EVALUATING A UML-BASED MODELING FRAMEWORK FOR PROCESS-RELATED SECURITY PROPERTIES: A QUALITATIVE MULTI-METHOD STUDY

Schefer-Wenzl, Sigrid, WU Vienna, Institute for Information Systems and New Media, 1090 Vienna, Austria, sigrid.schefer-wenzl@wu.ac.at

Sobernig, Stefan, WU Vienna, Institute for Information Systems and New Media, 1090 Vienna, Austria, stefan.sobernig@wu.ac.at

Strembeck, Mark, WU Vienna, Institute for Information Systems and New Media, 1090 Vienna, Austria, mark.strembeck@wu.ac.at

Abstract

In recent years, we developed a modeling framework for process-related security properties, the BusinessActivities Framework. This paper reports on a long-term empirical study to evaluate the applicability of four UML extensions included in the BusinessActivities Framework. We used an exploratory research design employing four interpretative case studies followed by three semi-structured interviews based on 30 real-world business processes from a large Austrian school center. The case work resulted in 23 process models. By assessing the model complexity quantitatively and by interpreting the case as well as the interview material, we found that modelers are predominantly affected by the upfront effort of establishing a conceptual background on process-related security concepts and by the semantic complexity of control-flow modeling in UML activity diagrams. Non-technical domain experts considered the visual process models as suitable communication instruments. The findings demonstrate the potential value of applying our modeling framework in a practitioner's setting.

Keywords: Process security, UML, Design research, Evaluation.

1 Introduction

Business processes specify how an organization's resources are used to achieve certain predefined business goals. Consequently, their correct execution is of major importance for the respective organizations. Although the information systems that support the execution of business processes are exposed to various kinds of security threats, process models, information systems, and corresponding security policies are usually defined separately (see, e.g., zur Muehlen et al., 2008). However, to enable the secure execution of business processes and to enforce security policies in information systems, process-related security aspects must be already considered in the early stages of business process design and throughout the entire software development lifecycle (see, e.g., Mouratidis and Jürjens, 2010).

In this context, we developed the BusinessActivities Framework that aims at supporting the specification, the implementation, and the enforcement of process-related security properties at various stages of the security-engineering process (see Section 2). The BusinessActivities Framework includes computation-independent models (CIM) for process-related security properties as well as corresponding platform independent (PIM) and platform-specific models (PSM). At the PIM level, we provide a number of UML extensions that extend the UML (OMG, 2011) with modeling primitives for process-related security properties. Our research on the BusinessActivities Framework follows a design science research approach (see, e.g., Hevner et al., 2004, Peffers et al., 2007). For this reason, our research includes the collection of empirical observations to establish how our design artifacts (i.e., the tooling and the corresponding engineering methods) can be improved (cf. Hevner et al., 2004).

In this paper, we focus on the evaluation of UML extensions which allow the integrated modeling of business processes and related Role-Based Access Control (RBAC) concepts. In recent years, RBAC (see, e.g., Ferraiolo et al., 2007) has developed into a de facto standard for access control. In RBAC, roles are used to model different job positions and responsibilities within an organization and/or information system. Permissions are assigned to roles according to the tasks each role has to accomplish. The roles are then assigned to human users according to their respective work profile. Roles are also used as an abstract concept for delegation (see, e.g., Crampton and Khambhammettu, 2008) or for the assignment of duties defined via obligations (see, e.g., Zhao et al., 2007). The need for integrated modeling of business processes and related access control concepts has been repeatedly identified in research and practice (see, e.g., List and Korherr, 2006).

Currently, empirical evidence (e.g., well-documented industrial case studies) on the suitability of model-driven security engineering artifacts is largely missing. Many related reports are based on small, fictitious examples for explaining certain approaches or concepts. Moreover, most often the related work does not provide a detailed description of the respective research process (see Section 5). Thus, to evaluate our modeling artifacts we first had to establish which factors actually contribute to the applicability of our model-driven engineering framework. In this paper, we describe a qualitative multi-method study to evaluate four of the UML extensions that are part of the BusinessActivities Framework. In particular, we conducted a series of case studies and interviews. The case studies are based on a collection of real-world business processes provided by a large Austrian school center. The interviews were conducted subsequent to the case studies to further evaluate the modeling artifacts produced during the case studies. The whole research process reported in this paper took about one year. In this period of time, the case studies (60 working days) and the interviews were conducted. Our evaluation was guided by the following two exploratory research questions:

RQ1: Which are the barriers to adopting our UML extensions by domain modelers having a basic background in UML activity modeling?

RQ2: Which are the barriers to using the process models based on our UML extensions for non-technical, non-security stakeholders in modeled organizations?

The remainder of this paper is structured as follows. Section 2 briefly describes the research object of our qualitative research: the UML extensions included in the BusinessActivities Framework. Subsequently, Section 3 details our multi-method research design as well as the data collection/analysis of our case studies and semi-structured interviews, respectively. Section 4 discusses the observations from the study and reviews these observations in the light of the two guiding research questions. In Section 5, we elaborate on the limitations of our approach and of the resulting findings. Section 6, presents related work and Section 7 concludes the paper.

2 Overview of the UML Extensions for BusinessActivities

In this section, we shortly summarize the modeling artifacts evaluated in this paper. In (Strembeck and Mendling, 2011), we present an integrated approach for modeling processes and process-related RBAC models. Based on a formal metamodel (at the CIM level) for process-related RBAC models, we define a domain-specific extension for UML activity diagrams (at the PIM level). In addition, the BusinessActivites framework includes extensions for several other RBAC-related security properties. In this paper, we evaluate four of these extensions:

(1) The *Duty extension* (Schefer and Strembeck, 2011a, Schefer, 2011) enables the integrated modeling of duties defined in obligation policies and process-related RBAC models. Thereby, we support process modelers when defining tasks, which require the fulfilment of certain duties when these tasks are executed in order to comply with certain laws and regulations.

(2) The *Delegation extension* (Schefer and Strembeck, 2011b) provides modeling support for the delegation of roles, tasks, and duties in the context of process-related RBAC models. Delegation is an important concept to increase flexibility in authorization and obligation management. However, due to the complex interplay of delegation assignments with other process- and RBAC-related concepts, we propose explicit delegation abstractions for integrated RBAC process models.

(3) The *Context Constraint extension* (Schefer-Wenzl and Strembeck, 2012a) enables the definition of context-aware RBAC models for business processes. In an IT-supported workflow, process-related context constraints can be defined as means to consider context information in access-control decisions. A context constraint specifies that certain conditions must be fulfilled to permit the execution of a particular task.

(4) The *Break-Glass extension* (Schefer-Wenzl and Strembeck, 2012b) supports the modeling of process-related break-glass policies. In emergency situations, certain subjects sometimes have to perform important tasks although they are usually not authorized to perform these tasks. Break-glass policies have been introduced as a sophisticated exception handling mechanism to resolve such situations. They enable selected subjects to break or override the standard access control policies of an information system in a controlled manner.

Figure 1a shows a simplified medical examination process modeled as a UML BusinessActivity. The example process uses some of the new modeling elements introduced in the BusinessActivity extension (Strembeck and Mendling, 2011). The process from Figure 1a starts when a patient arrives at the hospital. The process includes four so-called BusinessActions. BusinessActions are special purpose tasks that can be linked to, e.g., roles and constraints. A BusinessAction is depicted as a UML2 Action symbol (a round-cornered rectangle) that includes the letter “B” in a compartment in the upper right corner. The process from Figure 1a also visualizes task-based entailment constraints, which place some restriction on the subjects who are allowed (or required) to execute a particular task. For example, a subject-binding constraint defines that two bound tasks must be performed by the same individual within the same process instance. In the example process from Figure 1a, a subject-binding constraint is defined between the tasks t_1 and t_2 to ensure that the same physician who performed the examination in the "Medical examination" task also evaluates appropriate medical treatment options. This subject binding is indicated via "SBind" entries in the corresponding task symbol. Figure 1b illustrates corresponding task-to-role, role-to-subject, and role-to-role assignments. For example,

subject S_1 is assigned to the Junior Physician role. All members of the Junior Physician role are permitted to perform the "Medical examination", "Determine treatment options", and the "Medical treatment" tasks.

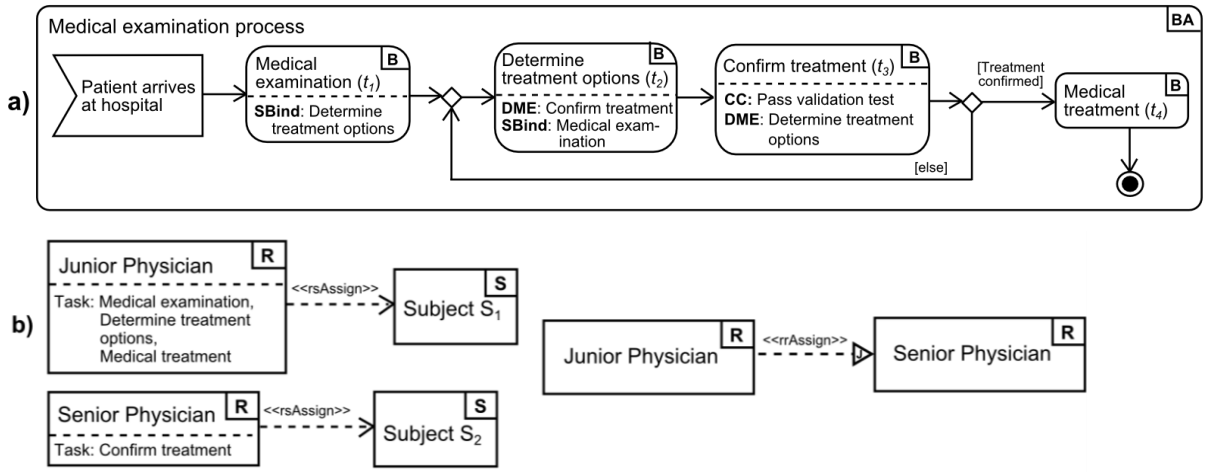

Figure 1. Simplified medical examination process

3 Research Design

The two guiding research questions (RQ1, RQ2) established different requirements on our research design. First, each research question targets a distinct group of subjects: domain modelers (technical experts) applying the UML extensions as well as non-technical domain experts using the resulting diagrams, respectively. Second, as RQ1 was to be explored in a real-world modeling situation, a substantial upfront effort was required to complete a set of non-trivial modeling tasks to produce models which are then to be evaluated with regard to RQ2. Addressing both research questions in a single research step was therefore discarded. Third, the modeling artifacts resulting from investigating RQ1 would be available to extract basic quantitative data about certain internal attributes of the extended UML models (e.g., model sizes and structuredness). This quantitative data would allow for contextualizing observations with respect to RQ1 and RQ2, for instance, by reflecting on the model complexity.

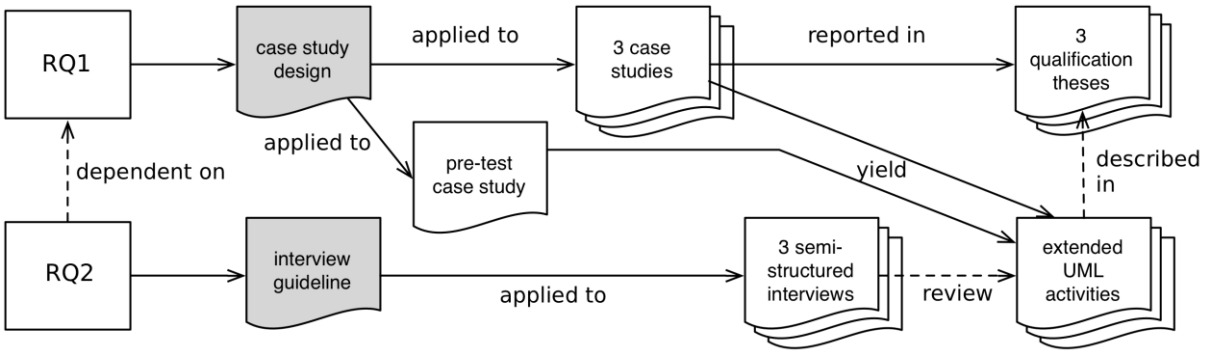

Figure 2. A multi-method research design

Therefore, we adopted a sequential multi-method research design (see Tashakkori and Teddlie, 2010) with two subsequent research phases and two different research instruments (see Figure 2). As for RQ1, we designed interpretative case studies (see Klein and Myers, 1999) because we wanted to address RQ1 using non-trivial process engineering tasks. RQ2 would then be covered by subsequent semi-structured interviews (see Hove and Anda, 2005) which would allow us to collect data

concerning the communicability as perceived by important stakeholders. In addition, the interviews would permit clarifying critical model details for the respondents to improve the quality of the answers.

3.1 Case Study Design

To address the first research question (RQ1), we designed a series of case studies, with each case study focusing on one particular UML extension (see Section 2) in isolation. In the following, we summarize the case study objective, the real-world setting to be studied (the case), the frame of conceptual reference (theory), and the details of data collection (methods; see Robson, 2011, Runeson and Höst, 2009):

Case selection: The case setting was provided by a large Austrian school center offering different education levels ranging from elementary to advanced levels. The primary case artifacts were the textual descriptions of about 30 organizational processes collected by members of the school during a process management initiative. The control flow of some processes was graphically visualized depicting the sequence of tasks as well as of corresponding authorized and responsible persons. However, these processes were visualized using an ad hoc graphical notation. Furthermore, most of the processes were described in a detailed textual/tabular listing of activities with varying levels of granularity. These process descriptions included references to legal requirements (e.g., paragraphs in the Austrian law concerning teaching in schools) as well as to other internal or external regulatory documents (e.g., recommendations and guidelines of the Austrian Department of Education). In many organizations, such data are usually confidential. The school center, however, offered us detailed process descriptions including additional information on authorized and responsible persons for each task. At the same time, the school center allowed us to publish our results in anonymized form. Moreover, administrative and faculty members of the school were available to collect further feedback. In each case study, a subset of these processes was modeled by applying one of our UML extensions.

Objective: The case studies were designed to identify possible barriers that users of our extensions might experience. The practical objective was to construct a collection of extended UML activities to visualize organizational processes in the above case. The references to and process details extracted from the regulatory documents were to be translated into explicit model elements that are provided by our UML extensions.

Conceptual framing: In Strembeck and Mendling (2011), we provide the key vocabulary and the abstracted concepts to consider RBAC engineering artifacts (roles, tasks, subject-role assignments, etc.) in a business process context (see Section 2). This includes a formal metamodel for integrating RBAC artifacts into business processes. Each case study was designed to cover one particular UML extension.

Methods: The first case study evaluates the Break-Glass extension presented in Schefer-Wenzl and Strembeck (2012b) and was performed by the authors of this paper for pre-test purposes. We tested the whole data collection and data analysis process as described below. The results of this case study are included in this paper for comparison purposes. For conducting the remaining three case studies, we recruited one graduate and two undergraduate students of WU Vienna as case subjects to perform the case study work as part of their qualification theses, i.e., Bachelor or Diploma theses. The authors of the UML extensions acted as supervisors for the corresponding three case studies. To control the important confounding factor of experience, we focused on the experience with a mainstream modeling language (UML) and experience from education (in particular regarding RBAC concepts). We required the students to have successfully passed a course teaching the basics of the UML and a second course providing an introduction to computer security topics (including RBAC). Student subjects with an assessable level of educational experience provided a practical operationalisation of a

basic UML background, as required by RQ1. Due to the time-unbound nature of the case studies and resource constraints, we did not consider modeling professionals as research subjects (see Section 4).

For data collection, the students had to perform an in-depth analysis of the documents describing the school's processes as well as an analysis of the respective internal and external regulations. Moreover, in case of ambiguities, uncertainty, or insufficient documentation, we requested additional information and feedback from the school's staff. Given that certain process models did not contain any security details specific to a given UML extension, or relevant materials for a detailed security modeling step were missing, the process models were excluded from the corresponding case study. In each case study, the following steps were performed:

(1) The respective student got acquainted with one of the UML extensions presented in Section 2.2 based on the research material and a tutorial by the authors of the extension.

(2) Next, the student analyzed the process model collection and identified processes to be modeled via the modeling extension.

(3) The student modeled the processes as UML activities applying the respective UML extension.

(4) In repeated feedback rounds, the supervisor checked the syntactical correctness of the process models and discussed possible improvements with the student.

(5) The student provided revised versions of the process models.

(6) After the supervisor approved the process models, selected administrative and faculty members of the school answered questions to provide additional information, clarify uncertainties, and to confirm the correctness of the extended process diagrams modeled in the case study.

(7) The student produced final model revisions.

3.2 Interviews

After all case studies were finished, the authors of this paper evaluated the results of all case studies by performing semi-structured interviews with three members of the school, including the head master, one teacher, and one member of the administrative staff. This approach was chosen because interviews are one of the most important methods supporting case study research (see, e.g., Runeson and Höst, 2009). For qualitative case studies it is recommended to choose subjects from different parts of the organization to involve different roles in the interviews. The interview was carefully designed using the guidelines from Hove and Anda (2005). It consisted of five main open-ended questions. Each interview varied between 20 and 25 minutes in length. The answers were recorded by using field notes which were then subsequently analyzed by the interviewer considering RQ2. Table 1 details the main questions asked in the interviews. The results of the interviews will be discussed in Section 4.

Q1	Do the process models provide added value for the school? If yes, in how far can the school/members of the school benefit from the extended process models?
Q2	How will the extended process models potentially be used in the school?
Q3	What do you think about our approach of integrating process models and related security aspects? Advantages/Disadvantages?
Q4	Do you have difficulties in understanding different parts of the processes? If yes, which parts are easy to understand and which parts are difficult or not comprehensible?
Q5	Do you have any suggestions on how the graphical representation of the processes can be improved?

Table 1. Questions from semi-structured interviews

4 Results

We completed four case studies, each focusing on one specific UML extension (duties, delegation, context constraints, and break-glass policies, see Section 2). Overall, 23 processes were modeled in the four case studies. Each of the case studies modeled a subset of the 30 processes from the process collection. These subsets are overlapping, because seven of the 23 modeled processes are included in more than one case study. One process was even selected in all four case studies. Each modeler chose the subset of processes for his/her case study depending on their suitability to be modeled via the corresponding UML extension (see Section 3.1).

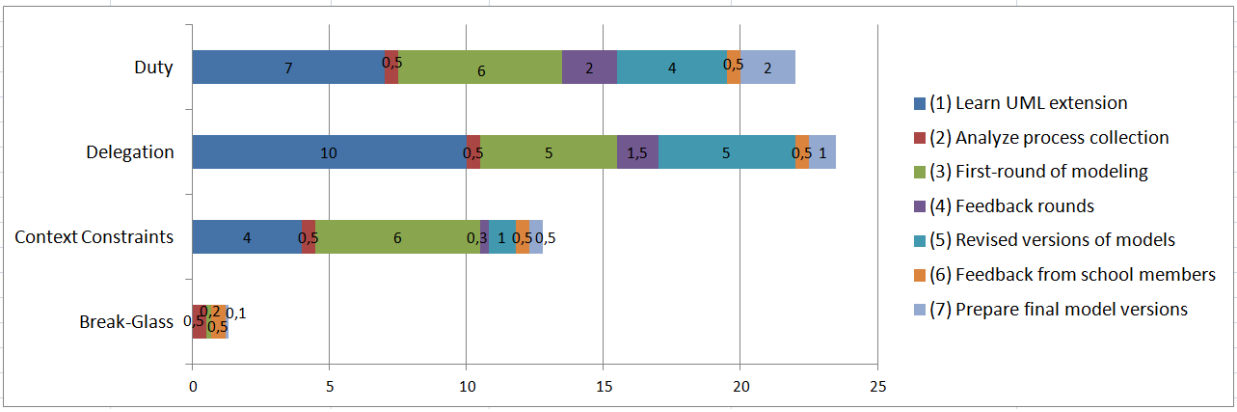

Figure 3. Case study length in days (based on 8-hours working days)

Figure 3 details the estimated periods of time for each phase in the case studies (see Section 3.1 for details on each phase). The case studies took between 1.3 days for the pre-test case study on the Break-Glass extension and 23.5 days for the Delegation extension. Due to the fact that the Break-Glass case study was performed by the authors of this UML extension, the resulting amount of time of about 1.3 working days approximates the minimum time effort needed for performing the case study, if the modeler was already familiar with the corresponding UML extension.

4.1 Observations

For each case study, Table 2 lists the number of processes modeled via the respective UML extension, the number of different symbols used for the processes in the case study, how many of these symbols were taken from the extended symbol set, as well as the number of revisions until the final version of the processes was approved.

Case Study	Number of modeled processes	Number of different symbols used	Number of new symbols used	Number of revisions
Duty Extension (Schefer and Strembeck, 2011a)	10	15	9	3
Delegation Extension (Schefer and Strembeck, 2011b, Schefer-Wenzl et al., 2012)	4	16	12	4
Context constraints Extension (Schefer-Wenzl and Strembeck, 2012a)	7	12	7	2
Break-glass Extension (Schefer-Wenzl and Strembeck, 2012b)	2	10	6	1

Table 2. Case study details

For example, row 1 in Table 2 shows that the case study on duties involved ten processes exhibiting requirements on an explicit duty management to meet certain legal requirements. Therefore, the 10 processes were modeled using the DutyNodes extension presented in Schefer and Strembeck (2011a). Throughout all processes modeled in this case study, 15 different UML symbols were used. Nine of these symbols were new symbols introduced by the DutyNodes UML extension. The last column in Table 2 indicates the number of revisions (3) that were necessary in order to produce the final version of the extended process models, which was approved by both the supervisor and the school center's headmaster.

4.2 Discussion

To create some background for interpreting the case and interview data in the light of the two research questions, we first comment on the characteristics of the resulting process models. For larger process models (i.e., of size greater than or equal to 50 nodes), it was empirically established that model understandability decreases and defect probability increases (Mendling et al., 2010). Therefore, the objective is to limit process models to as few elements as possible or to split larger processes into smaller parts (e.g., via structured activities in the UML). In our case studies, the number of tasks identified for each process description influenced the model size primarily. The maximum model size observed amounts to 50% of the empirical threshold established in Mendling et al. (2010), that is, a maximum of 25 nodes for the largest model in our case studies. Similarly, the routing paths per model element are comparatively small, with a low connector degree affecting model understandability and defect probability positively (Mendling et al., 2010). In fact, the typical node has the minimal degree of 2 (e.g., one incoming and one outgoing edge for a typical and minimally connected BusinessAction). This very low connector degree is partly explained by the comparatively low number of gates (i.e., a median gates count of 2), which have a connector degree higher than 2, used in the model. To sum up, the 23 process models can be considered of lower to medium complexity in terms of the guidelines established by Mendling et al. (2010). This finding is, in itself, encouraging given the complexity and heterogeneity of the input material (process descriptions, legislative text, regulatory documents) of the process engineering tasks.

Regarding *RQ1*, we found that the main challenge for the students was to understand the underlying security concepts to be modeled via the corresponding UML extension rather than applying the modeling extensions. For example, before being able to apply the BusinessActivityDelegation extension (Schefer and Strembeck, 2011b, Schefer-Wenzl et al., 2012), the respective student had to understand the concept of delegation roles in RBAC (see, e.g., Zhang et al., 2003). Besides this conceptual entry barrier, the students struggled with specifying valid process control flows in terms of standard UML activity semantics. A frequent issue was process flow locks due to unrealized implicit joins (Schattkowsky and Foster, 2007). As soon as these two barriers had been overcome, we found that the extension syntax was used correctly according to their semantics at the first time of usage. The later revisions between supervisors and students were primarily devoted to perfective refactorings of the process models. These involved the review and the removal of redundantly modeled tasks as well as the simplification of control flow structures (e.g., loops). An important refactoring task was the relabelling of several symbols. For example, we identified more significant labels for concrete roles and duties. Another focus was that Action labels were turned into a object-verb form (Schattkowsky and Foster, 2007). In personal communication, the three students concluded that the UML extensions were easy to learn when having a critical knowledge of the UML and were perceived as useful for modeling the corresponding security aspects.

In the semi-structured interviews on *RQ2* the 23 process models were presented to the headmaster, one teacher, and one member of the school's administrative staff. During the interviews, the five questions given in Table 1 were addressed. In particular, two advantages of the visually modeled processes were communicated to us: First, the headmaster emphasized that new employees who are not familiar with school procedures would now have a comprehensive and easy-to-understand, diagram-based documentation of key processes and related security concerns at hand. This would have the potential

of facilitating work tasks and communication with other school members during the first weeks after joining the school. This potential was mainly attributed to the fact that all processes are now documented in a consistent, integrated, homogeneous, and standardized way. This opinion may also support the frequently cited conjecture that models employing a process flow metaphor are suitable communication instruments for non-technical domain experts (see, e.g., Dumas et al., 2012). In addition, before our effort, only a few processes were depicted using an ad hoc (i.e., non-standard) visual notation. Most processes were described via textual documents in varying degrees of detail. The state of the organization's process descriptions was therefore inconsistent and inhomogeneous. Moreover, and second, the interview partners noted that the security-aware process models would improve the general awareness among the school members of how closely security requirements are related to key organizational processes. All three members of the school stated that the process models are easy to comprehend in their essence (e.g., task and role labels, basic sequencing of tasks, relations between duties and tasks). This could have been facilitated by the low to medium model complexity, as stated above.

5 Limitations and Threats

Case studies: The case study design was aligned to evaluating our modeling framework. As a consequence, the study design presents limitations to the generalizability of our findings. An important limitation results from the scope of a single organization for the four cases. The observations might therefore be specific to the domain of Austrian secondary schooling. However, within this domain, we aimed at a broad coverage of domain areas: the process models cover topics ranging from the school's process management to the emergency evacuation procedures. Nevertheless, future work must investigate whether the findings hold for different branches and different types of organizations. In addition, by limiting each case study and each student to applying a single UML extension, the findings do not reflect possible interactions between the extensions when used conjointly for a modeling task. For example, we might have missed positive effects (e.g., structuring, reuse of one extension's elements for another) or negative effects (e.g., contradicting modeling decisions and model defects) on modeling break-glass procedures, duties, and delegation in a single model. However, our research design was clearly confined to observing our UML extensions in isolation.

Besides, there is the risk that the extended time frames, which were available to the students to complete their modeling tasks, have introduced a learning effect. Any modeling task completed comparatively late during a case study might have been affected by the repeated revisions and feedback rounds between supervisors and students (see Table 2). To make the possible learning effects explicit, we documented the individual revisions and commented on change patterns in Section 4.2. However, granting these time frames allowed us to observe the effects of solving non-trivial modeling challenges using the UML extensions, which is otherwise impossible during a controlled experiment. Another personal bias could have been introduced by the authors of the UML extensions acting as the supervisors of the students working on the cases. The feedback rounds might have caused the supervisors to exert a critical influence on the decision-making process of the students who were applying the respective extension to solve a task (e.g., by communicating personal preferences towards certain modeling options). We tried to limit this influence by flipping the supervisor role for a given case study (and a given extension focus) between two extension authors.

Using graduate and undergraduate students as empirical subjects in the case studies cannot provide insights that can be generalized to professional practitioners facing comparable modeling tasks (see, e.g., Falessi et al., 2010). Recruiting students has the disadvantage of observing subjects having little experience both in process modeling (i.e., UML activity modeling) and in the domain modeled (e.g., legal framework for schooling processes). As a result, we observed erroneous control flows in the process models due to the students' unawareness of certain UML activity specificities. Most prominently, the hidden interaction between implicit joins performed by UML actions (see Schattkowsky and Foster, 2007) and decision nodes inserted prior to them was neglected repeatedly,

leading to deadlocks in the token flows. Even in the last revision, 10 out of the 23 models showed this defect. Note, however, that these defects result from the pitfalls of process modeling in general and cannot be related to the usage of our UML extensions.

Interviews: In likewise manner, there are threats to the observations from the three interviews. To begin with, they cannot be generalized beyond the narrow educational domain because the interview partners are all embedded into a single institution. There is also the risk of an interviewer bias because the interviewer is also author of the evaluated UML extensions. This double role might have affected the open-ended conversation of the interviews. To minimise this risk, the interviewer, however, tried to observe rather than steer the conversation and encouraged the interviewees to talk.

6 Related Work

While substantial empirical evidence on general process modeling has been reported (see Mendling et al., 2010) for an overview), more specific empirical studies in the field of process-related security engineering are rare; not to mention multimethod studies. In the few reports (Mouratidis and Jürjens, 2006, Mouratidis and Giorgini, 2007, Gao et al., 2004, Grünbauer et al., 2003, Schaad and Moffett, 2004, Giorgini et al., 2003) based on real-world cases, the underlying research designs are often not made explicit and the levels of descriptive detail spent on research methods vary significantly. Also, empirical instruments such as case studies are used in different design research phases. More importantly, however, the term "case study" is often used to denote mere examples of application for the documented modeling concepts, without a critical-analytical research objective (e.g., prototype improvement) and without guiding research questions (see. e.g., Runeson und Höst, 2009). A notable exception is a case study report by Accorsi and Stocker (2012). They describe a case design including phases and research questions on whether process-mining techniques as offered by ProM and conformance checking can be used to verify security properties (separation of duties, authorization) during security audits.

As for evaluating application cases, Gao et al. (2004) document an aspect-oriented approach for engineering access control in software systems by allowing for an extensible realization of the RBAC96 model in the UML and AspectJ aspects at the system level. To demonstrate the feasibility of their approach, they describe the case of introducing various RBAC levels into object-oriented middleware (CORBA Security). As in our setting, the case work is performed at the evaluation phase (i.e., the UML extension has been built). However, there are no case design and analysis applied beyond the exemplary CORBA application of the UML extension itself. Content-wise, our study covers advanced RBAC concerns (duties, context constraints, delegation). Similarly, Mouratidis and Giorgini (2007) present an application case from the health care domain (electronic Single Assessment Process, eSAP) to illustrate the agent-oriented Secure Tropos/i* modeling approach. In contrast to our approach, Secure Tropos/i* also covers the early software development phases (requirement analysis, architecture); and so does the case. However, there is no business process viewpoint with formally defined security-aware behavioral semantics. The UMLsec framework has been exemplified in two application cases: a biometric authentication protocol (Jürjens, 2005) and a point-of-sale transaction system (Mouratidis and Jürjens, 2006). UMLsec is centered on security properties of business data (confidentiality, integrity) rather than process security in more general. Both cases are not designed and reported as empirical case studies. Grünbauer et al. (2003) describe an internet banking case (authentication, confidentiality) of applying their CASE-based technique for modeling layered cryptographic protocols using state charts. Again, this application case is purely demonstrative and does not cover critical security concerns of process flows (RBAC, duties).

A second class of related work employs demonstrative cases in early design research, for identifying requirements of a model-driven security engineering approach, rather than to evaluate it. Giorgini et al. (2003) put forth a case on modeling the Secure Electronic Transactions (SET) procedures for credit cards. This case motivates the authors to add modeling support for task-goal-permission dependencies to Tropos/i*. Another motivating case in the banking domain is given by Schaad and Moffett (2004),

looking at the levels of organizational controls during credit application processes. These include separation of duties, delegation of obligations, and reviews, which are also considered in our modeling extensions. However, their Control Principle modeling framework does not integrate with a process flow viewpoint.

7 Conclusion

In this paper, we presented a multi-method study for the evaluation of different UML extensions for process-related security properties. The evaluation was conducted on a collection of real-world processes from a large Austrian school center. In particular, we systematically report on a set of case studies which were conducted in order to test the practical applicability of the respective UML extensions. Moreover, we conducted semi-structured interviews to further evaluate the artifacts that were produced in the case studies. The results from our multi-method study suggest that the UML extensions are suited for the application in real-world business settings. We also identified several points of improvement which will be considered in our future work. Moreover, our future work will include comprehensive studies to measure aspects such as the cognitive effectiveness of the UML extensions (see, e.g., Moody, 2009). In addition, we hope that our multi-method study inspires similar empirical research on other model-driven security engineering approaches.

References

Accorsi, R. & Stocker, T. 2012. On the Exploitation of Process Mining for Security Audits : The Conformance Checking Case. Symposium on Applied Computing (SAC).

Cramton, J. & Khambhammettu, H. 2008. Delegation and Satisfiability in Workflow Systems. SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies. Estes Park, CO, USA: ACM.

Dumas, M., Rosa, M. L., Mendling, J., Maesku, R., Reijers, H. A. & Semenenko, N. 2012. Understanding business process models : the costs and benefits of structuredness. Proc. of the 24th International Conference on Advanced Information Systems Engineering. Springer.

Falessi, D., Babar, M., Cantone, G. & Kruchten, P. 2010. Applying empirical software engineering to software architecture: challenges and lessons learned. Empirical Software Engineering, 15.

Ferraiolo, D. F., Kuhn, D. R. & Chandramouli, R. 2007. Role-Based Access Control, Artech House.

Gao, S., Deng, Y., Yu, H., He, X., Beznosovi, K. & Cooper, K. 2004. Applying Aspect-Orientation in Designing Security Systems: A Case Study. Proc. of 16th International Conference on Software Engineering and Knowledge Engineering (SEKE).

Giorgini, P., Massacci, F. & Mylopoulos, J. 2003. Requirement Engineering Meets Security: A Case Study on Modelling Secure Electronic Transactions by VISA and Mastercard. Conceptual Modeling - ER.

Grünbauer, J., Hollmann, H., Jürjens, J. & Wimmel, G. 2003. Modelling and Verification of Layered Security Protocols: A Bank Application. Proc. of the International Conference on Computer Safety, Reliability and Security (SAFECOMP).

Hevner, A. R., March, S. T., Park, J. & Ram, S. 2004. Design Science in Information Systems Research. Management Information Systems Quarterly, 28.

Hove, S. E. & Anda, B. 2005. Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research. Proc. of the 11th IEEE Int. Software Metrics Symposium.

Jürjens, J. 2005. Sound Methods and Effective Tools for Model-based Security Engineering with UML. Proc. of the 27th International Conference on Software Engineering (ICSE).

Klein, H. K. & Myers, M. D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Quarterly, 23.

List, B. & Korherr, B. 2006. An Evaluation of Conceptual Business Process Modelling Languages. In: Proceedings of the 2006 ACM Symposium on Applied Computing (SAC '06), Dijon, France, ACM.

Mendling, J., Reijers, H. A. & Van der Aalst, W. M. P. 2010. Seven Process Modeling Guidelines (7PMG). Information and Software Technology, 52.
Moody, D. L. 2009. The Physics of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering, 35.
Mouratidis, H. & Giorgini, P. 2007. Secure Tropos: A Security-Oriented Extension of the Tropos methodology. International Journal of Software Engineering and Knowledge Eng., 17.
Mouratidis, H. & Jürjens, J. 2006. Towards a comprehensive framework for secure systems development. Proc. of the International Conference on Advanced Information Systems Engineering (CAiSE). Springer.
Mouratidis, H. & Jürjens, J. 2010. From Goal-Driven Security Requirements Engineering to Secure Design. International Journal of Intelligent Systems, 25.
OMG. 2011. Unified Modeling Language (OMG UML): Superstructure [Online]. Available: http://www.omg.org/spec/UML/ [Accessed 7th Dec. 2012].
Peffers, K., Tuunanen, T., Rothenberger, M. & Chatterjee, S. 2007. A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24.
Robson, C. 2011. Real World Research, Blackwell.
Runeson, P. & Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14.
Schaad, A. & Moffett, J. 2004. Separation, Review and Supervision Controls in the Context of a Credit Application Process: A Case Study of Organisational Control Principles. Proc. of the 2004 ACM Symposium on Applied Computing (SAC).
Schattkowsky, T. & Forster, A. Year. On the Pitfalls of UML 2 Activity Modeling. In: Proceedings of the International Workshop on Modeling in Software Engineering , 2007. IEEE.
Schefer-Wenzl, S. & Strembeck, M. 2012a. Modeling Context-Aware RBAC Models for Business Processes in Ubiquitous Computing Environments. Proc. of the 3rd International Conference on Mobile, Ubiquitous, and Intelligent Computing (MUSIC). IEEE.
Schefer-Wenzl, S. & Strembeck, M. 2012b. A UML Extension for Modeling Break-Glass Policies. Proc. of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA). Springer.
Schefer-Wenzl, S. & Strembeck, M. & Baumgrass, A. 2012. An Approach for Consistent Delegation in Process-Aware Information Systems. Proc. of the 15th International Conference on Business Information Systems (BIS). Springer.
Schefer, S. 2011. Consistency Checks for Duties in Extended UML2 Activity Models. Proc. of the Sixth International Conference on Availability, Reliability and Security (ARES), International Workshop on Security Aspects in Process-Aware Information Systems. IEEE.
Schefer, S. & Strembeck, M. 2011a. Modeling Process-Related Duties with Extended UML Activity and Interaction Diagrams. Proc. of the International Workshop on Flexible Workflows in Distributed Systems, Electronic Communications of the EASST.
Schefer, S. & Strembeck, M. 2011b. Modeling Support for Delegating Roles, Tasks, and Duties in a Process-Related RBAC Context. CAiSE Workshops 2011, Proc. of the International Workshop on Information Systems Security Engineering (WISSE). Springer.
Strembeck, M. & Mendling, J. 2011. Modeling Process-related RBAC Models with Extended UML Activity Models. Information and Software Technology, 53.
Tashakkori, A. & Teddlie, C. 2010. Handbook of Mixed Methods in Social \& Behavioral Research, Sage Publications, Inc.
Zhang, X., Oh, S. & Sandhu, R. 2003. PBDM: A Flexible Delegation Model in RBAC. Proc. of the 8th ACM symposium on Access control models and technologies.
Zhao, G., Chadwick, D. & Otenko, S. 2007. Obligations for Role Based Access Control. Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 01.
Zur Muehlen, M., Indulska, M. & Kittel, K. 2008. Towards Integrated Modeling of Business Processes and Business Rules. Proc. of the 19th Australasian Conference on Information Systems