Information Technology

Integration von Sicherheitsaspekten in Geschäftsprozessmodelle

Integrating Security Aspects into Business Process Models

Dr. Achim D. Brucker: SAP AG, Vincenz-Priessnitz-Str., 1, 76131 Karlsruhe, Germany
Tel: +49–62–277–52595, E-Mail: achim.brucker@sap.com
Achim D. Brucker is a Senior Researcher and Project Lead in the "Product Security Research Team" as well as a member of the "Code Analysis Team" of SAP AG. His research interests include security, software engineering, and formal methods. In particular, he is interested in tools and methods for modelling, building, validating, and verifying secure and reliable systems. He also participates in the OCL standardisation process of the OMG. Further information can be found on his website: `http://www.brucker.ch`.

Keywords: D.4.6 [Software]: Operating Systems—Security and Protection; K.6.5 [Computing Milieux]: Management of Computing and Information Systems; SecureBPMN; BPMN; Access Control; Confidentiality; Break-glass; Delegation

Schlagworte: Sichere Geschäftsprozessmodelle; SecureBPMN; BPMN; Zugriffskontrolle; Vertraulichkeit; Break-glass; Delegation

MS-ID:		
Heft: 55/6 (2013)		

Abstract

Modern enterprise systems are often process-driven and, thus, rely heavily on process-aware information systems. In such systems, high-level process-models play an important role both for communicating business requirements between domain experts and system experts as well as basis for the system implementation. Since several years, enterprise system need to fulfil an increasing number of the security and compliance requirements. Thus, there is an increasing demand for integrating high-level security and compliance requirements into process models, i. e., a common language for domain experts, system experts, and security experts.

We present a security modelling language, called SecureBPMN, that can easily be integrated into business process modelling languages. In this paper, we exemplary integrate SecureBPMN into BPMN and, thus, present a common language for describing business process models together with their security and compliance requirements.

Zusammenfassung

Moderne Unternehmensanwendungen müssen die Unternehmen dabei unterstützen, ihre Geschäftsprozesse effizient auszuführen. In solchen Anwendungen spielen abstrakte Geschäftsprozessmodelle eine zentrale Rolle. Die Geschäftsprozessmodelle werden für die Kommunikation zwischen Geschäfts- und IT-Experten genutzt und dienen darüber hinaus als Basis für die Implementierung der Unternehmensanwendungen. Seit einigen Jahren müssen Unternehmensanwendungen einer steigenden Anzahl von Sicherheits- und Compliance-Anforderungen genügen. Hieraus ergibt sich ein gesteigerte Bedürfnis nach der Integration von Sicherheits- und Compliance-Anforderungen in die Geschäftsprozessmodelle.

In diesem Artikel stellen wir die Modellierungssprache SecureBPMN vor, welche es erlaubt, Sicherheitsanforderungen im Kontext von Geschäftsprozessmodelle zu spezifizieren.

1 Introduction

Modern enterprise systems are often process-driven. In such systems, high-level process models, e. g., expressed in terms of the Business Process Modelling Language and Notation (BPMN) [21], play an important role. On the one hand, business process models are used for communication business requirements between business experts and system experts. On the other hand, process models are used for the actual system implementation, i. e., as runtime artefact of a business process execution engine. As modern business processes combine human tasks with automated tasks (e. g., implemented by web services), a business process modelling language needs to bridge the gap between the language used by business experts and the language used by system experts.

Since several years, enterprise system need to fulfil an increasing number of the security and compliance requirements. One reason is for this is that the number of businesses that operate in regulated markets, i. e., that need to comply to regulations such as HIPAA [14] in the health care sector or Basel II [4] in the financial sector, is increasing. Such compliance regulations in along with the increased awareness of IT security result in need for modelling, analysing, and execution techniques for business processes that treat security, privacy, and compliance properties in business processes as first class citizen. This leads to the need for complex and dynamic security policies [13, 16, 22].

Consequently, the demand for an integrating high-level security and compliance requirements into process models and, thus, a language that fulfils the need of business experts, system experts, and security experts, is increasing. Already fulfilling the needs of business experts *and* system experts, at the same time, is challenging—bringing the security experts to the same table, makes it even more challenging.

To meet this challenge, we developed SecureBPMN: a security modelling language for expressing high-level security and compliance requirements such as role-based access control (RBAC), break-glass, separation-of-duty (SoD), delegation, or variants of the need-to-know principle. SecureBPMN is defined using a metamodel which makes it particularly suitable for integration into business process languages that are themselves defined by a metamodel. In this paper, we present SecureBPMN as such and its integration into BPMN. This integration provides a language that allows for specifying, analysing and executing business processes securely.

The rest of the paper is structured as follows: after introducing the selected security and compliance properties using a case study (Section 2), we present SecureBPMN in Section 3. Thereafter, we discuss our design choices in developing an BPMN tool chain supporting SecureBPMN as well as certain aspects of modelling secure business processes (Section 4). Finally, we discuss related work and draw conclusions in Section 5.

2 Security in Business Processes

Enterprises require often a multitude of security or compliance requirements that cannot be expressed directly in standard process or work flow modelling languages such as BPMN [21] or BPEL [20]. For example, Figure 1 illustrates a simple *travel approval* process: an employee can request a business trip (Request Travel) which needs to be approved both with respect to the actual absence from the office (Approve Absence) as well a with respect to the costs (Approve Budget). If both approvals are given, an external service company that assists the travelling employee, e. g., in case of an emergency, is informed (Contact Travel Service Company) about the business trip. Finally, a notification, if he or she is allowed to travel or not, is send to the requesting employee (Inform Requester).

This, relatively simple, process requires already a surprising large number of security and compliance requirements; for example:

- While every staff member is allowed to request travels, only a restricted set of persons should be able to approve a travel. In more detail, cost centre managers should only be able to approve travels that are charged on their cost centre. Similarly, project or line managers should only be able to approve the absence of their subordinates. Thus, already this simple scenario requires a *fine-grained access control* system that cannot be modelled using a simple role-based access control model.
- To avoid fraud, the same person should not be allowed to approve both the absence and the costs of a travel. Of course, such a strict application of the *separation of duty* principle may hinder regular business operations. Thus, a more fine-grained variant, e. g., travels that cost more than 500 Euro must be approved by two different subjects, is usually required. Thus, separation of duty (as well as complementary binding of duty) should restrict permissions and not whole tasks (actions) or a business process.
- If the travel request is approved, an external service provider (e. g., a travel agency for booking accommodations or company that assists travellers in emergencies) is contacted. While these companies need to know some details of the business trip (e. g., the date of travel and the destination), they are not allowed to learn confidential details such as the business reason of the trip. Applying the principle of *need to know* or *least privilege*, can ensure such strict confidentiality requirements.
- Applying the discussed security and compliance requirements strictly may harm the business, e. g., if travel requests are blocked due to a manager being on leave. Thus, a controlled way for transferring rights, i. e., *delegation*, is essential. To ensure that a delegation of tasks does not violate

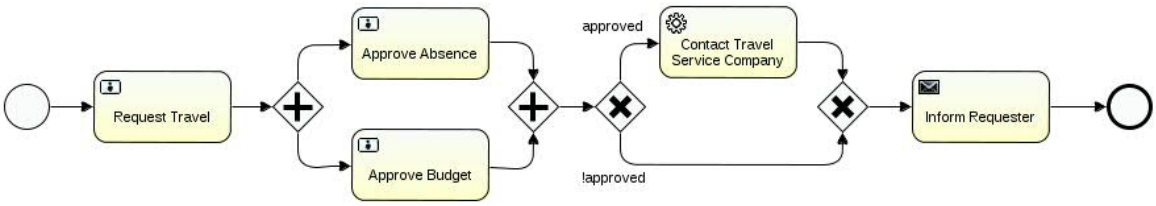

Figure 1: A simple business process for requesting and approving travel requests.

more important compliance rules, we also need to be able to specify the restrictions on delegations (e. g., certain tasks might not be delegatable at all or only delegatable to persons that already posses the necessary access rights).
While often related to health-care scenarios, studies, e. g., Bartsch [3], show that even more generic solutions for exceptional access control, such as break-glass, are needed by small and mid-size enterprises for ensuring that security enforcement does not prevent legitimate business transactions.

Even this simple scenarios already shows that describing the non-functional security and compliance requirements is a significant part of the overall business process design. In real-world scenarios, the effort for specifying and implementing the non-functional requirements can easily outgrow the effort for specifying and implementing the functional requirements.

3 SecureBPMN

Security and compliance should be modelled together with the business processes, instead of addressing them as an after-thought. We address this need for an integrated modelling language with SecureBPMN, a metamodel-based (Brucker and Doser [8] discus the details of metamodel-based language extensions) security language.

SecureBPMN can easily be integrated into business process modelling languages or work flow modelling languages. Figure 2 shows the (slightly simplified) metamodel of SecureBPMN and its exemplary integration into BPMN. SecureBPMN allows to describe the following security and compliance requirements:

- *Access Control:* The core of SecureBPMN is a hierarchical role-based (RBAC) [2] access control language supporting arbitrary constraints (AuthorizationConstraint) on the permissions. The constraints can, e. g., be used for expressing requirements like "managers can approve travels only if the requester is a subordinate of the manager." A Subject, in SecureBPMN, can be an individual User or a Group of subjects. Subjects are mapped to a Role hierarchy. SecureBPMN supports to explicitly permit (Permission) the actions (Action) on resources (Resource). In case of BPMN, resources are instances of the BPMN meta-classes Process, Activity, or ItemAwareElement. This part of SecureBPMN is, conceptually, very close to SecureUML [7].
- *Delegation:* SecureBPMN supports delegation with (TransferDelegation) and without (SimpleDelegation) transferring *all* (including access to data or back-end systems) access rights that are necessary to execute a task. The former only allows to delegate tasks to subjects that already possess the necessary rights. The latter allows to delegate tasks to arbitrary subjects that, then, can act on behalf of the original subject (Delegator). The number of delegations can be restricted by maxDepth, e. g., a maxDepth of zero forbids any delegation explicitly and value of one forbids a delegatee to delegate a task further. A delegation can be negotiable, i. e., the delegatee can refuse to do a delegated task. If a delegation is not negotiable, it is an order and the delegate has to do this task. We only need to model the delegator, as the delegatee is uniquely determined by the user that a task is assigned to.
- *Permission-level separation and binding of duty:* SecureBPMN models separation of duty (SoD) and binding of duty (BoD) as a sub-type of AuthorizationConstraint. In contrast to existing works, which constrains all actions on a task or service, this results in a fine-grained notion of these properties on the level of single permissions. Moreover, SecureBPMN generalises the, usually binary, SoD and BoD constraints to n-ary constraints: an SoD constraints models, that a Subject is not allowed to "use" more than max permissions out of n ($\mathsf{max} < n$); BoD is generalised similarly. If a SoD (BoD) constraint is already guaranteed by the RBAC configuration, it is called static SoD (BoD). Additionally, SecureBPMN supports history-resets (TriggerReset) for SoD (BoD) for processes with loops (similar to the work of Basin et al. [6]). Such resets allow to model that a SoD (BoD) constraints only needs to hold for the last (successful) execution of a loop and, thus, avoids the risk of successively "consuming" all subjects and, eventually, resulting in a dead lock.
- *Need-to-know principle:* Confidentiality or a

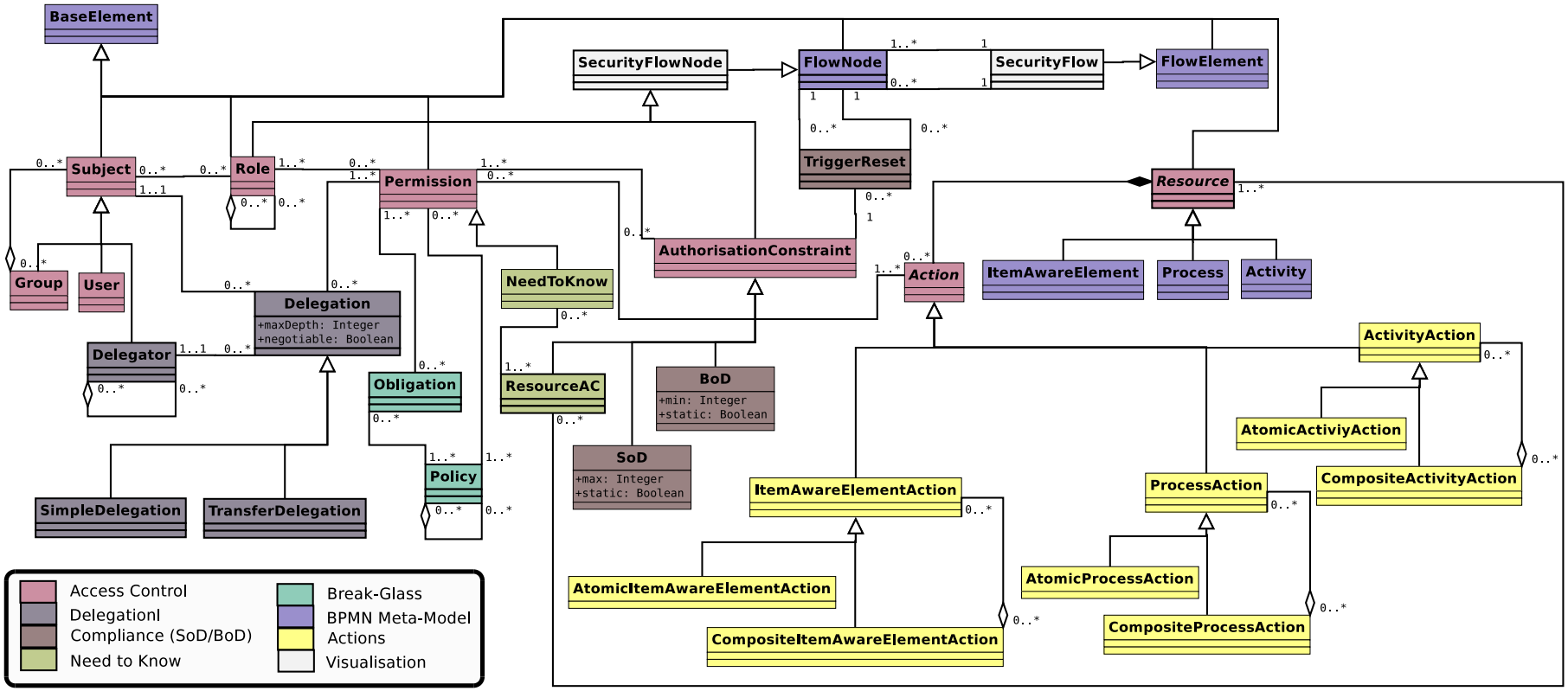

Figure 2: The SecureBPMN metamodel (simplified) and its connection to the BPMN metamodel

strict application of the need-to-know principle (NeedToKnow) is another important security property. In the context of business process-driven systems, this mainly refers to restriction the access to process variables or data objects (instances of the BPMN meta-class ItemAwareElement) and, thus, the process model internal data-flow. To allow the fine-grained restriction to access of certain resources (e. g., access to the travel details is not allowed, if the travel takes longer than 14 days), we model the need to know principle as specialised Permission that is associated with a specific authorisation constraint (ResourceAC).

- *Exceptional access control:* The strict enforcement of security and compliance requirements always bears the risk of hindering legitimate business transactions. Thus, an increasing number of enterprises implements break-glass or exceptional access control mechanism that allows regular users to override access control decisions in a controlled manner, e. g., adhering to certain obligations (Obligation) that are either defined on the permission or policy level. SecureBPMN supports such a mechanism using a hierarchy of security policies (defined by the meta-class Policy), i. e., implementing the approach presented in [10].

Conceptually, the integration of the SecureBPMN metamodel into the metamodel of BPMN is straight forward: BPMN defines the resources and actions that are constrained by the SecureBPMN language. On a technical level, the wish for diagrammatically representations of parts of the language as well as the fact that we can a metamodel only by subclassing (and not by introducing new superclasses) creates additional complexity:

- The SecureBPMN metamodel contains classes (SecurityFlowNode and SecurityFlow) that are not necessary for modelling security and compliance requirements. Their sole purpose is to provide a diagrammatic specification of certain requirements, e. g., SoD.
- As conceptually, we only would like to specify a common hierarchy of actions, this is technically impossible. To integrate SecureBPMN into BPMN, we need to define this hierarchy for each resource of BPMN (e. g., Activity) separately.

These parts of the metamodel (see Figure 2) are specific to BPMN and not part of the core of SecureBPMN.

4 Discussion and Future Work

The challenges raised in this section as well as the suggestions for further work are based on discussions with product groups of SAP AG and our own experience in applying SecureBPMN in several case studies in the domains e-government, air traffic management, and telecommunication services.

4.1 Security and Compliance Properties

The selection of security and compliance properties supported by SecureBPMN is based on discussion with various experts at SAP AG as well as our own case studies. In our experience, these properties cover the most important needs of business experts and, moreover, they can be expressed on the process level. Of course, there is a plethora of equally important security requirements (e. g., different types of encryption, authentication) that need to be considered as well. Still, these properties are usually on a technical level and, thus, need to be defined during the implementation of a secure business process. Nevertheless, the integration of technical properties into business process is an interesting line of future work.

4.2 Visualising Security Properties

One important property of BPMN is its support for describing business processes in a diagrammatic way that supports both the business experts as well as the system experts. Consequently, when extending such a language with a domain specific language for modelling security and compliance properties, it is tempting to provide visual representations for those properties as well. Figure 3 shows the user interface of our SecureBPMN modelling environment (which is based on Activiti BPMN Platform) in which we implemented a visual notation for SoD and BoD constraints (centre of the window). Applying this to larger case studies resulted quickly in over-populated diagrams that neither helped the business expert nor the security expert. Thus, we refrained from this approach and implemented dedicated property panes (lower part of the window). While such dedicated user interfaces provide the necessary tools for power users (i. e., security experts), they are not the best choices for increasing the awareness of business experts for security and compliance requirements. Thus, we still consider the question of finding a good (visual) representation of security and compliance requirements that can easily understood by business experts, system experts, and security experts to be open.

4.3 Diagrams vs. Models

Many users of visual modelling languages identify the models with their visual representation, e. g., the business process diagram. This misconception is, sadly, also perceptible in most business process modelling tools: these tools present the process diagram in the centre of their user interface (see Figure 3 for an example) and provide no access to the underlying model. We argue, that a model is something much more fundamental than a diagram, i. e., a diagram is only selected view on the model. Thus, often several diagrams, each of them visualising different aspects of a model, are necessary to capture the actual model. While this need for different

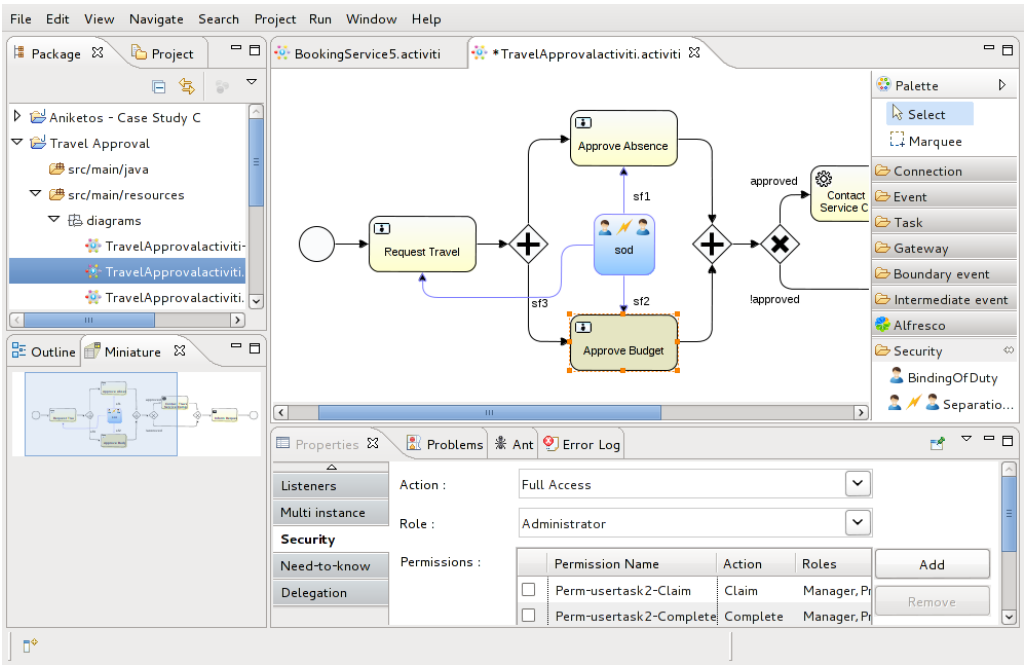

Figure 3: Specifying security requirements diagrammatically as well as using specialised user interfaces.

views (including, e. g., an abstract, tree-like view of all model elements and their properties) is already prevalent for modelling the functional aspects of a business process, it becomes inevitable when non-functional aspects such as security, compliance, or performance are added. Moreover, separating the model from its (visual) representation should also avoid the need for adding meta-classes purely for providing a visualisation (e. g., SecurityFlowNode and SecurityFlow in Figure 2).

4.3.1 Consistency Analysis

Modelling non-functional requirements of business processes increases the chance of conflicting requirements and, thus, the need for an design-time analysis. Besides well-known analyses like van der Aalst et al. [24] (which need to be extended to include non-functional requirements), specific checks for the security and compliance models are necessary, e. g., for

1. checking the internal consistency of the security specification, e. g., to ensure that the access control requirements and need-to-know requirements do not contradict each other.
2. checking the information (data) flow on the process level, e. g., to infer or check high-level need-to-know requirements.
3. checking that process-level security requirements are fulfilled on the implementation and configuration level. This is particular important for implementation and configuration artefacts that are not generated in a model-driven approach.
4. checking that the business processes are executable if the security requirements are enforced, i. e., there exists a valid execution trace from a start to an end event.
5. analysing different implementation techniques (e. g., resulting in different costs or runtime resource requirements) of security requirements.

4.3.2 Runtime Enforcement

While not the main scope of this paper, we want to mention that modelling security and compliance requirements can only be the start: these requirements need to be fulfilled at runtime, i. e., while executing the business processes in a business process execution engine. For example, in our prototype [12] we generate XACML [19] policies from the SecureBPMN models. An extended version of the Activiti BPMN runtime uses the generated XACML policies for enforce the access control, SoD/BoD, and the delegation requirements at runtime.

5 Conclusion and Related Work

We presented SecureBPMN, a security and compliance modelling language. By integrating SecureBPMN into business process modelling languages, SecureBPMN allows for modelling high-level security and compliance requirements together with the functional business requirements. SecureBPMN is supported by a BPMN modelling and execution framework [9, 12] that, in addition to the modelling and secure execution (i. e., including runtime enforcement of security and compliance policies) of business processes, supports the analysis of the consistency and correctness of the implementation.

There is a large body of literature extending graphical modelling languages with means for specifying security or privacy requirements. One of the first approaches is SecureUML [17], which is conceptually very close to the access control part of our BPMN extension. SecureUML is a metamodel based extension of UML that allows for specifying RBAC-requirements for UML class models and state charts. There are also various techniques for analysing SecureUML models, e. g., Basin et al. [5] or Brucker et al. [11]. While based on the same motivation, UMLsec [15] is not defined using a metamodel. Instead, the security specifications are written, in an ad-hoc manner, in UML profiles. In contrast, integrating security properties into business processes is a quite recent development, e. g., motivated by the work of Wolter and Schaad [25]. In the same year, Rodríguez et al. [23] presented a metamodel based approach introduction a secure business process type supporting global security goals. In contrast, our approach allows the fine-grained specification of security requirements for single tasks or data objects. Similar to UMLsec, Mülle et al. [18] present an attribute-based approach (i. e., the conceptual equivalent of UML profiles) of specifying security constraints in BPMN models.

Besides the modelling of (rather technical) security and compliance requirements, integrating risk and attack models into business processes is important line or research. For example, Altuhhova et al. [1] present an integration of the information security risk management model into BPMN. In what sense, such risk modelling and security requirement approaches can be combined, is still an open questions. For example, one could try to use SecureBPMN for describing countermeasures for the risks and treats expressed in the information security risk management model.

Acknowledgement. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no. 257930 (`http://www.aniketos.eu/`).

References

[1] O. Altuhhova, R. Matulevicius, and N. Ahmed. Towards definition of secure business processes. In M. Bajec and J. Eder, editors, *CAiSE Workshops*, *LNBIP 112*, pages 1–15. Springer, 2012.

[2] *American National Standard for Information Technology – Role Based Access Control*. 2004. ANSI INCITS 359-2004.

[3] S. Bartsch. A calculus for the qualitative risk assessment of policy override authorization. In *Security of information and networks (SIN)*, pages 62–70. ACM Press, 2010.

[4] Basel Committee on Banking Supervision. Basel III: A global regulatory framework for more resilient banks and banking systems. Technical report, Bank for International Settlements, Basel, Switzerland, 2010.

[5] D. Basin, M. Clavel, J. Doser, and M. Egea. Automated analysis of security-design models. *Information and Software Technology*, 51(5):815–831, 2009.

[6] D. Basin, S. J. Burri, and G. Karjoth. Separation of duties as a service. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 423–429. ACM Press, 2011.

[7] D. A. Basin, J. Doser, and T. Lodderstedt. Model driven security: From UML models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology*, 15(1):39–91, 2006.

[8] A. D. Brucker and J. Doser. Metamodel-based UML notations for domain-specific languages. In J. M. Favre, D. Gasevic, R. Lämmel, and A. Winter, editors, *Software Language Engineering (ATEM)*. Oct. 2007.

[9] A. D. Brucker and I. Hang. Secure and compliant implementation of business process-driven systems. In M. L. Rosa and P. Soffer, editors, *Security in Business Processes (SBP)*, *LNBIP 132*, pages 662–674. Springer, 2012.

[10] A. D. Brucker and H. Petritsch. Extending access control models with break-glass. In B. Carminati and J. Joshi, editors, *ACM SACMAT*, pages 197–206. ACM Press, 2009.

[11] A. D. Brucker, J. Doser, and B. Wolff. A model transformation semantics and analysis methodology for SecureUML. In O. Nierstrasz, J. Whittle, D. Harel, and G. Reggio, editors, *Model Driven Engineering Languages and Systems (MoDELS)*, *LNCS 4199*, pages 306–320. Springer, 2006.

[12] A. D. Brucker, I. Hang, G. Lückemeyer, and R. Ruparel. SecureBPMN: Modeling and enforcing access control requirements in business processes. In *ACM SACMAT*, pages 123–126. ACM Press, 2012.

[13] C. Fox and P. Zonneveld. *IT Control Objectives for Sarbanes-Oxley: The Role of IT in the Design and Implementation of Internal Control Over Financial Reporting*. IT Governance Institute, Sept. 2006.

[14] HIPAA. Health Insurance Portability and Accountability Act of 1996. `http://www.cms.hhs.gov/HIPAAGenInfo/`, 1996.

[15] J. Jürjens and R. Rumm. Model-based security analysis of the german health card architecture. *Methods Inf Med*, 47 (5):409–416, 2008.

[16] V. Kapsalis, L. Hadellis, D. Karelis, and S. Koubias. A dynamic context-aware access control architecture for e-services. *Computers & Security*, 25(7):507–521, 2006.

[17] T. Lodderstedt, D. A. Basin, and J. Doser. SecureUML: a UML-based modeling language for model-driven security. In J.-M. Jézéquel, H. Hussmann, and S. Cook, editors, *UML*, *LNCS 2460*, pages 426–441. Springer, 2002.

[18] J. Mülle, S. von Stackelberg, and K. Böhm. A security language for BPMN process models. Technical report, University Karlsruhe (KIT), 2011.

[19] OASIS. eXtensible Access Control Markup Language (XACML), version 2.0, 2005.

[20] OASIS. Web services business process execution language (BPEL), version 2.0, Apr. 2007.

[21] Object Management Group. Business process model and notation (BPMN), version 2.0, Jan. 2011.

[22] P. E. Proctor and N. MacDonald. Marketscope for segregation of duty controls within ERP and financial applications. ID Number G00161625, Gartner Research, Sept.25 2008.

[23] A. Rodríguez, E. Fernández-Medina, and M. Piattini. A BPMN extension for the modeling of security requirements in business processes. *IEICE - Trans. Inf. Syst.*, E90-D:745–752, March 2007.

[24] W. M. P. van der Aalst, M. Dumas, F. Gottschalk, A. H. M. ter Hofstede, M. L. Rosa, and J. Mendling. Correctness-preserving configuration of business process models. In J. L. Fiadeiro and P. Inverardi, editors, *FASE*, *LNCS 4961*, pages 46–61. Springer, 2008.

[25] C. Wolter and A. Schaad. Modeling of task-based authorization constraints in BPMN. In G. Alonso, P. Dadam, and M. Rosemann, editors, *BPM*, *LNCS 4714*, pages 64–79. Springer, 2007.