Workflow Nets with Roles

Robin Bergenthum, Jörg Desel, Sebastian Mauser
Department of Software Engineering, FernUniversität in Hagen, Germany

Abstract: We formalize the usual static role concept for workflow nets, introduce dynamic roles and study the relation between soundness of workflow nets with roles and without roles.

1 Introduction

The focus of formal business process modeling and related analysis methods is mainly on the control flow perspective. However, there is also research [HSV06, JKJ10, BBS07, RAHE05, PA07, RAH09, Pri08] on the resource perspective of business processes, i.e. on the people, systems and machines actually doing the work. In this paper we formally model human resources, i.e. the actors enabling a business process.

In this context, a central question is which actors are allowed to work on a given task of a process. Similar to the concept of role-based access control (RBAC) in the domain of IT-security [SCFY96], business process modeling languages as well as industrial workflow systems usually consider roles as an intermediary concept between tasks and actors [TCG04, RAHE05, AH02]. The permission to execute a task is associated with one or more roles, and conversely also actors are assigned to roles, thereby acquiring the permissions of the roles. Roles are mostly determined by the organizational units (also called groups) and the positions within an enterprise, but also skills and responsibilities are regarded. Typical modeling constructs for assigning roles to tasks are swimlanes referring to roles as in BPMN and EPCs or an annotation of tasks with roles as common for workflow (Petri) nets [AH02, Aal98, Wes07].

In this paper, we first provide a formal semantics of the usual role concept for workflow nets by a translation into Colored Petri nets (CP-nets) [Jen97]. There already exist papers that show how to model resource allocation and the handling of resources available for process instances by CP-nets [PA07, RAH09] (there is also some work considering nets in nets [Pri08] and plain Petri nets [JKJ10, HSV06]). These papers focus on the distribution of work items, resource allocation patterns, resource management, the life cycle of work items etc. These aspects are important for workflow systems, but they go beyond the pure role concept of workflow nets.

Moreover, we do not restrict our considerations to static roles but also define a formal semantics for workflow nets with dynamic roles which have informally been introduced in [BDHM11] in the context of learning processes. Besides the intended applicability for learning and teaching processes, workflow nets with dynamic roles are useful for formally modeling and analyzing business processes with complex authorization constraints. In

particular, the principles of separation of duty (SoD) and binding of duty (BoD) [TCG04] can nicely be represented by dynamic roles. For instance, when writing a review an actor gets a new role which is later on required for presentation of the review (BoD) but prohibits writing a second review (SoD). In the literature, there are several interesting approaches, mostly based on RBAC concepts [SCFY96], which introduce specific additional model constructs and notations for explicitly representing authorization constraints in the context of workflows, e.g. [TCG04].

The definitions of workflow nets with static and dynamic roles in this paper enable formal methods. In particular, it is possible to define behavioral correctness criteria regarding these role concepts. The most popular correctness criterion for workflow nets is soundness [Aal98, Wes07]. We first extend this notion in a natural way to workflow nets with static and dynamic roles. Then, we introduce a second correctness criterion, called consistency, which requires that the role perspective does not influence the control flow of a model. We briefly show that, in the case of static roles, soundness and consistency have a very simple and intuitive meaning, namely a workflow net with static roles is consistent iff, for each non-dead task, there is at least one actor having one of the roles assigned to the task, and it is sound iff it is consistent and the underlying workflow net is sound. Concerning workflow nets with dynamic roles the notions of soundness and consistency become more difficult. We show that a consistent model is sound iff the underlying workflow net is sound. Counterexamples show that further relations between consistency and soundness do not hold. Lastly, we discuss a characterization of consistency for dynamic roles which regards the role perspective separately from the underlying workflow net.

The paper is organized as follows. In Section 2 we recapitulate the standard role concept for workflow nets and define a formal semantics for this concept using CP-nets. In Section 3 we formally introduce and define dynamic roles for workflow nets. In Section 4 we discuss soundness and consistency for workflow nets with static and with dynamic roles.

2 Standard Role Concept

The role concepts of most business process modeling languages are similar. They basically allow an assignment of users and tasks to roles. This is also the case for the standard role concept of workflow nets [AH02, Aal98]. This section presents a formal semantics of this concept by translating it into a CP-net-notation. We assume that the reader is familiar with workflow nets and CP-nets. We use the following notations. $\mathbb{N}$ denotes the non-negative integers. For a finite set A, 2^A denotes the powerset of A and $\mathbb{N}^A$ the set of multisets over A. For $m \in \mathbb{N}^A$ we write $m = \sum_{a \in A} m(a) \cdot a$. If $m(a) > 0$ we write $a \in m$.

Definition 1. *A workflow net (WF-net) is a tuple $N = (P, T, F, i, f)$, where*

- *P and T are finite sets of places and transitions fulfilling $P \cap T = \emptyset$,*
- *$F \subseteq (P \times T) \cup (T \times P)$ is a flow relation,*
- *$i, f \in P$ are places satisfying $(T \times \{i\}) \cap F = \emptyset$ and $(\{f\} \times T) \cap F = \emptyset$,*
- *for any node $n \in P \cup T$ there exists a path from i to n and a path from n to f.*

The behavioral semantics, i.e. the occurrence sequences, of a WF-net is given by considering the corresponding marked Petri net with the initial marking $1 \cdot i$.

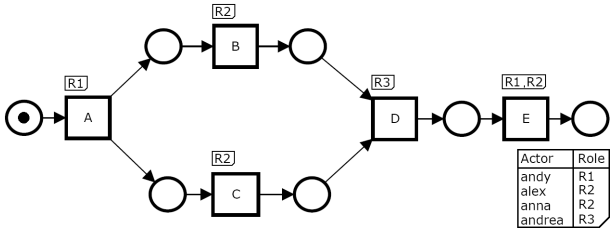

Figure 1: A WFR-net.

Definition 2. *A workflow net with roles (WFR-net) is a tuple* $NR = (N, R, A, l, r)$*, where*

- $N = (P, T, F, i, f)$ *is a WF-net,*
- R *is a finite set of roles,*
- A *is a finite set of actors,*
- $l : T \to 2^R$ *is a labeling function assigning a set of roles to each transition,*
- $r : A \to R$ *is a function assigning a role to each actor.*

This definition represents roles and actors in an intuitive and simple way. It is similar to most standard role concepts for business processes. Each actor having one of the roles associated to a transition is permitted to execute a respective task. Unlabeled transitions (having assigned the empty set) can occur automatically without requiring an actor.

Figure 1 shows an example workflow net with roles. There are four actors, one with role R1, two with role R2 and one with role R3. Task A has to be done by an actor with role R1. Then, B and C can be executed concurrently by R2-actors. Afterwards, D requires the role R3. Finally, E can be accomplished by an actor with role R1 or with role R2.

To provide a formal operational semantics for WFR-nets, we define a translation into CP-nets. We extend the underlying WF-net by a place which serves as a resource pool, containing all actors with their associated roles. The role annotations of transitions are considered by appropriate guards. In this way roles, actors and execution permissions are regarded. We here just provide the definition of a CP-net [Jen97]. For the operational semantics of CP-nets, see [Jen97].

Definition 3. *A colored Petri net (CP-net) is a tuple* $CPN = (C, P, T, F, V, c, v, g, e, m_0)$*, where*

- C *is a finite set of non-empty types (each type is a set called color set),*
- P *and* T *are finite sets of places and transitions fulfilling* $P \cap T = \emptyset$*,*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a flow relation defining a set of arcs,*
- V *is a finite set of variables,*
- $c : P \to C$ *is a coloring function assigning a type to every place,*
- $v : V \to C$ *is a coloring function assigning a type to each variable,*
- g *assigns a boolean expression using variables from* V *to every transition,*
- e *assigns an expression of type* $\mathbb{N}^{c(p)}$ *using appropriate variables from* V *to every arc,*
- $m_0 : P \to \bigcup_{p \in P} \mathbb{N}^{c(p)}$ *assigns an initial marking* $m_0(p) \in \mathbb{N}^{c(p)}$ *to every place* p*.*

Black tokens are represented by the color set $UNIT = \{()\}$.

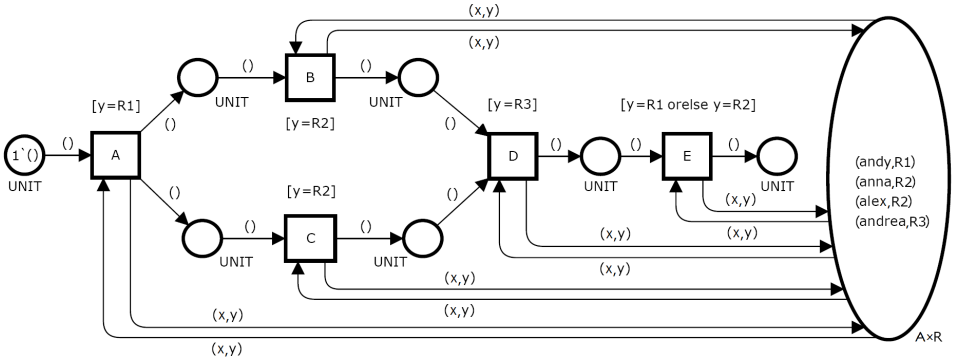

Figure 2: CP-net corresponding to a WFR-net.

Definition 4. *Given a WFR-net* $NR = (P, T, F, i, f, R, A, l, r)$*, we define the corresponding CP-net* $CPN_{NR} = (C, P', T, F', V, c, v, g, e, m_0)$ *by*

- $C = \{UNIT, A, R, A \times R\}$,
- $P' = P \cup \{p_{res}\}$,
- $F' = F \cup (\{t \in T \mid l(t) \neq \emptyset\} \times \{p_{res}\}) \cup (\{p_{res}\} \times \{t \in T \mid l(t) \neq \emptyset\})$,
- $V = \{x, y\}$,
- $c(p) = UNIT$ *for* $p \in P$, $c(p_{res}) = A \times R$,
- $v(x) = A$, $v(y) = R$,
- $g(t) = [y = g_1\ orelse\ \ldots y = g_n]$ *for* $t \in T$ *satisfying* $l(t) = \{g_1, \ldots, g_n\}$,
- $e(z) = 1 \cdot ()$ *for* $z \in F$, $e(z) = (x, y)$ *for* $z \in (\{t \in T \mid l(t) \neq \emptyset\} \times \{p_{res}\}) \cup (\{p_{res}\} \times \{t \in T \mid l(t) \neq \emptyset\})$,
- $m_0(i) = 1 \cdot ()$, $m_0(p) = \emptyset$ *for* $p \in P \setminus \{i\}$, $m_0(p_{res}) = \sum_{a \in A}(a, r(a))$.

In order to fire a transition of the introduced CP-net, the variables x and y have to be bound. In this way an actor x having the role y is allocated to the task. The transition guard ensures that the allocated actor is permitted to execute the task, i.e. it is checked that his role y is assigned to the task in the WFR-net. When an actor executes a task, he is removed from the place p_{res}. As soon as the task is accomplished, the actor is released by giving it back to the place p_{res}. In this way the place p_{res} guarantees that at any time an actor can only be allocated to one task. An exception are tasks of the original WFR-net having an empty set of roles. These automatic tasks require no actor from the resource pool p_{res}. Remark that the concept of releasing an actor allocated to a task as soon as the task is completed is an important difference to the approaches in [HSV06, JKJ10, BBS07], where also WF-nets regarding resources are formally discussed. Figure 2 depicts the CP-net corresponding to our example WFR-net from Figure 1.

We have restricted ourselves to a basic role model in this section. This basic model can be extended in different directions:

- The basic role model does not regard process instances. However, different process instances can easily be distinguished by using a copy of the WFR-net for each instance and connecting each copy with the place representing the resource pool. Then, this place does not only ensure that an actor cannot execute two tasks of one process

instance at once but also that an actor cannot execute two tasks of different process instances at once. One resource place can also be used for different process models. Then, the same actors are shared among the process instances of several processes.

- In the basic role model an actor can only be assigned to one role, and we do not consider a hierarchy among roles. While these restrictions reduce the modeling comfort, they do not restrict the modeling capabilities, since these aspects can equivalently be represented by the concept of alternative role annotations. For explicitly modeling multiple roles of actors, we can associate sets of roles instead of single roles to each actor. Then, the transition guards check whether a specific role is contained in the set of roles of an actor. A hierarchy among roles can be expressed by a consistency condition on the function assigning sets of roles to actors.
- The basic model does not regard collaborative tasks which require a joint execution by several actors with certain roles, e.g. two authors that write a paper together. We can extend the role model by collaborative tasks as follows (see [BDHM11] for details). Instead of one actor, a collaborative task consumes a certain number of actors from the resource pool. To ensure that each of the actors has an appropriate role, we can use the "andalso"-operation in the guards.

3 Dynamic Roles

The idea of dynamic roles is to change role assignments of actors depending on tasks executed by an actor, i.e. the role of an actor depends on his task history. We will extend the intuitive modeling language of WFR-nets by this concept. If an actor having a certain role changes his role when executing a task, we represent this by extending the label of the transition. Formally, we simply consider pairs of roles consisting of the old and the new role, i.e. the labeling function now assigns a set of pairs of roles instead of a set of single roles to the transitions. If a role assignment does not change when executing a task, the old and the new role coincide.

Definition 5. *A workflow net with dynamic roles (WFDR-net) is a tuple NR*$= (N, R, A, l, r)$*, where*

- $N = (P, T, F, i, f)$ *is a WF-net,*
- R *is a finite set of roles,*
- A *is a finite set of actors,*
- $l : T \to 2^{R \times R}$ *is a labeling function assigning a set of pairs of roles to each transition,*
- $r : A \to R$ *is a function assigning a standard role to each actor.*

Figure 3 shows an example workflow net with dynamic roles. There are two actors, both initially having the role R1. Both are allowed to execute task A. Also the concurrent tasks B and C require the role R1, i.e. B as well as C can be accomplished by any of the two actors. However, one and the same actor cannot execute both tasks (separation of duty), since each of them causes a role change. The actor executing B gets the role R2, the one executing C gets the role R3. Therefore, the two actors have to share the two tasks among each other. After B and C, task D can be done by either an R2-actor or an R3-actor, i.e. by any of the two actors. Finally, E has to be executed by an actor with role R2. Thus, this task requires the actor that executed B before (binding of duty).

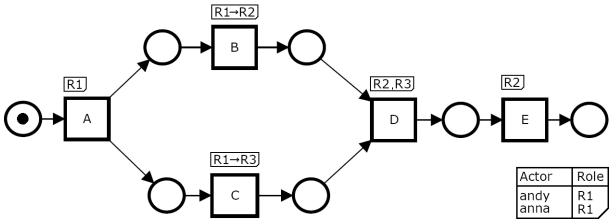

Figure 3: A WFDR-net.

Analogously to the last section, we define the semantics of a WFDR-net by a translation into a CP-net. In addition to Definition 4, the dynamic roles of actors are regarded. When firing a transition three variables have to be bound. Besides the actor x, the variable y_1 represents the role required to execute the task and y_2 represents a role change.

Definition 6. *Given a WFDR-net NR*$= (P, T, F, i, f, R, A, l, r)$*, we define the corresponding CP-net* $CPN_{NR} = (C, P', T, F', V, c, v, g, e, m_0)$ *by*

- $C = \{UNIT, A, R, A \times R\}$,
- $P' = P \cup \{p_{res}\}$,
- $F' = F \cup (\{t \in T \mid l(t) \neq \emptyset\} \times \{p_{res}\}) \cup (\{p_{res}\} \times \{t \in T \mid l(t) \neq \emptyset\})$,
- $V = \{x, y_1, y_2\}$,
- $c(p) = UNIT$ *for* $p \in P$, $c(p_{res}) = A \times R$
- $v(x) = A$, $v(y_1) = R$, $v(y_2) = R$,
- $g(t) = [(y_1 = g_{1,1}\, andalso\, y_2 = g_{1,2})\, orelse\, \ldots (y_1 = g_{n,1}\, andalso\, y_2 = g_{n,2})]$ *for* $t \in T$ *satisfying* $l(t) = \{(g_{1,1}, g_{1,2}), \ldots (g_{n,1}, g_{n,2})\}$,
- $e(z) = 1 \cdot ()$ *for* $z \in F$,$e(z) = (x, y_1)$ *for* $z \in \{p_{res}\} \times \{t \in T \mid l(t) \neq \emptyset\}$, $e(z) = (x, y_2)$ *for* $z \in \{t \in T \mid l(t) \neq \emptyset\} \times \{p_{res}\}$,
- $m_0(i) = 1 \cdot ()$, $m_0(p) = \emptyset$ *for* $p \in P \setminus \{i\}$, $m_0(p_{res}) = \sum_{a \in A}(a, r(a))$.

Figure 4 shows the CP-net corresponding to the WFDR-net from Figure 3.

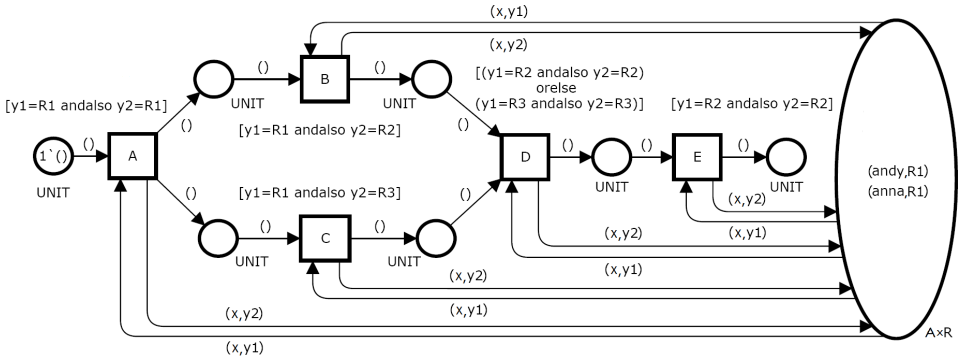

Figure 4: CP-net corresponding to a WFDR-net.

4 Soundness

In this section we discuss soundness of WFR-nets and WFDR-nets. A net may exhibit errors such as deadlocks, livelocks or garbage being left in the process after termination. Soundness is a basic behavioral property that each proper procedure should satisfy. For WF-nets the property states that "for any case, the procedure will terminate eventually, and at the moment the procedure terminates there is a token in place f and all the other places are empty" [Aal98]. Moreover, there should be no dead tasks.

Definition 7. *A WF-net $N = (P, T, F, i, f)$ is sound if*

(S1) *From each marking reachable from $1 \cdot i$, the marking $1 \cdot f$ is reachable.*
(S2) *There are no dead tasks w.r.t the initial marking $1 \cdot i$.*

As an example, the WF-net underlying the nets shown in Figure 1 and 3 is sound. Remark that the soundness definition originally included a third property stating that for each marking m reachable from $1 \cdot i$ with $m(f) \geq 1$, there holds $m = 1 \cdot f$. However, it was later shown that this property follows from (S1).

We now define a notion of soundness of WFR- and WFDR-nets which integrates the resource perspective of business processes. For this purpose we consider the corresponding CP-nets and formulate two requirements generalizing the properties (S1) and (S2).

Definition 8. *Let NR$= (N, R, A, l, r)$ be a WFR- resp. a WFDR-net and $CPN_{NR} = (C, P', T, F', V, c, v, g, e, m_0)$ the corresponding CP-net. A marking m_f of CPN_{NR} is called final marking if $m_f(f) = 1 \cdot ()$ and $m_f(p) = \emptyset$ for $p \in P \setminus \{f\}$*

Definition 9. *Let NR$= (N, R, A, l, r)$ be a WFR- resp. a WFDR-net and $CPN_{NR} = (C, P', T, F', V, c, v, g, e, m_0)$ the corresponding CP-net. The net NR is sound if*

(S1') *From each reachable marking of CPN_{NR}, a final marking is reachable.*
(S2') *CPN_{NR} has no dead tasks.*

It can easily be verified that the nets in Figure 2 and 4 fulfill the properties (S1') and (S2'). That means, the WFR-net in Figure 1 and the WFDR-net in Figure 3 are sound.

In WFR-nets and WFDR-nets the resource perspective is defined on top of the control flow perspective. The latter is given by a WF-net and the former by role annotations and actors. This concept clearly separates the resource view from the control flow view. Therefore, the actors and roles of a WFR-net resp. a WFDR-net should have no influence on the control flow of the model. They should only describe the resource allocation allowed within the business process. Otherwise, the two views are not consistent. Consequently, besides soundness we in the following introduce a second correctness property which is concerned with consistency of the resource perspective to the control flow perspective.

By requiring actors with certain roles for firing transitions, the control flow can only be restricted. That means, if a transition can occur in a certain marking of a WFR-net resp. a WFDR-net this transition is also enabled in the corresponding marking of the underlying WF-net where the corresponding marking is given by just neglecting the place p_{res}. Thereby, a transition occurrence of a CP-net, called binding element, is given by

a pair consisting of the fired transition and the firing mode (i.e. the binding of the variables) of the transition [Jen97]. For instance, in the net of Figure 4 the binding element $(A, < x = andy, y_1 = R1, y_2 = R2 >)$ is enabled.

Definition 10. *Let NR*$= (N, R, A, l, r)$ *be a WFR-net resp. a WFDR-net,* $CPN_{NR} = (C, P', T, F', V, c, v, g, e, m_0)$ *the corresponding CP-net and* m *a marking of* CPN_{NR}*. Then the marking* m^u *of* N *given by* $m^u(p) = |m(p)|$ *for* $p \in P$ *is called corresponding marking of* m*.*

Lemma 1. *Let* m *be a marking of* CPN_{NR}*. If the binding element* (t, b) *is enabled in* m *leading to the follower marking* m'*, then* t *is enabled in the marking* m^u *of* N *leading to the follower marking* m'^u*.*

Proof. By construction, compared to N, the CP-net CPN_{NR} just has the additional place p_{res} (together with arcs connecting the place with transitions). A well-known property of Petri nets is that adding a place to a Petri net can only restrict the enabledness of transitions and does not influence the markings w.r.t. other places. □

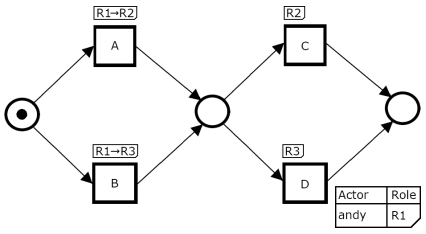

Figure 5: WFDR-net which is sound but not consistent.

It is possible that an enabled transition of N is prohibited in *NR*, since appropriate actors are missing, i.e. the place p_{res} prohibits the occurrence. If behavior specified by the control flow model cannot occur due to missing actors, this indicates inconsistency between the control flow and the resource perspective, i.e. the resources are not appropriate for the given control flow. Therefore, we formulate the following correctness criterion which ensures the reverse implication to Lemma 1.

Definition 11. *Let NR*$= (N, R, A, l, r)$ *be a WFR- resp. a WFDR-net and* $CPN_{NR} = (C, P', T, F', V, c, v, g, e, m_0)$ *be the corresponding CP-net. The net NR is called consistent if for each reachable marking* m *of* CPN_{NR} *and each transition* t *which is enabled in the marking* m^u *of* N*, there is a binding* b *such that* (t, b) *is enabled in* m*.*

Using Lemma 1 we can summarize the following relationship between the behavior of a WFR-net resp. a WFDR-net and the underlying WF-net in terms of occurrence sequences.

Lemma 2. *If an occurrence sequence* $(t_1, b_1) \dots (t_n, b_n)$ *is enabled in the initial marking* m_0 *of* CPN_{NR} *leading to the follower marking* m*, then* $t_1 \dots t_n$ *is enabled in the initial marking* $1 \cdot i$ *of* N *leading to the follower marking* m^u*. In the case NR is consistent we also have: If* $t_1 \dots t_n$ *is enabled in the initial marking* $1 \cdot i$ *of* N*, then there are bindings* $b_1, \dots, b_n$ *such that* $(t_1, b_1) \dots (t_n, b_n)$ *is enabled in the initial marking* m_0 *of* CPN_{NR}*.*

Proof. The statements follow from $m_0^u = 1 \cdot i$ by inductively applying Lemma 1 and Definition 11. □

The WFR-net of Figure 1 and the WFDR-net of Figure 3 are both consistent. Figure 5 shows a WFDR-net which is not consistent. However, this net and also its underlying WF-net are sound. The net contains two subsequent alternatives, first between task A and B, then between task C and D. The role annotations ensure that whenever the actor initially executes A, then he next has to execute C, although D is also enabled in the underlying WF-net. Similarly, when starting with B, the actor then has to execute D. Therefore, the role perspective of this net forbids behavior which is allowed by the control flow model, namely the occurrence sequences AD and BC.

4.1 WFR-nets

An important observation is that for WFR-nets the correctness property of consistency is already included in the soundness property.

Lemma 3. *A sound WFR-net NR*$= (N, R, A, l, r)$ *is consistent.*

Proof. If *NR* is not consistent, there is a reachable marking m of CPN_{NR} and a transition t such that t is enabled in the marking m^u of N, but for each b the binding element (t, b) is not enabled in m. It follows that the place p_{res} prohibits the firing of t. Since the marking of p_{res} never changes, the transition t is dead in CPN_{NR}, i.e. *NR* does not fulfill (S2). Consequently, *NR* is not sound. □

Since soundness of WF-nets is well investigated, we discuss the relation between soundness of a WFR-net and soundness of the underlying WF-net. It can first be shown that soundness of the underlying WF-net is a necessary condition for soundness of a WFR-net.

Lemma 4. *If NR*$= (N, R, A, l, r)$ *is a sound WFR-net, then also* N *is sound.*

Proof. If N is not sound, one of the conditions (S1) or (S2) is not satisfied. We show for each case that *NR* is not sound.

If N does not fulfill (S1), then there is a reachable marking m from which $1 \cdot f$ is not reachable. Either a marking m' with ${m'}^u = m$ is reachable in CPN_{NR} or this is not the case. In the first situation, by Lemma 1, a final marking is not reachable from m', since a final marking m_f fulfills $m_f^u = 1 \cdot f$. That means *NR* does not fulfill (S1'). In the second case, by Lemma 2, *NR* is not consistent and therefore Lemma 3 shows that *NR* is not sound.

If N does not fulfill (S2), then there is a dead task. By Lemma 2 this task is also dead in *NR*, i.e. *NR* does not fulfill (S2'). □

In general, consistency does not imply soundness, since it formulates no requirements on the control flow of the underlying WF-net. Soundness of the underlying net also does not imply soundness of a WFR-net because the WFR-net can contain dead tasks due to missing actors. However, we now show that both properties together, consistency and soundness of the underlying WF-net, ensure soundness of a WFR-net.

Lemma 5. *Let NR*$= (N, R, A, l, r)$ *be a WFR-net. If NR is consistent and* N *is sound, then also NR is sound.*

Proof. If *NR* is not sound, one of the conditions (S1') or (S2') is not satisfied. We show for each case that either *NR* is not consistent or N is not sound.

If *NR* does not fulfill (S1'), then there is a reachable marking m of CPN_{NR} from which no final marking is reachable. By Lemma 2, the marking m^u is reachable in N. In the case *NR* is consistent, with Lemma 1 it follows that the marking $1 \cdot f$ is not reachable from m^u, since a marking m_f of CPN_{NR} fulfilling $m_f^u = 1 \cdot f$ is a final marking. That means, N does not fulfill (S1).

If *NR* does not fulfill (S2'), then there is a dead task. In the case *NR* is consistent, by Lemma 2, this task is also dead in N, i.e. N does not fulfill (S2). □

The previous lemmas imply the following characterization of soundness for WFR-nets.

Theorem 1. *A WFR-net NR*$= (N, R, A, l, r)$ *is sound iff NR is consistent and* N *is sound.*

This characterization shows how to design sound WFR-nets. First, a sound WF-net is constructed. Then, roles and actors are added in a way which does not influence the control flow given by the WF-net.

So far the structural interpretation of the behavioral property consistency is not clear, and thus we do not know how to ensure the property when designing the role perspective of a WFR-net. Therefore, we provide a simple characterization of consistency:

Lemma 6. *A WFR-net NR*$= (N, R, A, l, r)$ *is consistent iff, for each transition* t *with* $l(t) \neq \emptyset$ *which is not dead w.r.t.* N*, there is an actor* $a \in A$ *such that* $r(a) \in l(t)$.

Proof. If *NR* is not consistent, the proof of Lemma 3 shows that there is a transition t which is not dead w.r.t. N but cannot fire w.r.t. CPN_{NR} due to the place p_{res} (which has a constant marking). It follows: $l(t) \neq \emptyset$ and there is no $a \in A$ such that $r(a) \in l(t)$.

If there exists a transition t with $l(t) \neq \emptyset$ which is not dead w.r.t. N such that there is no $a \in A$ with $r(a) \in l(t)$, then there is a reachable marking m of N which enables t but for any reachable marking m' of CPN_{NR} and any b the binding element (t, b) is not enabled in m' because the constant marking of p_{res} prohibits t. Therefore, if a marking m' with $m'^u = m$ is reachable in CPN_{NR}, then *NR* is not consistent. Otherwise, by Lemma 2, *NR* is not consistent. □

From this characterization we can immediately deduce two very simple sufficient conditions for consistency of a WFR-net NR$= (N, R, A, l, r)$ which are purely structural. In particular, they are completely independent from the underlying WF-net N. In the first condition we only remove the restriction to dead tasks from the previous characterization. The second condition is a further simplification abstracting from tasks. It just requires that for each role there is at least one actor having the role.

(C1) If, for each transition t with $l(t) \neq \emptyset$, there is an actor $a \in A$ such that $r(a) \in l(t)$, then *NR* is consistent.

(C2) If, for each role $x \in R$, there is an actor $a \in A$ with $r(a) = x$, then *NR* is consistent.

The WFR-net from Figure 1 fulfills (C2), since there is an actor for all three roles.

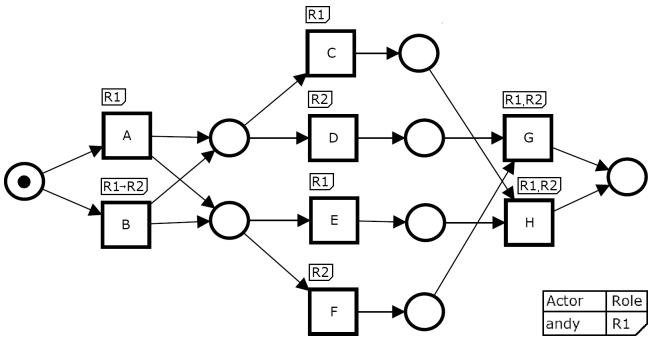

Figure 6: Sound WFDR-net.

4.2 WFDR-nets

For WFDR-nets, soundness and consistency become more difficult notions. In contrast to Lemmas 3 and 4 for WFR-nets, soundness of a WFDR-net does neither imply consistency of the WFDR-net nor soundness of the underlying WF-net. The WFDR-net of Figure 6 is sound, although it is not consistent and the underlying WF-net is not sound. The WF-net can run into a deadlock when firing tasks C and F or D and E. However, when firing C and E or D and F the net completes properly. Inconsistent role annotations of the WFDR-net prohibit the deadlocks of the underlying WF-net.

We have already discussed that soundness alone is not enough for a WFDR-net. For correctness, we are interested in sound and consistent WFDR-nets. For such nets it is possible to show that the underlying WF-net is also sound.

Lemma 7. *If NR*$= (N, R, A, l, r)$ *is a sound and consistent WFDR-net, then* N *is sound.*

Proof. If N is not sound, one of the conditions (S1) or (S2) is not satisfied. We show for each case that *NR* is not sound or not consistent.

If N does not fulfill (S1), then there is a reachable marking m from which $1 \cdot f$ is not reachable. In the case *NR* is consistent, by Lemma 2, a marking m' with $m'^u = m$ is reachable in CPN_{NR}. By using Lemma 1, a final marking is not reachable from m', since a final marking m_f fulfills $m_f^u = 1 \cdot f$. That means *NR* does not fulfill (S1').

If N does not fulfill (S2), then there is a dead task. By Lemma 2 this task is also dead in CPN_{NR}, i.e. *NR* does not fulfill (S2'). □

Moreover, analogously to Lemma 5 for WFR-nets, consistency of a WFDR-net together with soundness of the underlying WF-net implies soundness of the WFDR-net.

Lemma 8. *Let NR*$= (N, R, A, l, r)$ *be a WFDR-net. If NR is consistent and* N *is sound, then also NR is sound.*

Proof. The proof is analogous to Lemma 5. □

The two previous lemmas together yield the following equivalence.

Theorem 2. *A consistent WFDR-net NR*$= (N, R, A, l, r)$ *is sound iff* N *is sound.*

Altogether, we have shown how to design correct WFDR-nets. First, a sound WF-net is constructed which is then consistently extended by roles and actors. With this approach soundness of the resulting WFDR-net is guaranteed. Still, there are also sound WFDR-nets which are not consistent. For such a net it is possible that the underlying WF-net is not sound (Figure 6), but it is also possible that the WF-net is sound (Figure 5). For the sake of completeness note that in the case of a non-sound and non-consistent WFDR-net, the underlying WF-net can be both sound (Figure 5 with empty set of actors) or not sound (Figure 6 with empty set of actors).

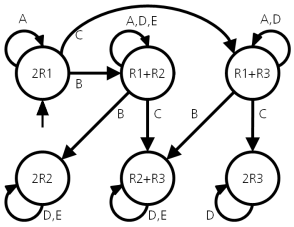

Figure 7: A role diagram.

We now investigate the property of consistency for WFDR-nets in detail. The aim is to find a characterization of consistency which explicitly regards the role perspective separately from the underlying WF-net. Then, as it is natural, consistency for the role perspective can be checked on top of the control flow perspective. For this purpose, we introduce the notion of role diagram which describes the dynamic role behavior of all the actors of a WFDR-net as given by the role annotations (neglecting the underlying WF-net). Figure 7 illustrates the role diagram of the WFDR-net from Figure 3. The role diagram of a WFDR-net is a non-deterministic finite automaton which models the overall dynamic resource perspective of a WFDR-net. States represent different role combinations of actors, and each transition represents a task that can be executed by a certain role combination as well as the role change triggered by this task execution.

Definition 12. *A non-deterministic finite automaton is a tuple $M = (Q, T, \delta, q_0)$, where*

- *Q is a finite set of states,*
- *T is a finite set of input symbols,*
- *$\delta \subseteq Q \times T \times Q$ is a transition relation and*
- *$q_0 \in Q$ is an initial state.*

Definition 13. *Let NR= $(P, T, F, i, f, R, A, l, r)$ be a WFDR-net. The role diagram $R_{NR} = (Q, T, \delta, q_0)$ of NR is defined inductively ($Q \subseteq \mathbb{N}^R$):*

- *$q_0 = \sum_{a \in A} r(a)$*
- *If $q \in Q$, $q(x_1) > 0$ and $(x_1, x_2) \in l(t)$ for $t \in T$ then $q' = \sum_{x \in R \setminus \{x_1, x_2\}} q(x) \cdot x + (q(x_1) - 1) \cdot x_1 + (q(x_2) + 1) \cdot x_2 \in Q$ and $(q, t, q') \in \delta$.*

A WFDR-net is consistent if the resource perspective, i.e. the place p_{res} does not prohibit any behavior of the underlying WF-net. That means, for each reachable state, if the WF-net allows the occurrence of a task, then there are actors capable of executing the task. In particular, the enabledness of a task has to be independent from the assignments of actors to previous tasks. Based on the concept of role diagram we formulate the following characterization for consistency of WFDR-nets.

Lemma 9. *Let NR= (N, R, A, l, r) be a WFDR-net and $R_{NR} = (Q, T, \delta, q_0)$ the role diagram of NR. Then, NR is consistent iff it fulfills the following property: For each occurrence sequence $t_1 \dots t_n$, $n \geq 1$, of N and each $(q_0, t_1, q_1) \dots (q_{n-2}, t_{n-1}, q_{n-1}) \in \delta$ there exists $q_n \in Q$ such that $(q_{n-1}, t_n, q_n) \in \delta$.*

Proof. If *NR* is not consistent, there is a marking m of CPN_{NR} reachable by an occurrence sequence $(t_1, b_1) \dots (t_{n-1}, b_{n-1})$ and a transition t_n which is enabled in the marking m^u of N, such that (t_n, b_n) is not enabled in m for each binding b_n. By Lemma 2, $t_1 \dots t_n$ is an occurrence sequence of N. Moreover, by construction of R_{NR} it holds $(q_0, t_1, q_1) \dots (q_{n-2}, t_{n-1}, q_{n-1}) \in \delta$ such that $q_{n-1} = \sum_{(a,x) \in A \times R} m(p_{res})(a, x) \cdot x$. Since t_n is not enabled in m w.r.t. the place p_{res}, it follows that there is no $(a, x_1) \in m(p_{res})$ such that $(x_1, x_2) \in l(t_n)$. Thus, there is no $x_1 \in q_{n-1}$ such that $(x_1, x_2) \in l(t_n)$. It follows that there does not exist a state $q_n \in Q$ with $(q_{n-1}, t_n, q_n) \in \delta$.

If there is an occurrence sequence $t_1 \dots t_n$ of N and $(q_0, t_1, q_1) \dots (q_{n-2}, t_{n-1}, q_{n-1}) \in \delta$ such that there is no $q_n \in Q$ with $(q_{n-1}, t_n, q_n) \in \delta$, then by construction of R_{NR} and Lemma 2 there are bindings $b_1 \dots b_{n-1}$ such that $(t_1, b_1) \dots (t_{n-1}, b_{n-1})$ is an occurrence sequence of CPN_{NR} which leads to a marking m with the following properties: $q_{n-1} = \sum_{(a,x) \in A \times R} m(p_{res})(a, x) \cdot x$ and m^u is the follower marking of the occurrence sequence $t_1 \dots t_{n-1}$ of N. By assumption there is no $x_1 \in q_{n-1}$ such that $(x_1, x_2) \in l(t_n)$ and thus there is no $(a, x_1) \in m(p_{res})$ such that $(x_1, x_2) \in l(t_n)$. Consequently, there is no b_n such that (t_n, b_n) is enabled in m, although t_n is enabled in m^u, i.e. *NR* is not consistent. □

With Lemma 9, consistency of the role perspective can nicely be checked on top of a given WF-net by comparing the marking graph of the WF-net and the role diagram.

The maximal occurrence sequences of the WF-net underlying the WFDR-net from Figure 3 are ABCDE and ACBDE. Thus, to verify consistency of the WFDR-net, for each prefix of these sequences we have to check that the property formulated in the previous lemma is satisfied by the role diagram of Figure 7. For instance, given the sequence ABCDE, starting in the initial state of the role diagram there is only one sequence of state transitions corresponding to ABCD. For the follower state R2+R3, it has to be checked that there is a state transition given by the task E.

From Lemma 9 we can also deduce reasonable sufficient conditions for consistency of a WFDR-net *NR*$= (N, R, A, l, r)$ with role diagram $R_{NR} = (Q, T, \delta, q_0)$ which are more simple to check. First, a simplification can be achieved in the case of deterministic role annotations: If the role diagram is deterministic, then it is enough to check whether each occurrence sequence of N is included in the role diagram. Second, we can consider the situation that there are always enough actors to perform each task of the net. For this purpose, we regard the set of all states $Q' \subseteq Q$ of the role diagram reachable by an occurrence sequence of N. That means, Q' represents all reachable role combinations in the resource place p_{res}. We formulate the following condition which is analogous to (C1) for WFR-nets: If for each task, each reachable role combination contains a role which is allowed to execute the task, then *NR* is consistent. Moreover, we can simplify this condition analogously as in the case of (C2) for WFR-nets: If each reachable role combination contains all roles of *NR*, then *NR* is consistent. Thereby, the set of all reachable role combinations Q' can be computed by projecting the states of the product automaton of R_{NR} and the marking graph of N onto the R_{NR}-component.

(D) If R_{NR} fulfills the property $(q, t, q'), (q, t, q'') \in \delta \implies q' = q''$, then *NR* is consistent iff R_{NR} accepts each occurrence sequence of N (when considering all states Q as final states).

(C1') If for each $t \in T$ with $l(t) \neq \emptyset$ and each $q \in Q'$, there exists $(x_1, x_2) \in l(t)$ such that $q(x_1) > 0$, then *NR* is consistent.

(C2') If for each $q \in Q'$ and each $x \in R$ there holds $q(x) > 0$, then *NR* is consistent.

Since the role diagram of Figure 7 is deterministic, we can verify consistency of the WFDR-net in Figure 3 by checking condition (D).

5 Conclusion

We have shown how to formally extend Petri net process models by static and dynamic role concepts. Then, we have discussed correctness of respective models. An important topic for future research is a detailed discussion of the extensions of the modeling languages mentioned in Section 2. Moreover, we plan to develop further analysis methods for the modeling languages, e.g. mining of respective process models from event logs of information systems.

References

[Aal98] W. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[AH02] W. van der Aalst and K. van Hee. *Workflow Management: Models, Methods, and Systems.* MIT Press, 2002.

[BBS07] K. Barkaoui, R. Benayed, and Z. Sba. Workflow Soundness Verification Based on Structure Theory of Petri Nets. *IJCIS Journal*, 5:51–62, 2007.

[BDHM11] R. Bergenthum, J. Desel, A. Harrer, and S. Mauser. Modeling and Mining of Learnflows. In *to appear in ToPNoC*. Springer, 2011.

[HSV06] K. van Hee, N. Sidorova, and M. Voorhoeve. Resource-Constrained Workflow Nets. *Fundam. Inf.*, 71:243–257, 2006.

[Jen97] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* Springer, 1992, 1994, 1997.

[JKJ10] G. Juhás, I. Kazlov, and A. Juhásová. Instance Deadlock: A Mystery behind Frozen Programs. In *Petri Nets 2010, LNCS 6128*, pages 1–17. Springer, 2010.

[PA07] M. Pesic and W. van der Aalst. Modeling Work Distribution Mechanisms using Colored Petri Nets. *International Journal on Software Tools for Technology Transfer*, 9(3-4):327–352, 2007.

[Pri08] O. Prisecaru. Resource workflow nets: an approach to workflow modelling and analysis. *Enterp. Inf. Syst.*, 2(2):101–120, 2008.

[RAH09] N. Russell, W. van der Aalst, and A. ter Hofstede. Designing a Workflow System Using Coloured Petri Nets. pages 1–24. Springer, 2009.

[RAHE05] N. Russell, W. van der Aalst, A. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In *CAiSE 2005, LNCS 3520*, pages 216–232. Springer, 2005.

[SCFY96] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[TCG04] K. Tan, J. Crampton, and C. Gunter. The Consistency of Task-Based Authorization Constraints in Workflow Systems. In *CSFW-17*, pages 155–169. IEEE, 2004.

[Wes07] M. Weske. *Business Process Management – Concepts, Languages and Architectures.* Springer, 2007.