k, ((WF', m'), (RN, rm)), 0)[*](k, ((WF', o.k), (RN, rm)), 0)$. Let $Y = (t, t^*)$, with $\lambda(t) = \lambda(t^*)$. If t^* is not enabled in rm , since (RN, rm_0) is live, there exists a sequence of transitions $rt_1 \dots rt_k$ such that $rm[rt_1 \dots rt_k]rm'[t^*]rm_1$. The steps $Y_1 = (; rt_1), \dots, Y_k = (; rt_k)$ are enabled in M' and $M'[Y_1 \dots Y_k]M''[Y]$. Thus, for every transitions $t_i \in \sigma'$ from WF we can obtain a corresponding sequence of steps in $RWFN$. Finally, we can obtain a sequence of transitions π such that $M'[\pi](k, ((WF', o.k), (RN, rm^*)), 0)$. No vertical synchronization step fires in π and π does not remove tokens from o . Also, no transitions remove tokens from o except t' . Thus, π can fire from M and $M = (k_1, ((WF', m' - o.k_2), (RN, rm)), k_2)[\pi](k_1, ((WF', o.k - o.k_2), (RN, rm^*)), k_2)$. Now, the vertical synchronization step can fire $k - k_1$ times producing the final marking $(0, ((WF', 0), (RN, rm)), k)$.

4 Conclusions

This paper presented a special class of nested Petri nets used to model both the resource perspective and the process perspective for a workflow, allowing the handling of k cases at a time. The two perspectives are represented as two independent object-nets. 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 ([3, 4, 10, 14, 7, 13]), resources and roles are not represented in the same Petri net as the process. A notion of k -soundness was introduced in order to study the logical correctness of workflows: even if the workflow is 1-sound, when k cases are processed, it is possible that some cases could not be handled due to insufficient resources. We proved that k -soundness is decidable for a special class of RWF-nets.

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: *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, Springer-Verlag, Berlin, 1997.
3. 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.
4. K. van Hee, N. Sidorova, M. Voorhoeve: *Resource-Constrained Workflow Nets*. Lindemann, Burkhard, Czaja, Skowron, Schlingloff, Suraj (Eds.): Proc. of the International Workshop on Concurrency, Specification and Programming (CS & P) 2004, pp. 166-177, Informatik-Berichte der Humboldt-Universität zu Berlin, Nr. 170, September 2004.
5. K. Hee, N. Sidorova and M. Voorhoeve: *Generalised Soundness of Workflow Nets Is Decidable*. In Application and Theory of Petri Nets 2004, vol. 3099 of Lecture Notes in Computer Science, pages 8295. Springer-Verlag, Berlin, 2004 .
6. 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.
7. J. Li, Y. Fan, and M. Zhou. *Performance modeling and analysis of workflow*. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 34:229242, 2004
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. M. zur Mühlen. *Resource modeling in workflow applications*. Proceedings of the 1999 Workflow Management Conference (November 1999), pp. 137-153, 1999.
10. M. Netjes, W.M.P. van der Aalst, H.A. Reijers. *Analysis of resource-constrained processes with colored petri nets*. In K. Jensen, Proc. of the 6th Workshop and Tutorial on Practical Use of Coloured Petri nets and the CPN Tools (CPN'05), pp. 251-265, 2005
11. O. Prisecaru. *Resource workflow nets:a Petri Net Formalism for Workflow Modelling*. In Proc. of MSVVEIS 2007, pages 1121, 2007.
12. O. Prisecaru. *Resource workflow nets: an approach to workflow modelling and analysis*. Enterprise Information Systems, 2:101120, 2008.
13. H. A. Reijers. *Design and Control of Workflow Processes*. Business Process Management for the Service Industry, volume 2617 of LNCS. Springer-Verlag, Berlin, 2003.
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, Proc. 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.