

Modelling and Verification of Interorganizational Workflows with Security Constraints: A Petri Nets-Based Approach

Oana Otilia Captarencu

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

Abstract. Interorganizational workflows represent workflows which involve several business processes, belonging to different organizations, which need to coordinate their actions in order to achieve a common goal. This paper proposes a Petri net model which permits the modelling and verification of interorganizational workflows. The model allows the explicit representation of the organizational dimension of each component workflow, the shared use of resources among different organizations as well as the specification of security constraints. A property of soundness, which describes the correct execution of the interorganizational workflow, is defined and proven decidable.

Keywords: interorganizational workflows, modelling, verification, security constraints, Petri nets.

1 Introduction

A workflow is defined as a set of coordinated tasks that take place inside an organization in order to achieve a common business objective. Tasks may be carried out by resources (such as human users) and they involve specific data. Thus, several aspects (dimensions) of a workflow can be identified: the process dimension (which specifies the order of tasks), the organizational dimension, the data-flow dimension. Interorganizational workflows consist of several workflows belonging to different organizations, which execute independently but need to coordinate their actions at certain points, in order to accomplish a common goal.

One direction in the current research related to interorganizational workflows aims at finding suitable formal models to permit their specification and the analysis of properties such as the correct execution. These issues become increasingly difficult to address when considering details related to the organizational aspect of the workflows and security constraints. Among the formal methods which have been proposed for modelling interorganizational workflows, Petri nets present the advantage of an intuitive graphical representation, besides their formal semantics. Their use has been proposed in: [2, 3, 5, 10, 12, 15, 16]. The approaches in [2, 3, 8, 12, 18] study a soundness property describing the correct execution

of interorganizational workflows, but do not address the organizational dimension of the component workflows and do not impose security constraints. The organizational aspect is considered in [5, 17, 19]. The correctness problem is only studied in [17], which proposes a nested Petri net model, but does not include security constraints. The importance of considering security constraints for interorganizational workflows has been emphasized in papers like ([6, 11, 16]), but few approaches consider the verification of the correct execution under security constraints. The approach in [6] uses Petri nets for describing the interorganizational workflow, allows the specification of the organizational aspect and of security constraints, but does not address any correctness issues. In [11], the authors provide access control mechanisms inside the component organizations and use Petri nets for verification, but only in the process dimension. In [16], multilevel security features are included in the interorganizational workflow and the soundness property is discussed, but the approach abstracts from the organizational dimension of workflows.

This paper introduces Nested Interorganizational Workflow Nets (NIWF-nets) - an approach for the modelling of interorganizational workflows which permits the specification of security constraints. In our organizational model, permissions to execute certain tasks are associated to roles, where a role represents a class of users with similar capabilities. In order to increase flexibility and efficiency, we allow certain roles to be used by external organizations for the execution of tasks. Our model permits the enforcing of security constraints: we consider separation of duties constraints (SOD), which have been identified as an efficient mechanism to prevent fraud inside an organization ([4, 7, 13]). Such constraints can specify that a user who executed a task in the current workflow instance, cannot execute a set of other tasks, in order to prevent fraudulent actions of the user. SOD constraints are particularly useful when certain roles are involved in executing tasks from different organizations, as they can be used to avoid conflict of interests and undesired access to sensitive information.

Our approach is based on nested Petri nets ([14]), in which tokens may be Petri nets themselves (object-nets). In our model, there exist object-nets for every local workflow and every role in each organization. The role-nets will incorporate the mechanisms needed to enforce the SOD constraints. The communication between component workflows is modelled using a communication net. We will define and analyze a soundness property which describes the correct execution of the interorganizational workflow, under security and resource constraints.

2 Nested Interorganizational Workflow Nets

In this section we propose a model for interorganizational workflows, NIWF-nets (nested interorganizational workflow nets), based on nested Petri nets ([14]).

We will first present an introductory example of our model. We consider an interorganizational workflow which consists of two workflows, WF_1 and WF_2 . WF_1 belongs to a company which manufactures products, which are designed by WF_2 . The existing roles are: *Analyst* and *Clerk* in WF_1 and *Designer* in

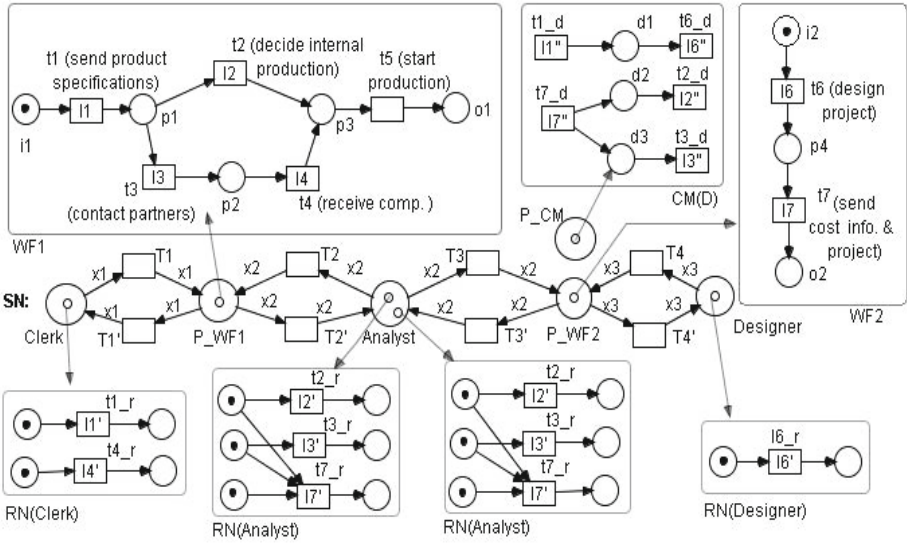


Fig. 1. An example of a NIWF-net modelling an interorganizational workflow

WF_2 . In WF_1 (Fig.1) a *Clerk* sends the product specifications to WF_2 (t_1). Upon receiving a request, in WF_2 a *Designer* designs the product (t_6). Then, the cost of the product is established by an *Analyst* and the project is sent to WF_1 (t_7). After receiving the project, in WF_1 an *Analyst* role can decide the internal production (t_2) or that components of the product should be made by external partners (t_3). After the components are received by a *Clerk* (t_4), the production can be started (t_5 - does not require a role for its execution). The coordination of the two workflows is described using the dependency relation between tasks: $\mathcal{D} = \{(t_1, t_6), (t_7, t_2), (t_7, t_3)\}$. Thus, task t_6 should fire after t_1 , and t_2, t_3 after t_7 in WF_2 . In order to avoid conflict of interests, since t_2 and t_3 in WF_1 and t_7 in WF_2 are executed by an *Analyst* role, one should impose that t_2 (t_3) in WF_1 and t_7 in WF_2 should not all be executed by the same user with the *Analyst* role. Thus, the SOD constraints consist of one rule: $t_7 \Rightarrow \neg(t_2, t_3)$. In the *NIWF-net* in Fig. 1, there exist several object-nets: WF_1 and WF_2 are Petri nets describing the process aspect of the workflows. Given the relation \mathcal{D} , the communication object $CM(\mathcal{D})$ contains transitions corresponding to the transitions involved in \mathcal{D} . A transition t_d in $CM(\mathcal{D})$ has a label l'' if its corresponding task t has the label l . For every role r , there exists a role-net, $RN(r)$. Since the *Analyst* role can execute t_2, t_3 and t_7 , there exist corresponding transitions in $RN(Analyst)$, with labels matching those of t_2, t_3 and t_7 (l'_2, l'_3, l'_7). Because of the SOD rule, t_{7_r} has as input places the input places of t_{2_r} and t_{3_r} . The system net SN consists of two places for WF_1 and WF_2 , a place for each role and a place for $CM(\mathcal{D})$. The transitions in SN move the role-nets between the role places and the workflow places. In the marking in Fig.1, t_1 in WF_1 and t_6 in WF_2 are enabled. Because t_6 is labelled and it is involved in \mathcal{D} , this means

it can only fire at the same time with t_{6_r} in $RN(Designer)$ and t_{6_d} in $CM(\mathcal{D})$ (i.e. there should be a *Designer* role available at P_{WF_2} and the communication module should permit the firing of this task). Since t_{6_d} is not enabled, t_6 cannot be executed. If T_1 fires in SN , a user with the role *Clerk* is moved to place P_{WF_1} . Now, a synchronization step (t_1, t_{1_r}, t_{1_d}) is enabled and its firing adds tokens to places p_1, d_1 and to the output place of t_{1_r} . After a role-net $RN(Designer)$ is moved to place P_{WF_2} by firing T_4 , the synchronization step (t_6, t_{6_r}, t_{6_d}) is also enabled. Assume that an Analyst role-net has been moved to place P_{WF_2} , the synchronization step (t_7, t_{7_r}, t_{7_d}) fired (i.e an analyst executed t_7) and than the *Analyst* role-net is moved back to place *Analyst*. In the marking of this object-net $ob_2 = (RN(Analyst), m_1)$, there are no tokens in the input places of its transitions. The same analyst cannot execute t_2 in WF_2 : if T_2 fires and ob_2 is moved to P_{WF_1} , although t_{2_d} is enabled in $CM(\mathcal{D})$, t_{2_r} is not enabled in ob_2 .

In what follows, we will define the specification of an interorganizational workflow, and than, the NIWF-net associated to it.

Each component workflow is executed inside an organization, which must define the roles responsible with the execution of tasks. Several users can be assigned to each role. We permit the existance of external roles: the users assigned to these roles can execute tasks belonging to other workflows. The interaction between the component workflows is specified using a partial relation on those tasks which need to be coordinated. Separation of duties rules can be defined in order to ensure the secure execution of the interorganizational workflow. If a role is authorized to execute several tasks (within or outside the organization), one can restrict a user with this role to execute some possible conflicting tasks.

Thus, a specification \mathcal{S} for an interorganizational workflows is a tuple $S = \langle WF, Roles, roles, \mathcal{D}, SOD \rangle$ such that:

- $WF = \{WF_1, \dots, WF_n\}$ is the set of all the component workflows. Let \mathcal{T} be the set of all the tasks in these workflows.

- *Roles* is the set of all the roles in the component workflows.

- *roles* : $\mathcal{T} \rightarrow 2^{Roles}$ is a function which specifies, for a task, the set of roles which are permitted to execute it.

- the dependency relation $\mathcal{D} \subseteq \mathcal{T} \times \mathcal{T}$ is a partial order which defines the dependency between tasks belonging to different workflows. If $(t_1, t_2) \in \mathcal{D}$, this means that t_1 must execute before t_2 .

- *SOD* is the set of separation of duties rules: a SOD rule $t \rightarrow \neg(t_1, \dots, t_k)$ specifies that, if a certain user with a role r authorized to execute tasks t, t_1, \dots, t_k , executes t , then it cannot execute anymore none of the tasks t_1, \dots, t_k . Also, if the user executes one of t_1, \dots, t_k , he cannot execute t . If $t \rightarrow \neg(t_1, \dots, t_k)$, then it also holds that $t_i \rightarrow \neg(t), \forall i \in \{1, \dots, k\}$. Tasks t, t_1, \dots, t_k could belong to different workflows. SOD rules can be used to avoid fraudulent actions of a user inside one organization, but also to avoid conflict of interests, if a user executes sensitive tasks in different organizations.

We will use *workflow-nets* (WF-nets [1]) to model the process dimension of workflows. A WF-net contains two special places, i and o , such that $\bullet i = \emptyset$ and $o \bullet = \emptyset$ and every element in the net belongs to a path from i to o . The marking with

one token in place i (denoted by i) is the initial marking of the net, while the marking with one token in o (denoted by o) is the final marking.

We will model each $WF_k \in \mathcal{WF}$ by a WF-net. i_k is the initial place of WF_k .

Given a specification \mathcal{S} and a role $r \in \mathcal{Roles}$, the corresponding *role-net* $RN(r) = (P, T, F)$ is obtained as follows:

- $T = \{t_r | \exists t \in \mathcal{T} : r \in \text{roles}(t)\}$

- $P = \cup_{t_r \in T} \{in_{t_r}, out_{t_r}\}$

- for each $t_r: (in_{t_r}, t_r), (t_r, out_{t_r}) \in F$; for each t_r such that $t \rightarrow \neg(t1, \dots, tn) \in \text{SOD}$, let in_{ti_r} be the input places of the transition ti_r corresponding to ti in $RN(r)$, for all $i \in \{1, \dots, n\}$. Then, $(in_{t1_r}, t_r), \dots, (in_{tn_r}, t_r) \in F$

An initial marking for $RN(r)$, denoted by mr_0 , contains one token in every input place. For each task that can be executed by a role r , there exists a corresponding transition t_r with one input place and one output place. If there exists a SOD rule involving the tasks which can be executed by r , $t \rightarrow \neg(t1, \dots, tn)$, then let $t_r, t1_r, \dots, tn_r$ be the transitions corresponding to the tasks in the rule. The input places of $t1_r, \dots, tn_r$ are connected to t_r , hence the firing of t_r prevents $t1_r, \dots, tn_r$ from firing. Also, if any transition ti_r fires, it disables t_r , which is consistent with the fact that ti and t_r cannot be executed by the same role. Thus, SOD constraints will be enforced through the structure of $RN(r)$.

Given a specification \mathcal{S} , a communication net associated to the dependency relation \mathcal{D} , $CM(\mathcal{D}) = (P_D, T_D, F_D)$, is defined as follows:

- $P_D = \{p_d | d \in \mathcal{D}\}$: a place for each pair of dependent tasks.

- $T_D = \{t_d | \exists (t', t) \in \mathcal{D} \vee (t, t') \in \mathcal{D}\}$: a transition for each task involved in \mathcal{D} .

- $F_D = \{(p_d, t_d) \in P_D \times T_D | d = (t', t) \in \mathcal{D}\} \cup \{(t_d, p_d) \in T_D \times P_D | d = (t, t') \in \mathcal{D}\}$

An initial marking for $CM(\mathcal{D})$, denoted by md_0 , is the empty marking.

Definition 1. Given a specification $\langle \mathcal{WF}, \mathcal{Roles}, \text{roles}, \mathcal{D}, \text{SOD} \rangle$, a nested interorganizational workflow net, NIWF, is a tuple

$NIWF = (Var, WF_nets, Role_nets, CM(\mathcal{D}), SN, L, \lambda)$ such that:

- $Var = \{x_r | r \in \mathcal{Roles}\}$ is the set of variables in the net.
- $WF_nets = \{WF_1, \dots, WF_n\}$ is set of WF-nets corresponding to \mathcal{WF} .
- $Role_nets = \{RN(r) | r \in \mathcal{Roles}\}$ is a set of role-nets.
- $CM(\mathcal{D})$ is an object-net (communication net), obtained from relation \mathcal{D} .
- $SN = (P_{SN}, T_{SN}, F_{SN}, M_0)$ is the system net:
 - P_{SN} contains a workflow place P_{WF} for each workflow WF , a role place P_r for each role $r \in \mathcal{Roles}$ and a place P_{CM} .
 - T_{SN} and F_{SN} contain transitions and labelled arcs connecting role places to workflow places: Let P_r be a role place. For each workflow WF_i in which r can execute tasks, $\exists T_i, T'_i \in T_{SN}$ such that $(P_r, T_i), (T_i, P_{WF_i}), (P_{WF_i}, T'_i), (T'_i, P_r) \in F_{SN}$ and all these arcs are labelled with the variable x_r .
 - M_0 assigns marked object-nets to the places of SN : $M_0(P_{WF_k}) = (WF_k, i_k)$, for all $k \in \{1, \dots, n\}$, $M_0(P_r) = n'_r(RN(r), mr_0)$ ($n_r \in \mathbb{N}$) for each $r \in \mathcal{Roles}$ (i.e. there are n_r role-nets $RN(r)$ in P_r) and $M_0(P_{CM}) = (CM(\mathcal{D}), md_0)$.
- $L = L_{WF} \cup L_R \cup L_D$ is a set of labels: L_{WF} is used for transitions in the component workflows; $L_R = \{l' | l \in L_{WF}\}$; $L_D \subseteq \{l'' | l \in L_{WF}\}$.

- λ is a partial labelling function on transitions, such that:
 - tasks in the workflows can have a label from L_{WF}
 - for each $t \in \mathcal{T}$, $r \in \text{Role}(t)$, let t_r be the transition corresponding to t in $RN(r)$. Then, $\lambda(t_r) = l'$, where $l = \lambda(t)$
 - if $t \in \mathcal{T}$ such that $(t, t') \in \mathcal{D}$ or $(t', t) \in \mathcal{D}$, then for $t_d \in T_D$: $\lambda(t_d) = l''$, where $l = \Lambda(t) \in L_{WF}$.

A net-token is a marked object-net. Let A_{net} be the set of all net tokens.

A marking of NIWF is a function M which assigns to each place P a multiset of net-tokens: $M(P) : A_{net} \rightarrow \mathbb{N}$.

Let T a transition in SN , P its input place and P' its output place. Let $P_r \in \{P, P'\}$ be the corresponding role place connected to it and $\text{var}(T) = x_r$, where x_r is the variable labelling the arc between P_r and T .

- A binding b associated to T assigns to x_r a net-token $(RN(r), m)$.
- If M is a marking of a NIWF-net, T is enabled in M w.r.t. a binding b if $b(\text{var}(T)) \in M(P)$. The firing of T produces a new marking, M' : $M[T[b]]M'$, such that: $M'(P) = M(P) - b(\text{var}(T))$ and $M'(P') = M(P') + b(\text{var}(T))$.

There are several types of steps, defining the behaviour of NIWF-nets:

(1) **Transport step:** If T is a transition in SN enabled in a marking M w.r.t. a binding b , then T is a transport step.

(2) **Object - autonomous step:** Let M be a marking, P a place of SN , $\alpha = (EN, m) \in M(P)$ and t a transition in α such that $\lambda(t)$ is undefined. t is an object-autonomous step enabled in M if $m[t]m'$. The resulting marking of NIWF, M' , is obtained from M by replacing, in $M(P)$, the net-token α with the net-token $\alpha' = (EN, m')$.

(3) **Synchronization step:** Let M be a marking, P a place of SN . Let $\alpha_1 = (EN_1, m_1)$ and $\alpha_2 = (EN_2, m_2)$ two net-tokens in $M(P)$ such that (1) $\exists t$ in (EN_1, m_1) with $\lambda(t) = l \in L_{WF}$ and $m_1[t]m'_1$; (2) $\exists t'$ in (EN_2, m_2) with $\lambda(t') = l' \in L_R$ and $m_2[t']m'_2$; (3) if $\exists t''$ in $(CM(\mathcal{D}), m)$ with $\lambda(t'') = l'' \in L_D$, then $m[t'']m'$. Then, we say that $Y = (t, t', t'')$ (or $Y = (t, t')$, if t'' does not exist) represents a synchronization step enabled in marking M . The resulting marking, M' , is obtained from M by replacing in $M(P)$ the net-token α_1 with the net-token $\alpha'_1 = (EN_1, m'_1)$, the net-token α_2 with the net-token $\alpha'_2 = (EN_2, m'_2)$ and $(CM(\mathcal{D}), m)$ with $(CM(\mathcal{D}), m')$ in $M(P_{CM})$.

If t in α_1 has a label $l \in L_{WF}$, then α_1 is a WF-net. If t' in α_2 has a label l' , then α_2 is a role-net and t' is a transition t_r corresponding to t . If there exists t'' in $CM(\mathcal{D})$ with a label l'' (i.e. $t'' = t_d$ and t is involved in external interaction with transitions from other workflows) t'' should be enabled and t , t' and t'' must fire synchronously, that is: a user assigned to role r , when executing task t , interacts with the communication module (sends a message) in order to check if the external conditions for executing the task are met.

3 Verification of Interorganizational Workflows

A notion of soundness was defined for WF-nets, expressing the minimal conditions a correct workflow should satisfy ([1]). A WF-net is sound iff (1) for every

marking m reachable from marking i , there exists a firing sequence from m to marking $o : (\forall m)((i[*]m) \implies (m[*]o))$; (2) all the transitions in WF can fire.

A final marking of a NIWF-net is a marking in which all the object roles reside in their initial places and each workflow place only contains the corresponding workflow, in a final marking. We denote by \mathcal{M}_f the set of final markings. A marking in the set \mathcal{M}_f is reachable iff all the component workflows successfully ended their execution, even when their behaviour is restricted by the SOD and resource constraints and by the communication mechanisms.

Definition 2. *A nested interorganizational workflow net NIWF is sound iff:*

(1) (WF_j, i_j) is a sound workflow net, $\forall j \in \{1, \dots, n\}$.

(2) Every step Y should be able to fire: $(\exists M \in [M_0] : M[Y])$.

(3) For every reachable marking M , there exists a firing sequence to a marking from \mathcal{M}_f : $(\forall M)((M_0[*]M) \implies (M[*]M_f, M_f \in \mathcal{M}_f))$.

(4) The communication net is bounded: $(\exists n \in \mathbb{N})(\forall M \in [M_0], (CM(\mathcal{D}), m) \in M(PCM))(\forall p_d \in P_D : m(p_d) \leq n)$.

Condition (4) requires that the participant workflows should not be allowed to send an infinite number of messages to the communication module.

Using the results from [9, 14], it can be proven that the boundedness problem is decidable for NIWF-nets. Also, it holds:

Lemma 1. *Assume NIWF is a NIWF-net, such that all the component WF-nets are sound. Then, NIWF is bounded iff the communication net is bounded.*

Theorem 1. *The soundness is decidable for NIWF-nets.*

Condition (1) in Def. 2 is decidable ([1]). Condition (2) is also decidable (it is decidable for general nested Petri nets: [14]). The boundness of NIWF is decidable. If (1) holds and NIWF is unbounded, then condition (4) does not hold (Theorem 1), so the NIWF-net is not sound. If NIWF is bounded, then (4) holds and $[M_0]$ is finite. Since \mathcal{M}_f is finite, condition (3) is also decidable.

4 Conclusions and Future Work

In this paper we introduced a Petri net model, NIWF-nets, for interorganizational workflows which permits the specification of roles in the component workflows and the mobility and sharing of resources across organizations. Our model allows the specification of security constraints and offers a modular view of the interorganizational workflow. We defined a soundness property for NIWF-nets and proved its decidability. Our future work aims at extending the organizational model and including other security constraints.

References

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

2. van der Aalst, W.: Loosely coupled interorganizational workflows: Modeling and analyzing workflows crossing organizational boundaries. *Information and Management* 37(2), 67–75 (2000)
3. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: From Public Views to Private Views – Correctness-by-Design for Services. In: Dumas, M., Heckel, R. (eds.) *WS-FM 2007*. LNCS, vol. 4937, pp. 139–153. Springer, Heidelberg (2008)
4. Ahn, G., Sandhu, R.: The RSL99 Language for Role-based Separation of Duty Constraints. In: *4th ACM Workshop on Role-Based Access Control*, pp. 43–54. ACM, New York (1999)
5. Andonoff, E., Bouaziz, W., Hanachi, C.: An Agent-Based Model for Autonomic Coordination of Inter-Organizational Business Processes. *INFORMATICA* 20(3), 323–342 (2009)
6. Ayed, S., Cuppens-Boulahia, N., Cuppens, F.: Deploying Access Control in Distributed Workflow. In: *The 6th Australasian Conference on Information Security*, pp. 9–17. Australian Computer Society Inc., Darlinghurst (2008)
7. Bertino, E., Ferrari, E., Atluri, V.: The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Trans. on Inf. and Sys. Sec.* 2(1), 65–104 (1999)
8. Ge, J., Hu, H., Lu, P., Hu, H., Lü, J.: Translation of Nets Within Nets in Cross-organizational Software Process Modeling. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) *SPW 2005*. LNCS, vol. 3840, pp. 360–375. Springer, Heidelberg (2006)
9. Finkel, A., Schnoebelen, P.: Fundamental Structures in Well-Structured Infinite Transition Systems. In: Lucchesi, C.L., Moura, A.V. (eds.) *LATIN 1998*. LNCS, vol. 1380, pp. 102–118. Springer, Heidelberg (1998)
10. Jiang, P., Shao, X., Gao, L., Qiu, H., Li, P.: A process-view approach for cross-organizational workflows management. *Adv. Eng. Inf.* 20(2), 229–240 (2010)
11. Kang, M.H., Park, J.S., Froscher, J.N.: Access control mechanisms for inter-organizational workflows. In: *The 6th ACM Symposium on Access Control Models and Technologies*, pp. 66–74. ACM, New York (2001)
12. Kindler, E., Martens, A., Reisig, W.: Inter-operability of Workflow Applications: Local Criteria for Global Soundness. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) *BPM 2000*. LNCS, vol. 1806, pp. 235–253. Springer, Heidelberg (2000)
13. Knorr, K., Weidner, H.: Analyzing Separation of Duties in Petri Net Workflows. In: Gorodetski, V.I., Skormin, V.A., Popyack, L.J. (eds.) *MMM-ACNS 2001*. LNCS, vol. 2052, pp. 102–114. Springer, Heidelberg (2001)
14. Lomazova, I.A.: Nested Petri Nets - a Formalism for Specification and Verification of Multi - Agent Distributed Systems. *Fund. Informa.* 43, 195–214 (2000)
15. Lohmann, N., Massuthe, P., Stahl, C., Weinberg, D.: Analyzing Interacting WS-BPEL Processes Using Flexible Model Generation. *Data Knowl. Eng.* 64(1), 38–54 (2008)
16. Mikolajczak, B., Gami, N.: Design and Verification of Loosely Coupled Inter-Organizational Workflows with Multi-Level Security. *JCP* 3(1), 63–78 (2008)
17. Prisecaru, O.: A Petri net based approach for modelling resource constrained interorganizational workflows. In: Ultes-Nitsche, U., Moldt, D., Augusto, J.C. (eds.) *MSVVEIS 2008*, pp. 29–38. Insticc Press (2008)
18. Yamaguchi, S., Matsuo, H., Ge, Q., Tanaka, M.: WF-Net Based Modeling and Soundness Verification of Interworkflows. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* E90-A(4), 829–835 (2007)
19. Wirtz, G., Giese, H.: Using UML and object-coordination-nets for workflow specification. In: *2000 IEEE International Conference on Systems Man and Cybernetics Cybernetics*, pp. 3159–3164. IEEE Press (2000)