Security-Oriented Refinement of Business Processes

Peter Herrmann* and Gaby Herrmann†

*University of Dortmund
Department of Computer Science
44221 Dortmund, Germany
Peter.Herrmann@cs.uni-dortmund.de

†University of Essen
FB 5, Information Systems
45141 Essen, Germany
herrmann@wi-inf.uni-essen.de

Abstract

Economic globalization leads to complex decentralized company structures calling for the extensive use of distributed IT-systems. The business processes of a company have to reflect these changes of infrastructure. In particular, due to new electronic applications and the inclusion of a higher number of — potentially unknown — persons, the business processes are more vulnerable against malicious attacks than traditional processes. Thus, a business should undergo a security analysis. Here, the vulnerabilities of the business process are recognized, the risks resulting from the vulnerabilities are calculated, and suitable safeguards reducing the vulnerabilities are selected. Unfortunately, a security analysis tends to be complex and affords expensive security expert support. In order to reduce the expense and to enable domain experts with in-depth insight in business processes but with limited knowledge about security to develop secure business processes, we developed the framework MoSS$_{BP}$ facilitating the handling of business process security requirements from their specification to their realization. In particular, MoSS$_{BP}$ provides graphical concepts to specify security requirements, repositories of various mechanisms enforcing the security requirements, and a collection of reference models and case studies enabling the modification of the business processes. In this paper, the MoSS$_{BP}$ framework is presented. Additionally, we introduce a tool supporting the MoSS$_{BP}$-related security analysis of business processes and the incorporation of safeguards. This tool is based on object-oriented process models and acts with graph rewrite systems.

Keywords E-Commerce, Business Process, MoSS$_{BP}$, Object-Oriented Security Analysis, Graph Rewriting

1 INTRODUCTION

The evolution of the Internet from a net used predominantly by researchers to an instrument used by nearly everybody in industrial countries leads to the evolution of electronic commerce applications. These applications vary from business-to-business applications to business-to-consumer and administration-to-consumer applications. The growth of the number of e-commerce users, however, is weaker than expected. One argument for this development is that many potential users distrust e-commerce applications fearing personal damage due to real or assumed lack of security.

Companies consider this fear by designing secure business processes. When a company adapts its business processes to its IT-infrastructure in order to act with business partners (other companies or final consumers) electronically, the modification of the business processes have to fulfill certain security requirements. In particular, one has to reflect a new class of security aspects which are relevant to e-commerce-based but not to traditional business processes. For example the usage of signatures is a well established and legally unambiguous method to subscribe traditional “paper and pen” contracts. The use of digital signatures for signing contracts electronically is a new and not yet settled field in e-commerce. Moreover, due to either non-existing laws or laws containing impractical solutions, the legal consequences of electronic contract signatures are not yet clear.

A security analysis is a suitable method to address security aspects of a business process. The business process is audited for vulnerabilities and threats which may cause security risks. Based on this audit effective safeguards are selected, designed, and configured. In detail, an audit comprises a possibly iterated series of phases concerning the following subtasks (cf. [4]):

1. Identification of the business processes, their elements, and the related human principals,
2. valuation of the assets contained in the business processes and definition of their security levels,
3. identification of security requirements resp. vulnerabilities and threats,
4. assessment of resulting risks,
5. planning, design, and evaluation of suitable countermeasures.

Unfortunately, due to the complexity of real-life systems and their security requirements a security analysis tends to be complex and laborious. It is suited to well-trained security experts but not to experts in business application domains. Thus, the engineering of secure business processes is quite expensive since security experts have to be hired for this task. In the last years, however, new approaches were developed which reduce the expense and complexity of the analysis of computer systems. They utilize abstract formal models of the systems and of the security requirements (cf. [3, 11, 32, 36]). The system model forms the basis for the introduction of problem solutions which are described by model modifications. Finally, the abstract solutions are refined to implementable countermeasures. In this paper we adapt formal-based security analysis to the domain of business processes. In particular, we combine the two approaches MoSS$_{BP}$ and Object-Oriented Security Analysis in order to facilitate the automated realization of security requirements of business processes.

MoSS$_{BP}$ (**Mo**delling **S**ecurity **S**emantics of **B**usiness **P**rocesses, cf. [22]) is an approach to support domain experts which need not to be security experts. The security requirements, a business process has to fulfill in order to be secure, are modelled based on graphical design concepts provided by the framework. Moreover, MoSS$_{BP}$ introduces a procedure to handle modifications of business processes according to their security requirements. Therefore various existing enforcement procedures for security requirement as well as soft- and hardware tools realizing the protection are collected and reference models and case studies guide the modifications.

The approach *Object-Oriented Security Analysis* [24] reduces the efforts of an analysis further by using object-oriented description techniques and graph rewriting to facilitate the design of business process models and to enable automated model refinement. The corresponding tool-support is similar to object-oriented design tools which are well established in the field of software engineering (e.g., [40, 49]). The approach was successfully used for the security analysis of applications based on the middleware platform CORBA [25] and for information flow analysis of component-structured software [23], too.

In this paper, we apply Object-Oriented Security Analysis to the refinement of business processes in order to guarantee security requirements. The corresponding tool support is a useful complement to the MoSS$_{BP}$ framework. It supports the application of the MoSS$_{BP}$ methodology to modify business processes according to the required security requirements. The diagrams representing the business process and its security requirements, can be refined in a highly automated fashion by application of graph rewrite systems (e.g., [2]). A rewrite system consists of a set of graph rewrite rules. Each rule is a tuple of two graph patterns — a pre-pattern and a post-pattern —, an application condition, and an effect function. The rule can be applied to a graph if the graph contains a subgraph which is an instance of its pre-pattern. Moreover, the object attributes in the subgraph have to fulfill the application condition. By application of the rule the subgraph is replaced by an instance of the post-pattern. The attributes of the replacement objects are set according to the effect function.

The paper is structured as follows: First an overview of related approaches is given. Section 3 provides a survey of security requirements and corresponding business process elements. Thereafter we outline the architecture of the MoSS$_{BP}$-framework in section 4. The Object-Oriented Security Analysis approach and the corresponding tool support is introduced in section 5, followed by an application example in section 6.

2 RELATED WORK

The importance of business process' security is accepted in general (cf. [29]). Many approaches adapt access control and authorization methods used in database and operation system areas to the domain of business processes and workflows

(e.g., [1, 5, 9, 27, 44, 48]). But the handling of security requirements of these areas need a more broaden view. For example, two companies may interact by performing a common business process. The companies, however, may demand different, perhaps contradicting, security requirements from the common business process. A solution to this problem is provided by Pfitzmann [39]. The task management in business processes is addressed by Hung and Karlapalem [30] who use tokens for describing the capabilities and security clearances of human or computer agents performing tasks. A task is also provided by tokens and an agent may perform only a task if its tokens coincide with the tokens of the task.

A more comprehensive approach is SEMPER (Secure Electronic Marketplace for Europe, [35]) facilitating the construction of an open and secure electronic marketplace. SEMPER's main focus is the technical realization of activities fulfilling certain security requirements. The requirements are realized by means of security-related services which are classified by a four-layer architecture. The project COPS (Commercial Protocols and Services, [42]) has a broader view to security issues of electronic marketplaces than SEMPER. It enables the design of an infrastructure for marketplaces supporting all phases of a market transaction (i.e., gaining information, negotiation, completion). The security services offered by SEMPER and COPS can be assigned to the layer 1 of the MoSS$_{BP}$ architecture (cf. section 4) while the support components to design and maintain activities based on the services is part of layer 2.

A lot of work was done in the field of security analysis. Baskerville delineates three generations of security analysis methods [4]. The first generation are methods based on checklists. Here, a system is scrutinized for the availability of safeguards by means of checklists. Examples are SAFE [33], the Computer Security Handbook [28], and AFIPS [7]. Tools based on this method comprise [3, 8, 10, 19, 26, 45, 50].

The main drawback of the first generation is the informal and non-structured way of analysis which is hardly scalable to more complex computer systems. This is addressed by the so-called mechanistic engineering method [4]. This method focusses on identifying and solving detailed function system requirements facilitating the reduction of a complex system analysis into easier manageable system requirement examinations by the five steps listed in the introduction. This method was introduced by Parker [38] and Fisher [17]. A well-known tool is CRAMM (e.g., [12]) provided by the UK Government. Here, examiners scrutinize a computer system for its assets by means of checklist-based interviews with the system owners. Based on the interview results, CRAMM develops further questionnaires to determine the threats on the assets and to introduce suitable safeguards. Other tools based on mechanistic engineering are RISKPAC [13], BDSS [37], and CBISA [16].

Unfortunately mechanistic engineering-based security checks tend to be laborious and expensive. The third generation of so-called logical transformational systems intends to overcome this shortcoming by introducing abstract models of systems and security requirements. The extension SSADM of the tool CRAMM [11] is an early solution of this idea. Here, abstract specifications of a system, its problems, the security requirements, and possible technical options guiding the reviewing process are developed in parallel to the CRAMM interviews. Another approach is Baskerville's logical control design method [3] where relevant assets of a system and the threats to them are modelled in a process like way and collected in a dictionary. More recent approaches concentrate on formal modeling of processes and requirements. For instance, Kienzle and Wulf propose the use of hierarchical organized trees which are called Methodically Organized Argument Trees (MOAT) as a method to assess security of computer systems [32]. Here, security requirements are defined in the form of MOAT roots which can be refined or decomposed into subgoals resp. alternatives. Thereafter the leaves of the trees are justified either by formal verification or by informal plausibility checks. A similar method is the harmonizer approach of Leiwo and Zheng [36]. A major drawback of these approaches is that they root in abstract requirement descriptions. Thus they support the development of secure systems but are hardly suitable to the analysis of existing systems. The Risk Data Repository (RDR) approach of Kwok and Longley [34] centers on supporting security officers to maintain existing systems. The RDR consists of various domains describing relevant elements of a computer system, mappings between domains, and countermeasure diagrams.

Like us, Thoben concentrates on using security analysis for business systems. He developed an approach for the security and risk analysis of workflow based systems [47]. In contrast to our approach, he is interested mainly on the evaluation of attacks and risks which is performed by means of a fuzzy logic. The approach is not considered suitable to MoSS$_{BP}$ since it does not support the selection of countermeasures against attacks. Moreover, it is considered too complex to be used by a domain expert.

3 BUSINESS PROCESS ELEMENTS AND SECURITY REQUIREMENTS

To provide a useful definition of the security requirements for a business process, one has first to distinguish the various parts of the business process, the so-called *business process elements*. According to [15, 46] one can tell apart four main categories of business process elements:

- *Agents* represent people and machines performing activities,
- *Roles* represent rights and obligations, which are assigned to agents,
- *Artifacts* are material which is worked with,
- *Activities* represent tasks.

In order to reach a better correlation between business process elements and the security requirements to be fulfilled by them we adjusted these categories. On the one hand, for the sake of simplicity we omitted the category *role* since roles are assigned to agents. Therefore we can represent the role of an agent by the category *agent* as well. On the other hand, we refined categories in order to get more specific element types which relate directly to security requirements. The refined categories are listed below:

- *Agents*:
  - *Executing agent*: Agent performing a certain task.
  - *Ordering agent*: Agent who instructs another agent to perform a task.
  - *Agent of record*: Agent who is instructed by another agent to perform a task.
- *Artifacts*:
  - *Procedure*: Agents may act according procedures (algorithms) to execute activities.
  - *End product*: After executing a business process (or parts of it, e.g. activities), security requirements may relate to the produced end products. These security requirements may differ from security requirements of security objects used in the executed activities.
  - *Information* is represented by data. This kind of artefact includes all information which are not in the sub-categories *procedure* or *end product*.
  - *Material*: This kind of artifact includes all material which is not an *end product*.
  - *Information flow*[1]: The information flow describes all information exchanged between agents as well as all agents participating in the exchange process.
- *Activities*: An activity describes tasks in their entirety. It includes the executing agents, the procedures used, and the information/material which is used and produced (end product).

Security of computer-based systems mostly concerns confidentiality, integrity, and availability aspects. Our approach, however, is centered on domain experts who need not to be security experts as well. Therefore, a domain expert has a possibly rudimentary perception of business process security requirements which, moreover, is based on the notice of security in the traditional run of business processes. For this reason, it seems better to make a more detailed distinction of security requirements for business processes. In [20] we identified the security requirements listed below[2]. Here, we call the objects, security requirements concern with (i.e., agents, artifacts, and activities), *security objects* and persons acting as intruders *security subjects*.

[1] Generally, in business process models the information flow is not specified explicitly. However, it is relevant to realize certain security requirements.
[2] The list is subject to changes since a new business process may call for new security requirements.

- Common security requirements:
  - *Confidentiality*
  - *Integrity*
  - *Availability*
- Protection of personality:
  - *Anonymity*: The true identity of a security object is hidden.
  - *Pseudonymity*: Here, anonymity of a security object is realized in principle, but may be uncovered by authorized subjects.
  - *Privacy*: According to [29], privacy "is the right of individuals and organizations to control the collection, storage and dissemination of their information or information about themselves."
- Bindings:
  - *Legal binding*: An information or a specific end product is legally binding if it contains an agreement which can be proven at court.
  - *Non-repudiation*: Two different views are possible: At first, it should not be possible for a specific agent to deny doing activities in principle. At second, it should not be possible for agents to deny doing certain specific activities.
  - *Mutual dependencies*: Security objects are mutual depended if activities or properties of a security object lead to activities or properties of a related object.
- Physical property:
  - *Authenticity*: A security object is authentic, if it is what it pretends to be. It may be a copy.
  - *Originality*: A security object is original, if it is what it pretends to be and it is not a copy.
  - *Rights to use*: The rights to use a specific security object specify, which people are allowed to use the object and in which manner.
  - *Copyright*: The copyright of a security object is the right to reproduce it.
- *Hiding activities*: Three different views are possible: At first, an activity must be invisible if it is carried out, delegated to, or delegated by a certain agent. At second, an activity is hidden if it is executed by means of a certain procedure. At third, the performing of an activity must not be visible at all.

Of course, not every security requirement is relevant for each business process element (e.g., the requirement *copyright* is not reasonable for an *activity*). The useful correlations between security requirements and business process elements are listed in table 1 (cf. also [20, 21]). Moreover, in some relations we need further refinements of security requirements in order to address specific characteristics of security subjects or objects. For instance, with respect to *confidentiality* we have to distinguish which characteristics of a security object has to be confidential against possible security subjects. Possible characteristics of security objects are the content, structure, and existence of a security object.

4 MOSS$_{BP}$: A FRAMEWORK TO SUPPORT SECURITY OF BUSINESS PROCESSES

Domain experts have in-depth knowledge of the specific security requirements of a traditionally running business process. Moreover, only a domain expert knows if the risks resulting from the vulnerabilities of a business process are bearable. If a traditional business process is refined to a modern computer-based process, the domain expert demands that also the refined process fulfills the security requirements of the original (e.g., a digital signature should be as legally binding as a traditional

business process elements \ **security requirements**	confidentiality	integrity	availability	anonymity	pseudonymity	privacy	legal binding	non-repudiation	mutual dpendencies	authenticity	originality	rights to use	copyright	hiding activities
procedure	x	x	x	-	-	-	-	x	x	x	-	x	x	x
end product	x	x	x	-	-	-	x	-	x	x	x	x	x	-
information	x	x	x	-	-	x	x	-	x	x	x	x	x	-
material	x	x	x	-	-	-	-	-	x	x	x	x	-	-
executing agent	x	-	x	x	x	-	-	x	x	x	-	x	x	x
ordering agent	x	-	-	x	x	-	-	x	x	x	-	-	-	x
agent of record	x	-	x	x	x	-	-	x	x	x	-	x	-	x
activity	x	-	-	-	-	-	-	x	x	-	-	-	-	x

Table 1: Correlation between business process elements and security requirements

"paper and pen"-signature). But as denoted in section 3, in general, domain experts are not computer security experts and their perception of the business process security requirements are often rudimentary. Therefore, with respect to the design of secure computer-based business processes, a domain expert needs help in the exact definition of security requirements suitable to a business process. For example, she/he may demand, that a specific communication should be confidential. This formulation, however, is ambiguous since it does not state clearly if the identity of the communicating agents, the content of the communication, or the mere existence of the communication should be confidential. Furthermore, to support the design of a secure business process, a domain expert should perform a security analysis of the business process in order to guarantee that she/he considers all relevant security requirements. Moreover, the analysis supports the domain expert to determine the risks for the business process and the selection of suitable safeguards enforcing security requirements.

The project MoSS$_{BP}$ [20] was developed to support domain experts to develop exact definitions of security requirements as well as to modify business processes in order to fulfill the security requirements. MoSS$_{BP}$ uses the popular Unified Modeling Language (UML, cf. [6]) to create business process specifications. Moreover, it contains repositories of graphical concepts representing security requirements relevant to business process elements as well as case studies and reference models showing their application in order to facilitate business process modifications. Furthermore, procedures to create safeguards as well as software- or hardware building blocks are collected in repositories in order to clarify the design of the countermeasures. The framework is organized in an architecture of four layers as depicted in figure 1 (cf. [22]):

Layer 4 This layer supports the development of an abstract UML-based business process specification by means of a repository of graphical concepts describing typical business process elements and security requirements. The UML-diagrams are created by using these concepts.

Layer 3 To facilitate the modification of business processes, a set of reference models and case studies is included describing sub-processes enforcing security requirements. The sub-processes contain basic security elements and security activities. *Basic security elements* are abstract descriptions of security mechanisms which enclose all information for their realization (e.g., "verify digital signature *sig* of alleged signatory *White*"). An example of a *security activity* is the activity "deliver a licence anonymously under consideration of its originality".

Layer 2 This layer contains procedures to realize the basic security elements and the security activities of layer 3 (e.g., a procedure checking if the digital signature can be decrypted by means of the public key of the contract partner; a procedure checking the originality of the contract partner's public key by contacting a trusted third party acting as a

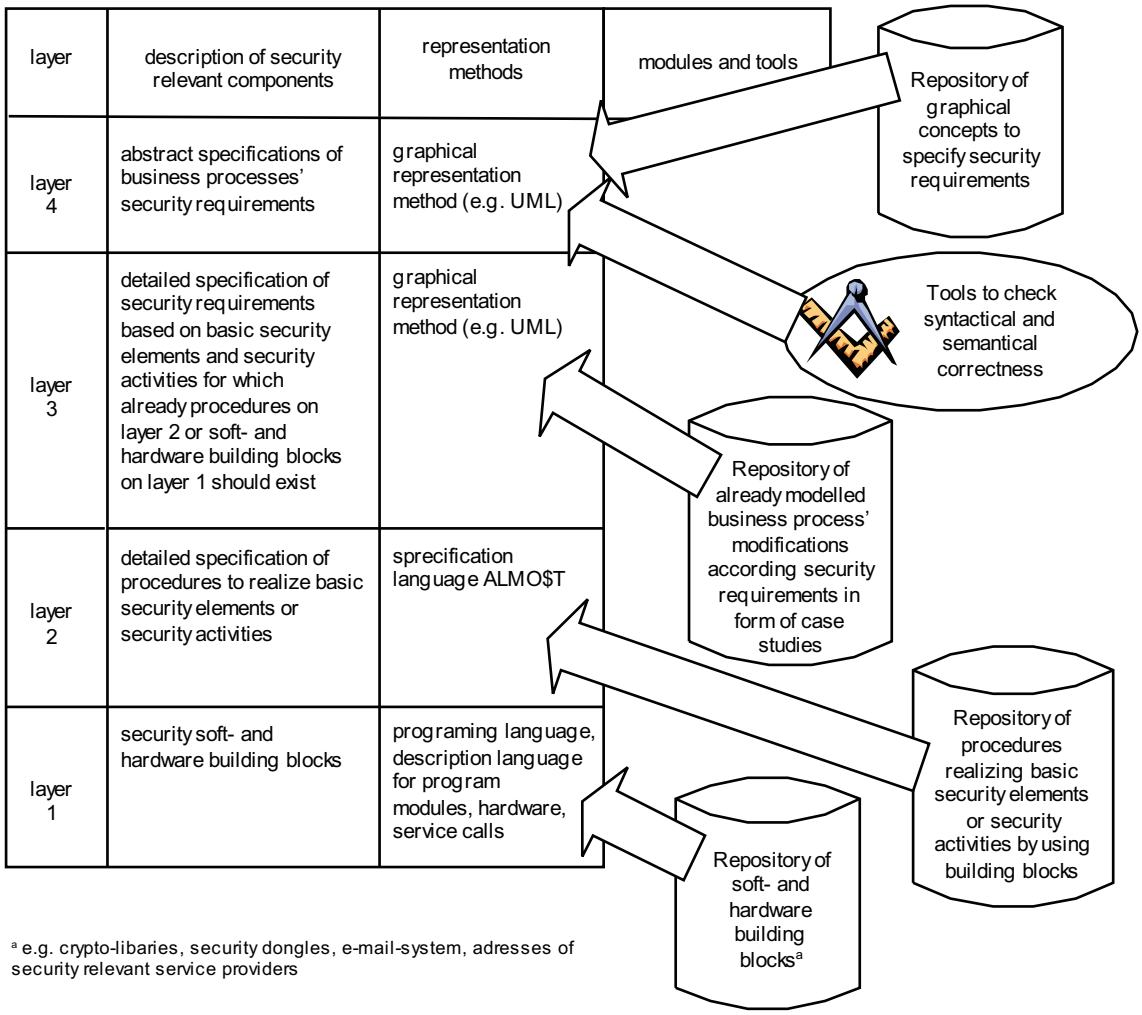

Figure 1: MOSS$_{BP}$ Architecture

certification authority). To describe these procedures and their combination in an easy and comprehensible fashion, we use the specification language ALMO$T (**A L**anguage for **Mo**deling **S**ecure Business **T**ransactions, cf. [41]) which was developed in cooperation with the project COPS [42].

Layer 1 Soft- and hardware building blocks to realize the procedures of layer 2 are collected in this layer. Examples, here, are a hardware encryption and decryption chip resp. a distributed application enabling communication with certification authorities.

After creating a business process specification and defining suitable security requirements (layer 4), the domain expert checks the repository of layer 3 for suitable reference models or case studies enforcing the security requirements. Thereafter, layer 2 is checked for procedures realizing the basic security elements and security activities contained in the reference models or case studies. These procedures use the soft- and hardware building blocks collected in layer 1. If no reference models, case studies, basic security elements, security activities, ALMO$T-procedures, or suitable building blocks are available to secure a particular business process, the domain expert engages a security expert realizing the missing components. If the security expert cannot enforce a security requirement, the domain expert has to decide between reducing the security demands and denying execution of the business process.

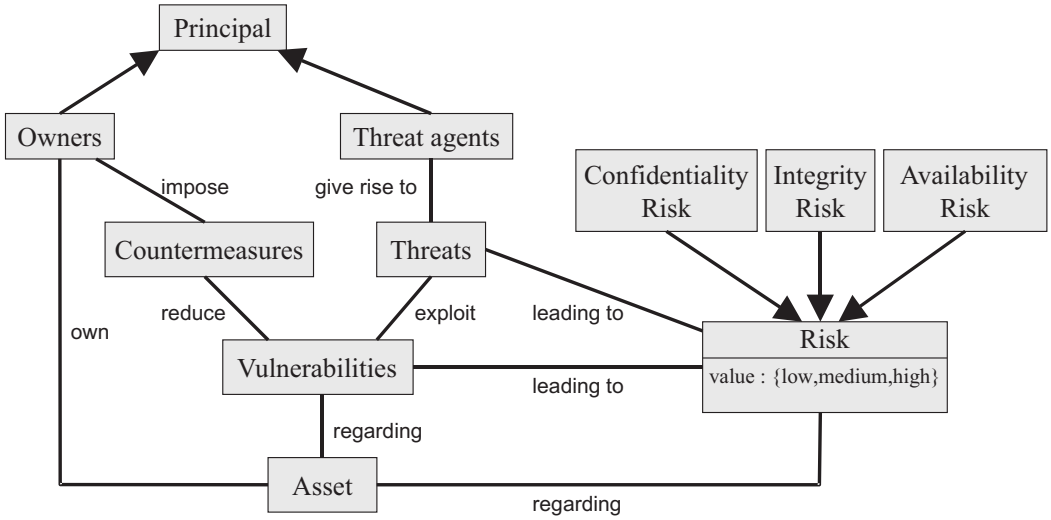

Figure 2: CC security classes

To support the modelling of business processes in UML, a graphical syntax editor seems to be necessary to enable syntactical and semantical correct business process specifications. This editor, moreover, should help to identify relevant security requirements. Furthermore, the modification process of layer 3 should also be facilitated by a tool since due to the large spectrum of different business processes one needs a confusing high number of different reference models or case studies. Here, a security analysis tool seems helpful since it enables the highly automated modification of business processes by integration of suitable basic security elements and security activities as well as the realizing of this elements by basic building blocks. Therefore we extended MoSS$_{BP}$ by an adaption of the object-oriented security analysis tool introduced in section 5 which can be used as a graphical syntax editor and for the security analysis of business processes.

5 OBJECT-ORIENTED SECURITY ANALYSIS OF BUSINESS PROCESSES

The security analysis of IT systems is standardized by ISO/IEC in the so-called set of *Common Criteria* (CC, cf. [31]) providing a methodology for vulnerability detection, risk assessment, and countermeasure integration. The terminology with respect to security issues in the CC is more technical than those used in MoSS$_{BP}$ to describe security purposes of business processes. Therefore, in this section we give a short introduction to the CC and its terminology to support the understanding of the approach for more technical oriented readers. Moreover, the relationship between the CC and the business oriented terminology of MoSS$_{BP}$ is mentioned, too.

Figure 2 delineates the main security classes and associations defined by the CC. The security relevant parts of a system are assets for their owners which, unfortunately, are constantly exposed to threats by intruders, called threat agents, who exploit the vulnerabilities of the assets for attacks. Therefore, the assets underly security risks. In order to minimize these risks, the asset owners impose countermeasures reducing the vulnerabilities of the assets. In our context the security objects (i.e., business process elements) correspond with the assets while the security subjects describe the intruders attacking an asset.

Our object-oriented approach [23, 24] facilitates the design of CC-compliant business models by providing a library of basic asset classes like networks, stations, applications, and data as well as associations between the classes. Moreover, more specialized classes are inherited from the basic classes in order to support modelling of business processes. We designed classes specifying the activities, agents, roles, and artifacts (cf. section 3) participating in a business process. Utilizing the class libraries, our tool SEMBA based on the toolset ARGO [49] facilitates the modelling of business processes and sub-processes in the form of UML object diagrams (cf. [6]).

Security level	Threat seriousness level						
	1	2	3	4	5	6	7
1	0	0	1	1	2	3	3
2	0	1	1	2	3	3	4
3	1	1	2	3	3	4	5
4	1	2	3	3	4	5	5
5	2	3	3	4	5	5	6
6	3	3	4	5	5	6	7
7	3	4	5	5	6	7	7

Table 2: Matrix for calculating risk values

Class attributes are used to describe the amount of protection, a business process needs to fulfill a certain security requirement. Each class contains attributes for all security requirements relevant for the modelled business process element (cf. table 1). While there are various methods to describe the amount of protection for an asset, our approach refers to the seven security levels corresponding to the evaluation assurance levels defined in the CC. For instance, level 7 shall be assigned to the legal binding property of a contract if a breaking of this contract without successful legal action leads to total collapse of the institution.

According to the CC, in the next analysis phase vulnerabilities and threats on the assets are identified. Furthermore, one has to estimate the seriousness of the vulnerabilities (i.e., the likelihood that they are in fact exploited to attack an asset). This seriousness, of course, depends on the safeguards used to protect the security requirement. For instance, a successful appeal against the repudiation of a contract is more likely if the subscription was witnessed by a notary. The seriousness is modelled by a class attribute (threat seriousness level) which, similarly to the security levels of the assets, may contain seven values.

Vulnerability and threat identification, however, tends to be laborious and complicated and therefore is not well suited to domain experts with limited knowledge on security. Therefore and in order to be consistent with MoSS$_{BP}$, we altered the procedure in this place. Instead of adding vulnerabilities and threats, the domain expert may identify security requirements and assign them to the business process elements. The security requirements are also modelled as classes and instances of these classes are added to the business process model. The tool can support the analysis process by suggesting useful security requirements itself. Here, for the first time in the security analysis we apply a graph rewrite system which modifies the UML object diagram by adding security requirement objects. The graph rewrite system consists of graph rewrite rules (cf. [2]) each consisting of a pre-pattern, a post-pattern, an application condition, and an effect function. The pre-pattern contains a UML diagram describing a business process sub-system which, to be secure, requires a certain security requirement. The post-pattern describes the sub-system extended by an object modelling the security requirement and edges linking the new object with certain sub-system objects. Thus, by executing the rule, the security requirement object is added to each part of the UML model corresponding to the pre-pattern. The application condition may be used to restrict the execution of a rule to certain object attribute settings while by the effect function attributes may be altered.

Thereafter a graph rewrite system is used for determining the risks on the assets. For each pair of a business process element and a security requirement a risk object is created stating the risk that the business process element may not be used correctly due to a violation of the corresponding security requirement. Moreover, the tool calculates the risk level which is modelled by a class attribute, too. The risk level depends on the security level of the asset and on the seriousness level of the security requirement (cf. [14]). Currently, we apply the matrix[3] in table 2 which, however, can be altered according to the security policies of an enterprise. Finally, the domain expert has to assess the risks which is also supported by a graph rewrite system. If all risks are bearable, the security analysis can be terminated now.

If the risks for the business process cannot be accepted, the security analysis proceeds to the safeguard assignment phase in order to enforce the security requirements and, in consequence, to reduce the risks. The countermeasures are defined in another class library and a graph rewrite system is used to introduce them to the UML diagram. Attributes of a countermeasure object describe a protection level and the estimated costs of imposing the countermeasure. In a first step, the

[3]The risk level 0 states that no risk is assumed and the risk object is removed.

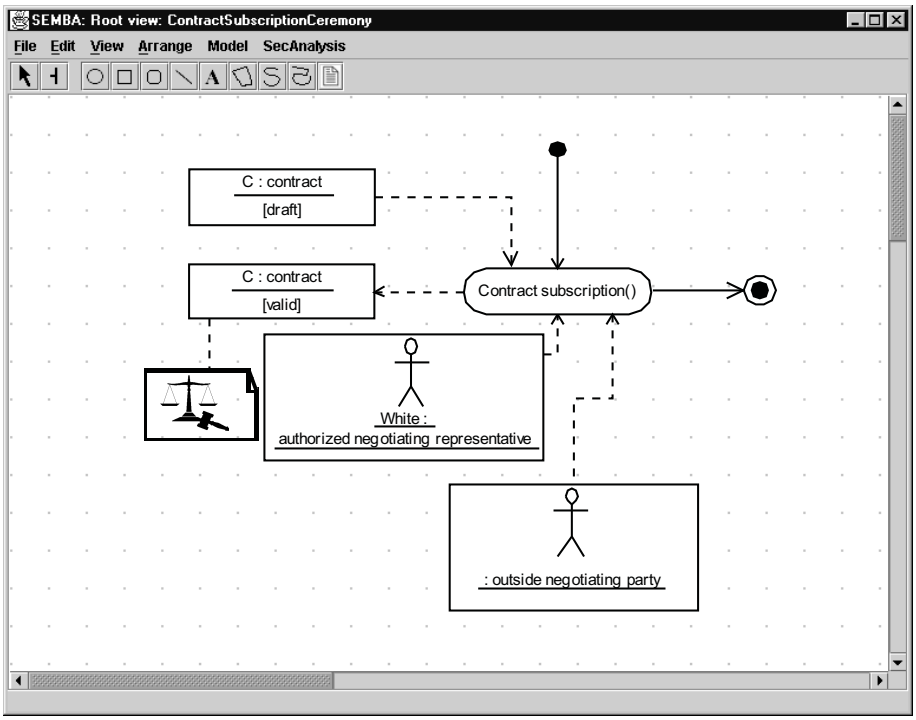

Figure 3: Contract Subscription Ceremony

tool suggests for each pair of a business process element and a risk object all countermeasures with a sufficient protection level (i.e., the protection level must be equal or higher than the risk level). Thereafter the tool compares the costs of the countermeasures and selects one with a good relation between costs and level of protection.

Since countermeasures may contain vulnerabilities themselves, the analysis iterates the vulnerability and threat identification phase as well as the risk evaluation phase. If the newly calculated risks can be accepted as bearable, the analysis terminates. Otherwise, new countermeasures are suggested and further iterations take place. After terminating the security analysis the domain expert may modify a real business process based on the resulting UML model.

6 EXAMPLE

As an example for the SEMBA-based security analysis we use a simple business process describing a contract subscription ceremony. In the first analysis step the domain expert creates a UML object diagram modelling the business process as depicted in figure 3[4]. SEMBA supports the design process by a library of process element classes and edges to link the class instances. In our example, we use three activity objects instantiated from the classes *process start*, *signing of contract*, and *process end*. The objects are linked by three solid arrows describing their order. Moreover, we have two objects representing the agents which participate in the ceremony. The object *White* of class *authorized negotiating representative* models the subscriber acting for the enterprise executing the business process, while the unnamed object of class *outside negotiating party* represents the partner enterprise. The objects are linked with the activity *Contract subscription* by dotted arrows of class *subscriber*. Since the business process deals with the contract *C* to be signed in both its draft and its valid state, we use two artifact objects representing the contract in either state. The dotted arrows of type *subscribe contract* depict that *C* is getting valid by means of the activity *Contract subscription*.

[4]Besides objects modelling the business process, this figure also contains a security requirement object (icon with the balance symbol) which, however, is added later to the diagram.

In the second step of the security analysis, the domain expert has to evaluate the importance of a correct subscription ceremony for the enterprise with respect to various security requirements. Let us assume that with respect to legal binding the contract is of average importance for the company. Therefore the domain expert associates the security requirement *legal binding* to the object *C* and allocates the intermediate security level 4 to the attribute *legal binding security level* of the object *C* in the valid state.

Thereafter, the graph rewriting capability of SEMBA is used for the first time in order to add additional relevant security requirements. In this example, for each contract object two security requirement objects of type *confidentiality* are generated describing that the existence resp. the content of the contract may be confidential. Furthermore, the contract objects are linked with an security requirement object *integrity* each, stating that the contract integrity must not be spoiled by changing its content. The two agent objects may sign the contract anonymously or pseudonymously. Therefore, security requirements of type *anonymity* and *pseudonymity* are created and linked to the agent objects. Finally, the activity object *Contract subscription* is supplemented by a security requirement object of the class *confidentiality* describing that the subscription act may be confidential.

Now the domain expert may decide which security requirements are necessary. Here, we assume that she/he selects only the security requirement *legal binding* which guarantees also the requirement content integrity. Thus, the other security requirement objects are removed and the diagram has the state depicted in figure 3. Here, the security requirement object of type *legal binding* is modelled by the balance icon. In the next step the seriousness of the security requirement is calculated from SEMBA. Since we do not have any mechanisms protecting the business process, the danger of successful disputes is high and seriousness level 7 is selected.

Thereafter, SEMBA creates an object describing the risk that the legal binding of contract *C* is successfully violated. According to table 2 the risk level is set to the value 5. The domain expert decides that the risk is too high to be accepted.

Thus, one has to modify the business process adding safeguards in order to guarantee legal binding of the subscribed contract. Legal binding of a contract requires the proof of the agreement codified in the information at court. The measurements used to realize this provableness, however, depends on the legal environment the contract partners are acting in. We assume that the legal environment is Germany which is stated by the attribute *legal environment* of object *C* in the valid state. Moreover, the kind of signature has to be fixed by setting the attribute *kind of signature* in the activity object *Contract subscription*. Here, the signature is performed electronically.

In Germany, the provableness of electronic agreements is defined by the law “Signaturgesetz” [18]. It tells apart three kinds of digital signatures:

Electronic Signature is electronic data which is added to other electronic data or relates to them and is used for authentication purposes.

Advanced Electronic Signature is a digital signature according to electronic signatures plus the following requirements:

- The digital signature is assigned solely to the owner of the signature key.
- It is possible to identify the owner of the signature key.
- The digital signature is created only with instruments which are controlled exclusively by the key owner.
- The digital signature is linked with the corresponding data in a way that ex post modifications of the data will be recognized.

Qualified Electronic Signature is a digital signature according to advanced electronic signatures plus the following requirements:

- The digital signature was created using a qualified certificate which was valid at siganture’s creation time.
- The digital signature was created using a secure device.

A qualified electronic signature has the highest security level and its probative force at court is the same as with a traditional signature.

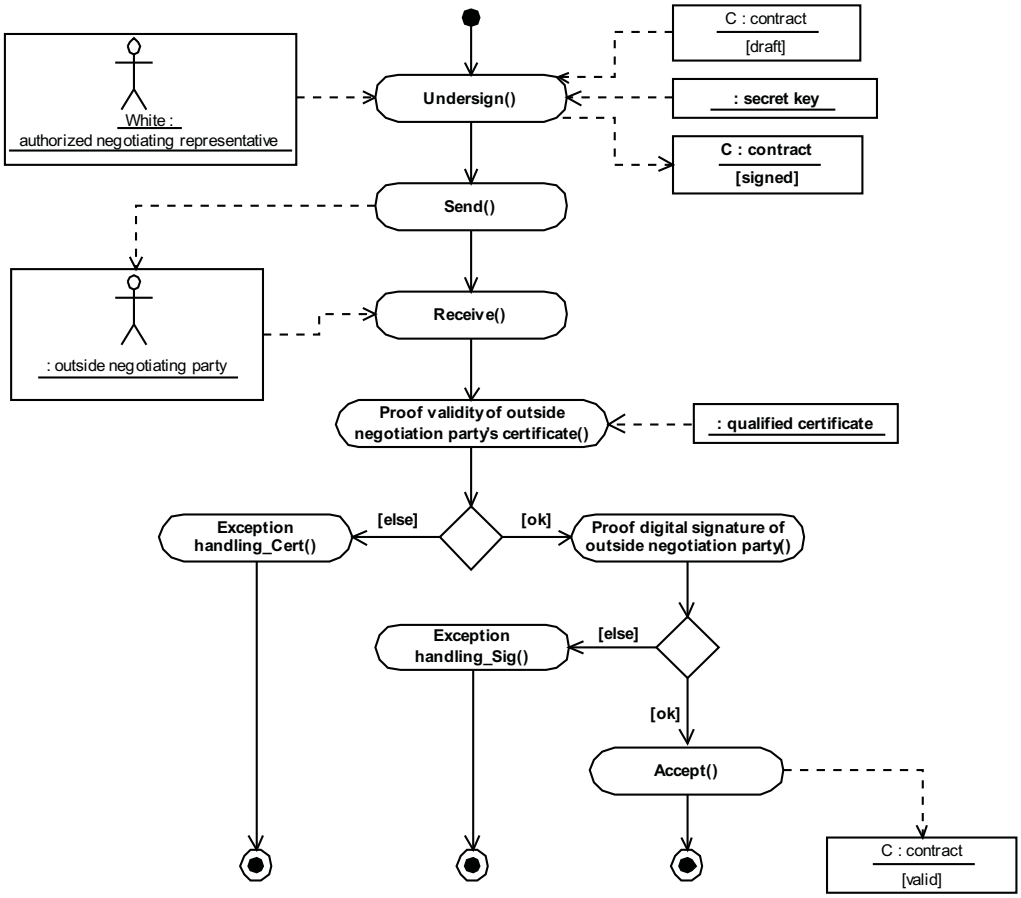

Figure 4: Augmented Subscription Ceremony

The MoSS$_{BP}$ framework contains at its layer 3 (cf. figure 1) a reference model describing subscriptions with qualified electronic signatures. The business process model is modified according to this model by a graph rewrite system leading to the UML diagram depicted in figure 4. The main difference to the original business process is the application of a qualified certificate[5] which is modelled by an unnamed object of class *qualified certificate*. The validity of a qualified electronic signature of the outside negotiating party may be proved using this certificate. To use digital signatures, moreover, the activities of the business process have to be modified. At first, in the activity *Undersign* the agent represented by the object *White* signs the contract C with its *secret key* moving C from the draft state to the signed state. Thereafter the signed contract C is sent to the negotiation party (activity *Send*). After receiving C from the outside negotiation party, the *qualified certificate* of this party is proved for being valid (activity *Proof validity of outside negotiation party's certificate*). If this proof fails, an exception handling takes place (activity *Exception handling_Cert*). If it succeeds, the digital signature of the outside negotiation party is proved by means of the public key in the certificate (activity *Proof digital signature of outside negotiation party*). If this proof fails, an exception handling takes place as well (activity *Exception handling_Sig*). Otherwise the contract is accepted (activity *Accept*) and contract C is getting valid.

Since the risk for the modified business process might still be too high, the domain expert repeats the security requirement evaluation. The application of qualified electronic signatures reduces the danger of successful disputes and SEMBA sets the seriousness level for the contract to level 1[6]. According to table 2 level 1 is calculated for the risk of legal binding violations. The domain expert accepts this very low risk as bearable and terminates the security analysis, here.

[5]Qualified certificates are only issued to natural persons and the issuing certification authority complies with certain properties in order to achieve trustworthiness.

[6]By electronic signatures or advanced electronic signatures the seriousness is set to a higher level.

7 CONCLUSION

In this paper, we introduced the MoSS$_{BP}$ framework supporting domain experts to define security requirements for business processes and to modify business processes in order to guarantee the requirements. The modelling of business processes, the identification of suitable security requirements, and the introduction of safeguards enforcing the requirements are supported by the object-oriented modelling tool SEMBA.

Currently, SEMBA models only functional aspects of business processes. A complete description of a business process, however, considers also at least two other perspectives (cf. [43]): at first, the *informational perspective* represents the information entities, their structure, and relationships between them. At second, the *organizational perspective* depicts in which place of an enterprise and by which agents activities are performed. For the selection of modifications in layer 3 of the MoSS$_{BP}$-architecture one has often to consider these perspectives as well. For instance, the modification in section 6 can only take place if certificates to prove digital signatures are available. The existence of the certificates, however, can only be detected by considering the informational perspective. Therefore we plan to extend SEMBA in order to support also UML diagrams modelling informational and organizational aspects. Moreover, the graph rewrite rules shall be extended in order to modify different diagrams simultaneously.

References

[1] V. Atluri, W.-K. Huang, and E. Bertino. An Execution Model for Multilevel Secure Workflows. In *Proceedings of the IFIP 11.3 Workshop on Database Security*, 1997.

[2] R. Bardohl, G. Taentzer, M. Minas, and A. Schürr. Application of graph transformation to visual languages. In *Handbook on Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages and Tools*, chapter 1. World Scientific, 1999.

[3] Richard Baskerville. *Designing Information Systems Security*. Wiley & Sons, Chichester, 1988.

[4] Richard Baskerville. Information Systems Design Methods: Implications for Information Systems Development. *ACM Computing Surveys*, 25(4):375–414, December 1993.

[5] E. Bertino, E. Ferrari, and V. Atluri. A Flexible Model Supporting the Specification and Enforcement of Role-Based Authorizations in Workflow Management Systems. In *Proceedings of the 2nd ACM Workshop on Role-Based Access Control*, 1997.

[6] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, 1999.

[7] P. Browne. *Security: Checklist for Computer Center Self-Audits*. AFIPS Press, Arlington, 1979.

[8] T. Bui and T. Sivasankaran. Cost-Effectiveness Modeling for a Decision Support System in Computer Security. *Computer Security*, 6(2):139–151, 1987.

[9] C. Bußler. Access Control in Workflow Management Systems. In *Proceedings of the IT Security'94 Conference*, pages 165–179. Oldenbourg-Verlag Munich, 1995.

[10] J. Carroll and W. MacIver. Towards an Expert System for Computer Facility Certification. In *Computer Security A Global Challenge*, pages 293–306. North-Holland, Amsterdam, 1984.

[11] CCTA. *SSADM-CRAMM Subject Guide for SSADM Version 3 and CRAMM Version 2*. CCTA, London, 1991.

[12] W. R. Chisnall. Applying Risk Analysis Methods to University Systems. In *Proceedings of the EUNIS 97 Congress*, Grenoble, 1997.

[13] Computer Security Consultants, Ridgefield. *Using Decision Analysis to Estimate Computer Security Risk*, 1988.

[14] R. Courtney. Security Risk Assessment in Electronic Data Processing. In *AFIPS Conference Proceedings of the National Computer Conference 46*, pages 97–104, Arlington, 1977. AFIPS.

[15] Bill Curtis, Marc I. Kellner, and Jim Over. Process modeling. *Communications of the ACM*, 35(9):75–90, 1992.

[16] Thomas Finne. Computer Support for Information Security Analysis in a Small Business Environment. In Jan H. P. Eloff, editor, *Proceedings of the IFIP TC11 WG 11.2 on Small Systems Security*, pages 73–88, Samos, 1996.

[17] R. Fisher. *Information Systems Security*. Prentice-Hall, Englewood Cliffs, 1984.

[18] Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften. Bundesgesetzblatt, part I, no. 22, May 2001. In German. An unofficial translation can be found in the WWW under http://www.sicherheit-im-internet.de/download/026-Signaturgesetz_englisch.doc.

[19] S. Guarro. Principles and Procedures of the LRAM Approach to Information Systems Risk Analysis and Management. *Computer Security*, 6(6):493–504, 1987.

[20] Gaby Herrmann. *Verläßlichkeit von Geschäftsprozessen — Konzeptionelle Modellbildung und Realisierungsrahmen*. Doctoral thesis, University of Essen, 2001. In German.

[21] Gaby Herrmann and Günther Pernul. Towards Security Semantics in Workflow Management. In *Proceedings of the 31st Annual Hawaii International Conference on System Sciences (HICSS-31)*, pages 766–767. IEEE Computer Society Press, January 1998.

[22] Gaby Herrmann and Günther Pernul. Viewing Business Process Security from Different Perspectives. *International Journal of Electronic Commerce*, 3(3):89–103, 1999.

[23] Peter Herrmann. Information Flow Analysis of Component-Structured Applications. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'2001)*, pages 45–54, New Orleans, December 2001. ACM SIGSAC, IEEE Computer Society Press.

[24] Peter Herrmann and Heiko Krumm. Object-oriented security analysis and modeling. In *Proceedings of the 9th International Conference on Telecommunication Systems — Modelling and Analysis*, pages 21–32, Dallas, March 2001. ATSMA, IFIP.

[25] Peter Herrmann, Lars Wiebusch, and Heiko Krumm. Tool-Assisted Security Assessment of Distributed Applications. In *Proceedings of the 3rd IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems (DAIS 2001)*, pages 289–294, Krakow, September 2001. Kluwer Academic Publisher.

[26] L. Hoffman, E. Michelman, and D. Clements. SECURATE — Security Evaluation and Analysis using Fuzzy Metrics. In *AFIPS Conference Proceedings of the National Computer Conference 47*, pages 531–540, Arlington, 1978. AFIPS.

[27] R. Holbein, S. Teufel, and K. Bauknecht. The Use of Business Process Models for Security Design in Organizations. In S. Katsikas and D. Gritzalis, editors, *Proceedings of the IFIP TC11 Conference on Information Systems Security*, pages 13–22. Chapman & Hall, London, 1996.

[28] D. Hoyt. *Computer Security Handbook*. Macmillan, New York, 1973.

[29] A. Hudoklin and A. Stadler. Security and Privacy of Electronic Commerce. In *Proceedings of the 10th International Bled Electronic Commerce Conference*, pages 523–535. Moderna Organizacija, 1997.

[30] Patrick C.K. Hung and Kamalakar Karlapalem. A Paradigm for Security Enforcement in CapBasED-AMS. In *Proceedings of the 2nd IFCIS International Conference on Cooperative Information Systems (CoopIS'97)*, pages 79–88, 1997.

[31] ISO/IEC. *Common Criteria for Information Technology Security Evaluation*, 1998. International Standard ISO/IEC 15408.

[32] Darrell M. Kienzle and William A. Wulf. A Practical Approach to Security Assessment. In *Proceedings of the Workshop New Security Paradigms '97*, pages 5–16, Lake District, 1997.

[33] L. Krauss. *SAFE: Security Audit and Field Evaluation for Computer Facilities and Information Systems*. Amacon, New York, 1972.

[34] Lam For Kwok and Dennis Longley. A Security Officer's Workbench. *Computers & Security*, 15(8):695–705, 1996.

[35] G. Lacoste. *SEMPER: A Security Framework for the Global Electronic Marketplace*, 1995. SEMPER document 431LG042/Draft/25 August 1997/public.

[36] Jussipekka Leiwo, Chandana Gamage, and Yuliang Zheng. Harmonizer — A Tool for Processing Information Security Requirements in Organization. In *Proceedings of the 3rd Nordic Workshop on Secure Computer Systems (NORDSEC'98)*, Trondheim, 1998.

[37] W. Ozier. Risk Quantification Problems and Bayesian Decision Support System Solutions. *Information Age*, 11(4):229–234, 1989.

[38] D. Parker. *Computer Security Management*. Reston, 1981.

[39] Andreas Pfitzmann. Technologies for Multilateral Security. In G. Müller and K. Rannenberg, editors, *Multilateral Security in Communications*, volume 3: Technology, Infrastructure, Economy, pages 85–91. Addison-Wesley, Munich, 1999.

[40] Terry Quatrani. *Visual Modeling with Rational Rose 2000 and UML*. Addison-Wesley, 2 edition, 2000.

[41] Alexander Röhm, Gaby Herrmann, and Günther Pernul. A Language for Modelling Secure Business Transactions. In *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99)*, pages 22–31. IEEE Computer Society Press, 1999.

[42] Alexander Röhm and Günther Pernul. COPS: a model and infrastructure for secure and fair electronic markets. *Decision Support Systems Journal*, 29(4):343–355, 2000.

[43] Alexander Röhm, Günther Pernul, and Gaby Herrmann. Modelling Secure and Fair Electronic Commerce. In *Proceedings of the 14th Annual Computer Security Application Conference (ACSAC'98)*, pages 155–164. IEEE Computer Society Press, 1998.

[44] H. Shen and P. Dewan. Access Control for Collaborative Environments. In *Proceedings of the CSCW'92 Conference*. ACM Press, New York, 1992.

[45] S. Smith and J. Lim. An Automated Method for Assessing the Effectiveness of Computer Security Safeguards. In *Computer Security A Global Challenge*, pages 321–328. North-Holland, Amsterdam, 1984.

[46] G. Starke. Business Models and their Description. In G. Chroust and A. Benczur, editors, *Workflow Management: Challenges, Paradigms, and Products (CON'94)*, volume 76 of *Schriftenreihe der Österreichischen Computer Gesellschaft*, pages 134–147. Oldenbourg-Verlag Wien, 1994.

[47] Winfried Thoben. *Wissensbasierte Bedrohungs- und Risikoanalyse Workflow-basierter Anwendungssysteme*. Reihe Wirtschaftsinformatik. B.G. Teubner-Verlag, Stuttgart, 2000. In German.

[48] R. Thomas and R. Sandhu. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. In *Proceedings of the IFIP WG11.3 Workshop on Database Security*. Chapman & Hall, London, 1997.

[49] Tigris. *ArgoUML Vision*, 2000. argouml.tigris.org/vision.html.

[50] M. Zviran, J. Hoge, and V. Micucci. SPAN — a DSS for Security Plan Analysis. *Computer Security*, 9(2):153–160, 1990.