Security Policies in Adaptive Process-Aware Information Systems: Existing Approaches and Challenges

Maria Leitner
University of Vienna
Vienna, Austria
maria.leitner@univie.ac.at

***Abstract*—Enabling security is one of the key challenges in adaptive Process-Aware Information Systems (PAIS). Since automating business processes involves many participants, uses private and public data, and communicates with external services security becomes inevitable. In current systems, security is enforced by an access control model and supplementary constraints imposed on workflow activities. However, existing systems provide individual implementations for security policies (e.g. separation of duties) and leave out other constraints (e.g. inter-process constraints). What is missing is a systematic analysis of security policies in PAIS. Hence, in this paper, we display state of the art and provide a taxonomy of security policies in PAIS. Furthermore, a detailed analysis of research challenges and issues is presented. We will show that there are still shortcomings and identify important requirements for security in PAIS. We will also point out open questions related to specifying, modeling, and changing security policies which will provide a road map for future research.**

***Keywords*-Security Policies; Process-Aware Information Systems;**

I. Introduction

Process-Aware Information Systems (PAIS) support the automation of business processes carried out by various participants using private and public data (e.g. bank account). Due to highly sensitive data processed by workflows and the multitude of different performers participating in workflows, the need for security is inevitable. In the case of adaptive PAIS (e.g. ADEPT [1]), where change and modification (e.g. control flow changes) are supported at various levels, enforcing security gets even more complicated.

Current PAIS provide security by using an access control model and further constraints (e.g. in [2]). However, the support of all workflow-related security policies (e.g. inter-process constraints) is not always given. Furthermore, supplementary rules such as laws also have to be enforced. Hence, there are still open research questions. What is missing is a systematic analysis of security policies in PAIS. With such a comprehensive investigation, we can further identify missing links and open issues. So far, the focus of current research is mainly on build time but should be extended to run time and change time.

In this paper, we are going to address challenges related to security policies in PAIS. First, we present current research and then provide a taxonomy on security policies. Further on, open research questions and issues of adaptive PAIS are examined. We demonstrate that there are still shortcomings and identify requirements to enable all security policies in PAIS.

The paper is structured as follows: Section II gives an overview of security policies in workflow systems. In Section III, research challenges and issues are discussed and requirements for security policies are identified. Section IV concludes the paper.

II. Security Policies in Adaptive PAIS

Security policies are a set of statements of a systems protection strategy [3]. In PAIS, security policies are often related to role-based access control restrictions or constraints (e.g. separation of duties). However, security policies require a more detailed definition due to the multi-faceted characteristics of workflow systems. Specifically, security policies in PAIS might relate to access control, control flow, information flow, data integrity, and availability. Therefore, policies can be specified for users, information (data), control flow, activities, and process instances. Hence, constraints can be enforced at build time (*static* constraints) and run time (*dynamic*) [2]. In PAIS, it is very common to refer to "constraints" when talking about security policies. Therefore, we will use both terms in this paper, signifying that they enforce a security guideline (e.g. authorization).

A. State of the Art of Security Policies

Commonly, security is enforced in PAIS by using an access control model which relies on an authentic organizational model (e.g. from a company with organizational units and roles). Permissions are associated with roles e.g. the role *Accountant* is authorized to execute the activity *PayBill*. For example, the Role-Based Access Control (RBAC) model (e.g. NIST RBAC [4]) expresses security policies based on the role-permission assignments and is commonly used to restrict access in information systems. Process-related RBAC models (e.g. Workflow RBAC [5]) have been developed using supplementary workflow concepts such as workflow instances (cases). Furthermore, delegation in terms of a

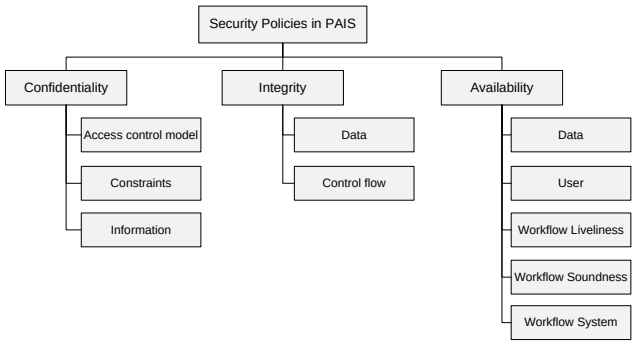


Figure 1. Taxonomy of Security Policies

reassignment of tasks to other agents has been investigated in e.g. [6].

Additionally, constraints are imposed on workflow activities enhancing access control decisions (e.g., separation of duties, time). Constraints can also be included in the access control model (e.g. [5]). The specification and enforcement of constraints in PAIS are displayed in e.g., [2]. Furthermore, external (often legally binding) guidelines are often demanded e.g. regulations as Sarbanes-Oxley Act (SOX) and Basel II or recommendations such as the NIST Special Publications on security. These rules have to be enforced within PAIS resulting in supplementary constraints.

Two surveys classify security requirements in business processes: In [7], the relation between process elements and security objectives is discussed. Compliance is enabled by process security modeling in [8]. Therefore, research approaches are compared by security criteria. Both surveys use in their classifications the core security objectives: confidentiality, integrity, and availability (also known as the CIA triad) as well as supplementary concepts (e.g. non-repudiation). In this paper, we use abstract concepts (e.g. users) to enable a generalized view on security policies in adaptive PAIS. Furthermore, we focus on *all* aspects of security policies in PAIS such as modeling or their application (e.g. mapping to activities) and *at all times* (build, run, and change time).

B. Taxonomy of Security Policies in adaptive PAIS

We created a taxonomy of security policies in PAIS (cf. Fig. 1) categorized by the *main* key concepts of information security: confidentiality, integrity, and availability (also used by e.g. [7], [8]). The classification can be extended with further security objectives such as non-repudiation or privacy.

- **Confidentiality:** In PAIS, confidentiality is usually ensured by an access control model and constraints associated with activities. Information should only be accessible to authorized users.
- **Integrity:** A security policy for the integrity of a control flow signifies that, for example, a certain activity has to be finished before another activity starts (e.g. activity *PayQuotation* has to be completed before *SendShipment*). Integrity of data means that no user who is unauthorized to access the data can modify it. Therefore, only authorized actions are carried out on data.
- **Availability:** In PAIS, availability may refer to the system, resources (e.g. data, users), or the control flow which can be verified with the workflow liveliness and soundness.

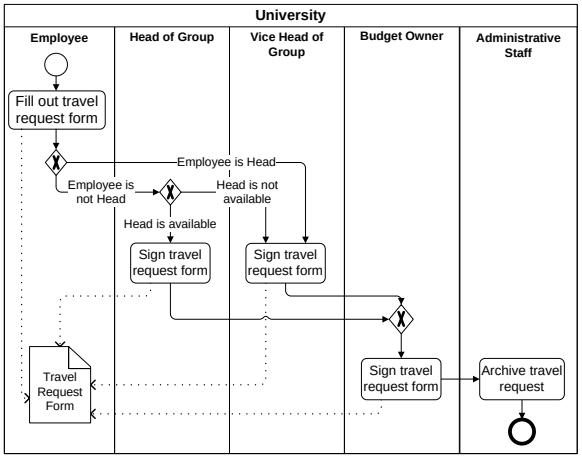
III. Challenges

In the following section, we will provide an overview of challenges and not yet fully discussed research problems related to security policies in PAIS. Furthermore, a problem description and requirements for each challenge are given. Challenges result from an extensive literature review and many case studies from various domains. We do not claim that the list of challenges is exhaustive and can be extended. To illustrate our findings, we introduce the following example: a travel request within the university domain. In a nutshell the process model depicted in Fig. 2(c) contains the following tasks: `Fill out travel request` requires that an *employee* has to fill in the required information for the travel request consisting of information on use (e.g. name), travel date (e.g. start and end date), budget information, signature, and date of signature. `Sign travel request` implies that a *head of group* and a *budget owner* perform this task and have to sign the travel request. But the travel request must be approved by two different persons meaning that the user executing `Sign travel request B` must be different from the user performing `Sign travel request H` (separation of duty). Finally, the travel request is archived (activity `Archive travel request`) by *administrative staff*.

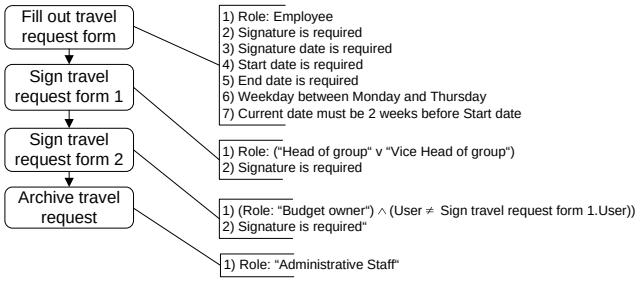
In addition to these process activities, a set of security policies should hold as depicted in Fig. 2(c) at the right. For example, if the *head of group* is not available then, the *vice head* can sign the travel request. Actually these policies should hold for all three process models (cf. Fig. 2(a)-2(c)).

A. Challenge 1: Modeling Security Policies

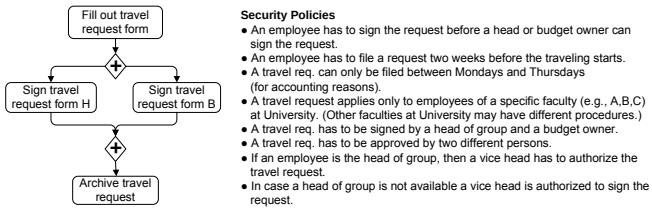
1) Policy Modeling: Process models can be specified traditionally (i.e. imperative) or constraint-based (i.e. declarative) (cf. Fig. 3). Whereas the control flow is modeled in the imperative approach, it has to be specified by constraints in the declarative approach (cf. [9]). Fig. 3 displays the same processes in two ways: the imperative model is displayed in Business Process Modeling Notation (BPMN) in Fig. 3(a) and the declarative model in ConDec in Fig. 3(b) (cf. [10]). All policies are stringently assigned to tasks in the imperative model (see Fig. 3(a)). In the declarative approach, a mapping function has to associate the tasks with the corresponding rules (cf. Fig. 3(b)).



(a) Inherent Representation



(b) Attached Representation

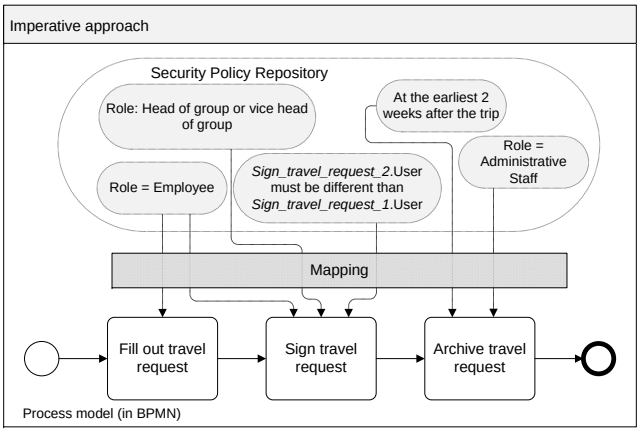


(c) Separated Representation

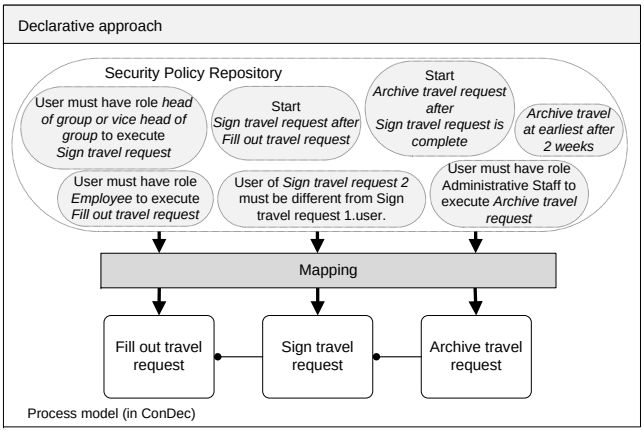
Figure 2. Travel Request: Process Modeling

Problems: While the imperative approach is very strict and definite, it may not be reasonable for certain domains where ad hoc decisions based on circumstances have to be made. In terms of security, a security expert has to examine all possibilities and specify all policies in advance in the declarative model. Especially in large systems, it is difficult to oversee all regulations, possible vulnerabilities, or risks for all processes. This might be demanding in a declarative model because *all* potential occurrences have to be examined.

Requirements: It is important to consider the policy modeling approach depending on the domain of the PAIS. The declarative approach presents a more flexible way for integrating ad hoc changes. However, enforcing security can be difficult because potential process paths have to be analyzed and the risks of threats minimized. The imperative modeling type enables a stringent security policy definition



(a) Imperative Modeling



(b) Declarative Modeling

Figure 3. Travel Request: Policy Modeling

(e.g. authorization) and is therefore more solid because not all potential pathways have to be foreseen.

2) Process Modeling: In PAIS, the (contextual) range of security policies can be very wide such as access control, data, or control flow guidelines as shown in Section II. Therefore, this "information mix" often results in a mixed specification and representation of security policies in PAIS, i.e. security policies in PAIS can be expressed and implemented in an *inherent*, *attached*, or *separate* way. **Inherent** security policies are part of the control and data flow of the process model. For example, checking if the *head of group* is available or not is explicitly defined as a decision point in the control flow as depicted in Fig. 2(a). **Attached** security policies are explicitly defined at design time as attributes of process activities. In Fig. 2(b), for example, a role restriction for *budget owner* is defined as attribute of activity `Sign travel request form 2`. Most research approaches dealing with authorization constraints such as [2] use attached modeling. In the **separated** representation, security policies are specified indirectly and are enforced at run time e.g. [11]. In the separate representation, the control flow is explicitly defined in the policies (redundantly to the

process model) and the matching to tasks occurs at run time.

Problems: Current commercial systems mostly use attached and inherent process modeling but research prototypes tend to focus on a separate policy implementation [12]. This distinction leads to various research scenarios. Imagine, that internal university guidelines change: if an *employee* is not the *head* then the *dean* (instead of the *vice head*) has to sign the document if the *head* is not available (see Fig. 2(a)-2(c)). In the inherent representation, finding and evaluating the security policies can be cumbersome and quite difficult considering *all* information is stored in the data and control flow of the process model. In this example, a new swim lane has to be created (*dean*) and the task has to be moved to the lane. In the attached modeling approach, only the policies associated to the activity have to be adjusted. So, the activities are examined and assigned policies are updated, but the process model does not have to be modified. In the separate representation, the relevant rule in the policy repository is configured.

Requirements: In general, it should be possible to separate the process model and the security policies from each other. Then, the checks for inconsistencies are more efficient in a repository (than e.g. going through each activity in a process). A repository supports also the administration of policies such as adding, updating or deleting which is essential due to continuously changing business requirements. Furthermore, separating process models from policies facilitates the evolution of processes. If the policies are not included in the data and control flow of the model, the process models can be significantly reduced to a minimal set of tasks as shown in Fig. 2(c).

3) Modeling Extensions: Currently, conceptual modeling approaches exist at a security objective level. For example in [8], several modeling approaches are discussed showing that some use modeling extensions e.g. to display anonymity in Unified Modeling Language (UML) diagrams [7]. At a functional level, extensions have been made to UML and BPMN: An UML approach provides a vocabulary for enabling access control definitions in UML-based models (e.g. [13]). In [14], an extension for BPMN to model task-based entailment constraints such as authorization constraints (e.g. separation of duties) is proposed.

Problems: Current approaches provide only some security function modeling and are at a very early stage. They neither provide patterns for all security policies nor specify a standardized vocabulary for enforcing security in PAIS. Current proposals use different symbols or text to display security. Modeling security policies in PAIS might include further challenges such as visualization. Imagine a large process model: Is it possible to present security-critical information in large models?

Requirements: It is necessary to identify the requirements to enable security modeling extensions for standard notations. We require to investigate how to display semantical or technical security features. In large process models, the model should be kept simple and complexity should not increase due to security extensions.

In summary, the discussion shows that PAIS express security policies in different ways. Current approaches differ in their concrete process and policy modeling.

B. Challenge 2: Separating Security Policies from other Constraints

Because of the existence of e.g., guidelines, regulations, or further rules influencing PAIS, it is difficult to find a clear distinction between security policies and other rules. A classification of constraints in PAIS is shown in [15] but security constraints are only considered on a structural level. In this paper, the semantical level is also considered.

Problems: Imagine a PAIS within the health care domain. There are a lot of guidelines that have to be included in the processes such as medical guidelines (e.g. *the patient has to take its medication twice a day*), public health guidelines (e.g. *inform the local public health department if a patient has tuberculosis*), law, and budget restrictions (e.g. *put at least 4 patients in one room*). However, all guidelines have to be integrated in the processes. But which guidelines can be specified as security policies?

Requirements: Security guidelines originate from various sources and have to be incorporated in PAIS. In this paper we propose that security policies in PAIS should be related to the security objectives confidentiality, integrity, and availability (cf. Section II-B). For example, the guideline *"A surgery can only be performed with two doctors"* is user-centric and relates to availability. Therefore, the rule can be categorized as a security policy in PAIS. This approach can be extended to other security principles (e.g. privacy).

C. Challenge 3: Mapping Policies to Process Activities

Security policies can be associated with process activities in an *inherent*, *attached*, or *separated* way (cf. Section III-A2). The inherent method uses mainly roles associated with permissions to set security policies but does not enforce all types of security policies (e.g. inter-process constraints). That is why commonly supplementary constraints are imposed on tasks. Current systems mostly use an attached approach where policies are assigned to the corresponding tasks. However, the association of the policies can be cumbersome and inefficient to administrate because all tasks have to be checked for adjusting a policy. The separated representation uses an independent repository where all policies are stored. However, a specific mapping function is needed to relate tasks and policies with each other.

In [16], workflow processes are verified against organizational security policies by transforming each to a common constraint language. Nevertheless, flexibility might be an issue due to the fact that the *whole* process model and all policies have to be translated in a common language to verify their compliance (even when minor changes occur).

Problems: Policies can be administered at build and run time. For example, dynamic constraints can only be enforced at run time (e.g. separation of duty, inter-process constraints). The inherent approach does not support all constraints such as inter-process constraints (cf. Section III-F). In the attached representation, the assignment of policies to each activity is done individually. Imagine, a large system with about 500 roles, 1500 activities, and 500 security policies. Associating each policy to many activities is troublesome and inefficient. In the separated approach, security policies have to be mapped to the corresponding activities. At this point it gets difficult: How can policies be associated with activities? Which criteria should be considered?

In a large system, scalability might be an issue. One policy can be assigned to one or more activities. Therefore, it should be possible to associate a policy with several activities at the same time.

Requirements: Ideally, there should be a mechanism that maps the security policies to process activities. However, to be able to handle the mapping, we need to know which rules should be associated with which activity. We demand an easy and manageable association of activities and policies at a fair level of complexity. Furthermore, we also require that scalability should be supported.

D. Challenge 4: Process Evolution

Current research manages secure process changes with access control models (e.g. [17]). But these proposals enforce mainly authorization constraints and not all security policies (e.g. inter-process constraints). The impact of process changes on security policies such as inconsistencies or conflicting policies has to be considered. Furthermore, in case of organizational model changes (e.g. outsourcing) the effects on constraints have to be examined. In [12], direct and indirect effects on constraints resulting from organizational changes are identified.

Problems: When a process changes such as add, delete, or move an activity all associated constraints have to be checked for correctness. For example, activity `Sign travel request form 2` is moved before `Fill out travel request` (cf. 2(b)). Apart from the fact that this sequence change is rather unlikely, the separation of duty constraint cannot be assured anymore. This can lead to unexpected results such as workflow blockage or exceptions.

Requirements: When a process changes, the corresponding policies (e.g. authorization constraints) should be validated. Because workflows can change at build and run time it is important to develop mechanisms to manage both scenarios. It should be possible to change an activity and to further maintain the security of the activity at the same level regardless of e.g. the changed control flow or data flow.

E. Challenge 5: Policy Evolution

The impact of security policy changes in PAIS has to be considered. Research has acknowledged the importance of avoiding inconsistencies in policies (e.g. [2]). Nevertheless, research has ignored security policy changes in PAIS.

Problems: Approaches focus mostly on build time strategies where policies are set and checked for conflicts (e.g. [18]). However, policies such as authorization constraints or legal regulations evolve over time. Imagine the banking sector, where regulations such as EU directives have to be implemented frequently. Already running workflow instances with defined security policies have to be changed at run time too. For example, new regulations e.g. laws apply by then and have to be enforced in all running instances.

Requirements: There is a need for administration of security policies such as adding, deleting, or updating rules. We require to enable an easy handling and maintenance of security policies at build, run, and change time. Therefore, security policies and process models should be administered separately.

F. Challenge 6: Inter-process Security Policies

In PAIS, process instances can be executed concurrently at run time. Hence, it might be necessary to enforce security policies over multiple process instances.

Problems: Current systems neither provide inter-process security policies nor enable a semantical support. However, the interaction between process instances becomes more important (e.g., in service-oriented architectures).

Imagine in the previous example, that due to budget restrictions the university allows for each employee to file only 20 travel requests per year (cf. Fig. 4). Further, the university limits the travel budget with 40.000 euros per year. So, if an employee travels to many long-distance destinations, the budget will be probably spent sooner than by short-distance flights. The total cost of all travels per employee should not exceed 40.000 euros per year. These restrictions should prevent misuse or fraudulent actions. However, so far research has mostly ignored inter-instance constraints. Only in [5], [19] inter-instance constraints are enforced over multiple instances.

Requirements: In PAIS, there is a growing need of more interaction between process instances. In particular, security policies should be enforced over multiple instances. When enabling interaction between instances, designers and practitioners have to tackle another challenge: How to model, implement, and enforce inter-instance constraints at a fair level of complexity.

IV. Conclusion

In this paper, a taxonomy for security policies in adaptive PAIS is presented. The main motivation behind is that while there are various constraints in workflow systems, a precise distinction between security policies and other rules

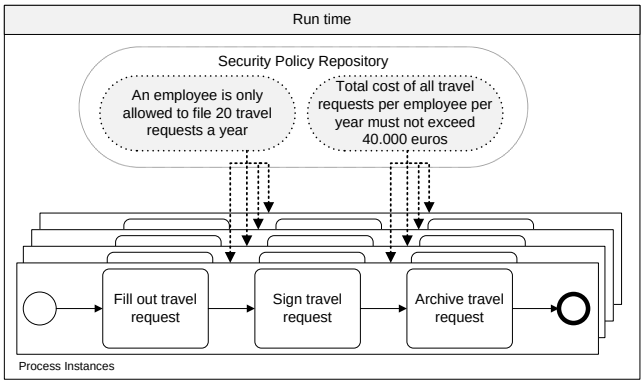


Figure 4. Inter-instance Security Policies

was missing. In our approach, constraints are categorized semantically by relating them to the security objectives confidentiality, integrity, or availability. Furthermore, we illustrated challenges and issues and identified requirements within the workflow domain. Challenges include distinguishing security policies from other rules, modeling policies within the process model, mapping policies to activities, process and policy evolution, and inter-process constraints. For example, only few research exists to include security related information in notations. In future work, we aim to tackle the remaining questions and will implement the concepts with a proof-of-concept prototype.

REFERENCES

[1] P. Dadam and M. Reichert, "The ADEPT project: a decade of research and development for robust and flexible process support," *Computer Science - Research and Development*, vol. 23, no. 2, pp. 81–97, May 2009.

[2] E. Bertino, E. Ferrari, and V. Atluri, "The specification and enforcement of authorization constraints in workflow management systems," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 1, pp. 65–104, 1999.

[3] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley Publishing, 2008.

[4] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 224–274, 2001.

[5] J. Wainer, P. Barthelmess, and A. Kumar, "W-RBAC - a workflow security model incorporating controlled overriding of constraints," *International Journal of Cooperative Information Systems*, vol. 12, no. 4, pp. 455–485, 2003.

[6] J. Wainer, A. Kumar, and P. Barthelmess, "DW-RBAC: a formal security model of delegation and revocation in workflow systems," *Information Systems*, vol. 32, no. 3, p. 365–384, 2007.

[7] P. Herrmann and G. Herrmann, "Security requirement analysis of business processes," *Electronic Commerce Research*, vol. 6, no. 3, pp. 305–335, Oct. 2006.

[8] M. Riesner and G. Pernul, "Supporting compliance through enhancing internal control systems by conceptual business process security modeling," in *ACIS 2010 Proceedings*, Jan. 2010.

[9] W. van der Aalst, M. Pesic, and H. Schonenberg, "Declarative workflows: Balancing between flexibility and support," *Computer Science - Research and Development*, vol. 23, no. 2, pp. 99–113, May 2009.

[10] M. Pesic and W. van der Aalst, "A declarative approach for flexible business processes management," in *Business Process Management Workshops*, ser. Lecture Notes in Computer Science. Springer, 2006, vol. 4103, pp. 169–180.

[11] M. Leitner, J. Mangler, and S. Rinderle-Ma, "Responsibility-driven design and development of process-aware security policies," in *Proceedings of the 6th International Conference on Availability, Reliability and Security (to appear)*. IEEE Computer Society, 2011.

[12] S. Rinderle-Ma and M. Leitner, "On evolving organizational models without loosing control on authorization constraints in web service orchestrations," in *CEC*, Shanghai, China, 2010.

[13] T. Lodderstedt, D. Basin, and J. Doser, "SecureUML: a UML-Based modeling language for Model-Driven security," in *UML 2002 - The Unified Modeling Language*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2460, pp. 426–441.

[14] C. Wolter and A. Schaad, "Modeling of Task-Based authorization constraints in BPMN," in *Business Process Management*, ser. Lecture Notes in Computer Science. Springer, 2007, vol. 4714, pp. 64–79.

[15] J. Mangler and S. Rinderle-Ma, "IUPC: identification and unification of process constraints," Tech. Rep., Apr. 2011. [Online]. Available: http://arxiv.org/abs/1104.3609

[16] C. Ribeiro and P. Guedes, "Verifying workflow processes against organization security policies," in *Proc. of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises*. IEEE Computer Society, 1999, pp. 190–191.

[17] B. Weber, M. Reichert, W. Wild, and S. Rinderle, "Balancing flexibility and security in adaptive process management systems," *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, p. 59–76, 2005.

[18] M. Strembeck and J. Mendling, "Generic algorithms for consistency checking of Mutual-Exclusion and binding constraints in a business process context," in *On the Move to Meaningful Internet Systems: OTM 2010*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6426, pp. 204–221.

[19] J. Warner and V. Atluri, "Inter-instance authorization constraints for secure workflow management," in *Proceedings of the eleventh ACM symposium on Access control models and technologies*. Lake Tahoe, California, USA: ACM, 2006, pp. 190–199.