Development Support of Anonymous Business Processes

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

Peter Herrmann
University of Dortmund
Computer Science Department
44221 Dortmund, Germany
Peter.Herrmann@udo.edu

Abstract

In today's globalized economic world companies get a more and more decentralized structure. In consequence, by application of large distributed IT-systems the business processes of the enterprizes are often carried out electronically. These mostly Internet-based distributed systems, however, are vulnerable against malicious attacks and the business processes have to be modified in order to fulfill an extended set of security requirements. Unfortunately, however, a company often cannot implement the necessary modifications itself since the own employees have a too limited knowledge of the security mechanisms to be used. Therefore external security experts have to be hired who, however, lack the sufficient knowledge of the business process in order to decide which security requirements have to be fulfilled to guarantee a secure execution.

To bring the domain experts who know the business processes and the security experts together, 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 case studies enabling the modification of the business processes. In this paper, we introduce how to apply the mechanisms of the MoSS$_{BP}$-framework in order to address the necessary security requirements. In particular, we point out how to use MoSS$_{BP}$ from the domain expert's as well as from the security expert's views. As an example, we use a business process performing requests for tenders in an electronic procurement scenario. The business process is altered in order to carry out the requests for tenders anonymously.

Key Words: E-Commerce, Business Process, MoSS$_{BP}$, Anonymity

1 INTRODUCTION

In the last years, electronic commerce has got very popular in the industrialized countries. Nevertheless, as proved dramatically by the recent .com crisis, the growth in the number of people and companies using business-to-business, business-to-consumer, or administration-to-consumer applications, is much weaker than previously predicted. One argument for this development is that many potential users distrust e-commerce applications fearing personal or corporate damage due to real or assumed lack of security. Therefore we see a high impact of security in order to realize business processes electronically (cf. [20]).

When transferring a business process which was previously carried out in a traditional way to an electronic execution, on the one hand its owner has to consider security issues resulting from the IT-infrastructure and from the IT-system. These security issues are independent of the structure of a business process since they are caused by vulnerabilities and threats which are inherent to particular IT-systems and the IT-infrastructure realizing the systems. To handle these security issues a security analysis by a security expert is necessary (cf. e.g., [2]).

On the other hand, the company being responsible for a business process has also to reflect security aspects which are inherent to the particular structure of the business process. In contrast to the IT-system based security issues, the domain experts (i.e., the responsible company employees) have the best knowledge of these security aspects which are relevant in traditional as well as in electronically performed business process executions. In a traditionally executed business process, the security issues usually are not formally specified since the domain experts have an intuitive understanding of the related threats which results from long years of experience. For instance, everybody knows that a contract has to be signed by all involved parties in order to guarantee that it is legally binding.

Often, however, the domain experts lack the knowledge how to make electronically handled business processes secure. For instance, it is very complex to guarantee that an electronic contract where manual signatures are not available is still legally binding. To support domain experts handling also the special security requirements of electronic business processes, we developed MoSS$_{BP}$ (**Mo**deling **S**ecurity **S**emantics of **B**usiness **P**rocesses, cf. [13]). MoSS$_{BP}$ provides a framework which enables the modelling of the security requirements based on the graphical design concepts of the popular Unified Modeling Language (UML, cf. [4]). 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 case studies guide the modifications.

A security expert is only needed if the necessary elements to realize a particular security requirement are not available in the MoSS$_{BP}$ repositories. The security expert tries to include the necessary tools resp. case studies into the repositories. If no such tool or case study exists or can be created, the domain expert has to decide if the corresponding security requirements can be reduced without making the related business process insecure. If that is not possible, the business process cannot be executed at all.

In this paper, we concentrate on the views of both the domain experts and the security experts. On the one hand, we will outline how a domain expert specifies a secure business process resp. a modification of a business process. On the other hand, we introduce in which way a security expert modifies a business process according to certain security requirements if no adequate soft-, hardware tools, enforcement procedures for security requirements, respectively case studies are available in the repositories. The procedure will be illustrated by looking on a procurement-related business process realizing anonymous requests for tenders.

The paper is structured as follows: In section 2, we will give an overview of related approaches. Section 3 provides a survey of security requirements and corresponding business process elements. In section 4, the different perspectives to model business processes in MoSS$_{BP}$ are introduced. Thereafter, we will outline the architecture of the MoSS$_{BP}$-framework in section 5 and the process how to act according MoSS$_{BP}$ (section 6). Finally, the approach is clarified by means of the tender request example in section 7.

2 RELATED WORK

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, 3, 5, 19, 31, 35]). A more comprehensive approach, addressing also the potentially conflicting security requirements of separate companies which collaborate in a business process, is provided by Pfitzmann [26]. Aspects of task management in business processes is addressed by Hung and Karlapalem [21] 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.

The construction of open and secure electronic market places is addressed by SEMPER (Secure Electronic Marketplace for Europe, [23]) which is particularly focussed on 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, [29]) has an even broader view to security issues of electronic marketplaces. 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 5) while the support components to design and maintain activities based on the services belong to the repositories of layer 2.

By application of a special fuzzy logic, Thoben concentrates on using security analysis to evaluate threats and risks of workflow-based business systems [34]. This 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

The business processes to be handled by MoSS$_{BP}$ are combinations of so-called *business process elements*. According to [6, 32] 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.

Nevertheless, to adapt the business process elements better to the security requirements which they shall fulfill, we adjusted these categories. 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. Moreover, we refined the categories in order to get more specific element types relating more directly to certain security requirements. The refined categories are listed below:

- *Agents*:
  - *Executing agent*: An agent performing a certain task.
  - *Ordering agent*: An agent who instructs another agent to perform a task.
  - *Agent of record*: An agent who is instructed by another agent to perform a task.
- *Artifacts*:
  - *Procedure*: A description (e.g., an algorithm) how agents proceed in order to execute activities.
  - *End product*: Result of executing a business process (or parts of it, e.g., activities).
  - *Information* is represented by data. This kind of artifact 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 applied procedures, the used information resp. material, and the resulting end products.

Security of computer-based systems mostly concerns confidentiality, integrity, and availability aspects. Our approach, however, is centered on domain experts who often have only a 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

[1]Generally, in business process models the information flow is not specified explicitly. However, it is relevant to model certain security requirements.

security requirements / **business process elements**	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

for business processes. In [11, 16] we identified the security requirements listed in the columns of table 1[2].

Since in the example introduced in section 7 we mainly focus on an anonymous and authentic handling of tenders, we sketch only the definitions of the security requirements *anonymity* and *authenticity* in the following:

- *Anonymity*: The true identity of a security object is hidden and no one is able to uncover it.
- *Authenticity*: A security object is what it pretends to be. It, however, may be a copy of the original object.

In the remaining parts of this paper, we call the objects, security requirements concern with (i.e., agents, artifacts, and activities), as *security objects* and persons acting as intruders as *security subjects*.

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 [11, 12]). Moreover,

[2]The list is subject to changes since a new business process may call for new security requirements.

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 *anonymity* we have to distinguish the following characteristics of the objects involved in carrying out an anonymous tender handling:

- Against whom anonymity of the agent is required,
- in which actions anonymity of the agent is required,
- for which information, end product, procedure, resp. information flow in the actions of the business process anonymity of the agent is required.

Moreover, in a MoSS$_{BP}$-based process (cf. section 6) one has to estimate the need of protection of a security object. Our approach refers to the seven security levels corresponding to the evaluation assurance levels defined in the Common Criteria (CC, cf. [22]). 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. Level 1 shall be assigned to the confidentiality of information if the disclosure of them is awkward only, but has no malicious effects on the business process.

4 MoSS$_{BP}$-Perspectives

In general, a business process is described by a process model which contains information on the process characteristics relevant to the purpose of the business target. According to [6], a combination of the following perspectives produces an integrated, consistent, and complete view of a business process (cf. [13]):

- The *informational perspective* represents the information entities, their structuring and relationships between them. In our approach, we use UML class diagrams (cf. [4]).
- The *functional perspective* shows which activities (processes) are performed and which data flow occurs between these activities. The functional perspective only represents the flow of data within the system. In our approach, we use UML activity diagrams.
- The *dynamic perspective* represents for each information entity all possible states and state transitions which may occur within the life cycle of the information entity. In our approach, we use UML state chart diagrams.
- The *organizational perspective* shows where and by whom activities are performed. This perspective corresponds to the organigram of an organization and to role models. In our approach, we use UML class diagrams.

Each perspective focuses on a very specific part of a business process. To achieve a better understanding and to analyze the whole business process, however, also an integrated view of all perspectives is necessary. Therefore, in addition to the four perspectives already mentioned, our framework supports a fifth perspective:

- The *business process perspective* offers an abstract and integrated view of the business process. In general, it is on a higher abstraction level in order to offer the domain expert an understandable image of the business process. Thus, the analysis of the security requirements, the business process should fulfill is made easier. The business process perspective is similar to the functional perspective but less detailed. Additionally, it refers to the informational perspective and to the organizational perspective. It describes the assignment of the activities to departments by using the UML-construct swimlane (cf. figure 3).

In order to provide business processes with safeguards, in general, more than one perspective has to be discussed. For instance, if one changes a tender handling process with authentic tenders from traditional to electronic execution, digital signatures have to be introduced (cf. section 7). In consequence, modifications have to be performed in the following perspectives:

- Functional perspective: A further activity `Check digital signature` is added, describing the check that a digital signature is valid.
- Informational perspective: An additional class for public key certificates is added.
- Dynamic perspective: Since the validity of certificates is time restricted, the provableness of digital signatures is only guaranteed for a certain amount of time. The life cycle of certificates must be specified.

In addition, to manage the problem of certificate's validity the archiving process has to be modified, too (e.g. in the organizational perspective a certification manager has to be added).

5 MoSS$_{BP}$: A FRAMEWORK TO SUPPORT SECURITY OF BUSINESS PROCESSES

If a traditional business process is refined to a computer-based process, the domain expert demands that the refined process fulfills all the security requirements of the original (e.g., a digital signature should be as legally binding as a traditional "paper and pen"-signature). Due to his often rudimentary knowledge of the security requirements to be fulfilled by an electronic business process, a domain expert needs help in the exact definition of the security requirements. For example, 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.

The project MoSS$_{BP}$ [11] 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. As outlined in section 4, MoSS$_{BP}$ uses UML diagrams 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 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. [13]):

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 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".

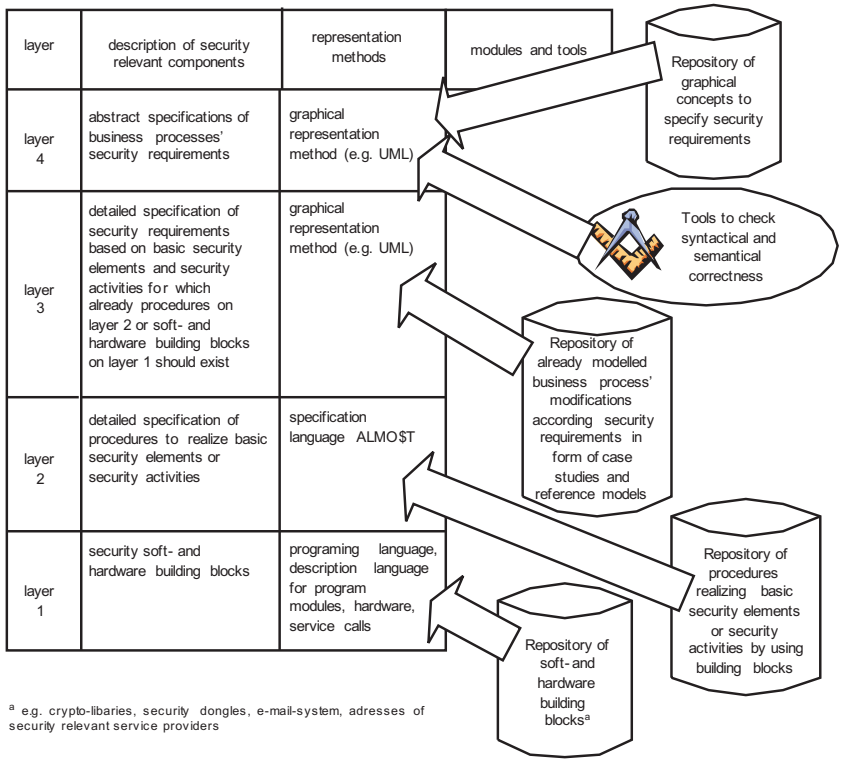

Figure 1: MoSS$_{BP}$-architecture

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 and a procedure checking the originality of the contract partner's public key by contacting a trusted third party acting as a certification authority). To describe these procedures and their combinations 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. [28]) which was developed in cooperation with the project COPS [29].

Layer 1: Soft- and hardware building blocks to realize the procedures of layer 2 are collected in this layer. Examples are a hardware encryption

and decryption chip resp. a distributed application enabling communication with certification authorities.

The proceeding through the layers in order to analyze a business process for necessary security requirements is introduced in the following.

6 MoSS$_{BP}$-PROCESS

To execute a computer-based business process, one has to create a model of the business process, to identify the necessary security requirements, and to search the repositories for corresponding building blocks, procedures, or case studies. If suitable elements are not available, they have to be designed and added to the repositories. If that is not possible, the domain expert has either to relax the security requirements or the business process cannot be carried out at all. The overall MoSS$_{BP}$-process is outlined in figure 2. Here, by different shades of gray we describe which layer of the MoSS$_{BP}$-architecture is addressed by a particular process step. The process consists of four phases which are introduced as follows:

Phase 1: The domain expert has to identify security requirements and assign them to the business process elements. Afterwards, a semantic check takes place testing if the assignments are correct (layer 4). For each wrong assignment an explanation is delivered and the domain expert has to decide what he really meant. If, for example, the security requirement *anonymity* is falsely assigned to an *activity*, the semantic checker outputs the following information:

- Explanation of the term *anonymity*,
- information, which security requirements are relevant for the security objects of category *activity*,
- information, for which categories of security objects the security requirement *anonymity* may be relevant.

Based on this information, the domain expert can correct the wrong assignments.

Phase 2: In the next phase, one checks for each security object if for all security requirements assigned to the particular object a corresponding soft- or hardware building block (layer 1) resp. a procedure (layer 2) exists. According to the result of this check, the process proceeds as follows:

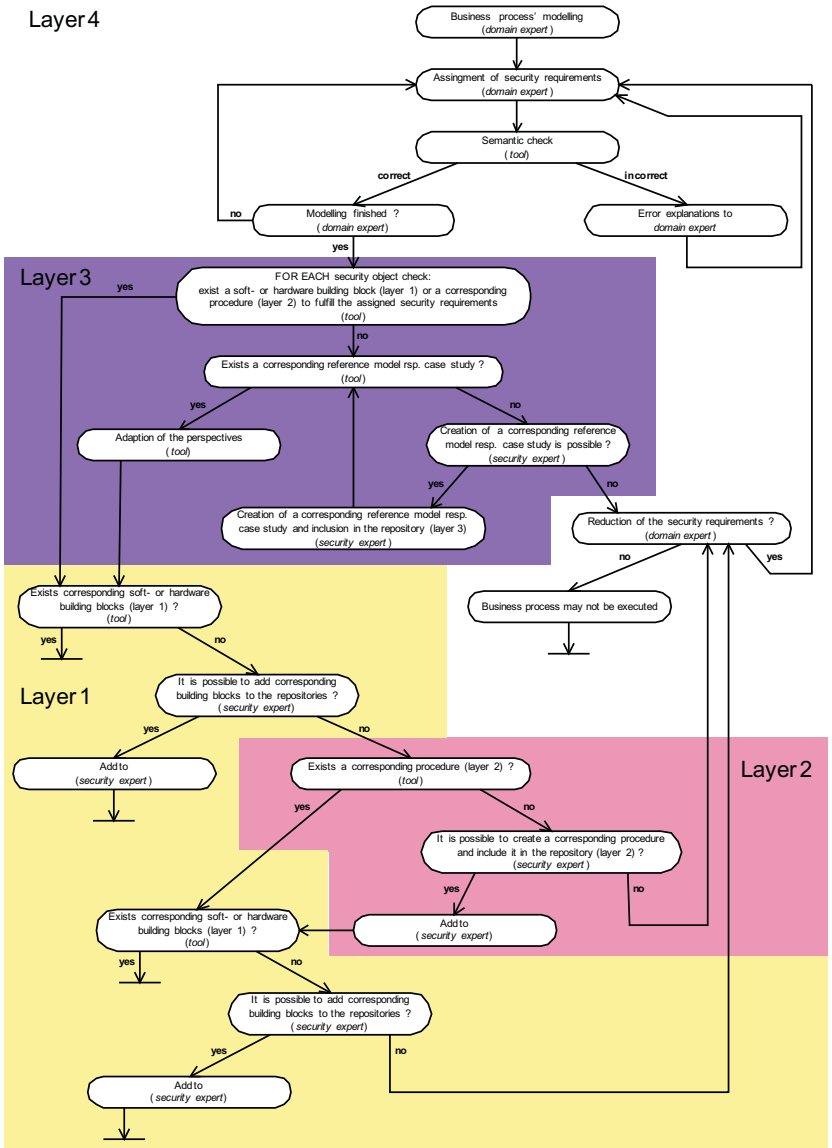

Figure 2: MoSS$_{BP}$-process

- If a building block realizing the security requirements exists (layer 1), the process finishes with a positive output for the particular security requirement.
- If a procedure exists (layer 2), one has to proceed to phase 3 checking if this procedure can be realized.
- If neither a building block nor a procedure exists for a security requirement, one checks if a corresponding case study is available in the repository at layer 3. If a case study fits, the perspectives of the business process model must be adopted to fulfill it. Otherwise, a security expert is notified in order to create a case study and to include it into the repository at layer 3. If it is not possible to design a case study, the domain expert is informed. He must decide if the security requirements of the business process can be reduced. If a reduction is not acceptable, the business process cannot be executed. If the domain expert modifies the security requirements, the MoSS$_{BP}$-process has to step back to phase 1 performing the semantic check (layer 4).

The case studies of the repository at layer 3 are specified by means of the same UML diagram types as the business process models at layer 4 and therefore most domain experts should be able to understand them. A case study specifies for each MoSS$_{BP}$-perspective how to act while realizing certain security requirements assigned to a specific security object. It contains basic security elements and security activities realized by procedures on layer 2 or soft- and hardware building blocks on layer 1.

Phase 3: In this phase, one checks if each security activity and each basic security element of a selected or newly developed case study is realized by a soft- or hardware building block (layer 1) or a procedure (layer 2). If neither a soft- or hardware building block nor a procedure exists, the security expert is informed. He tries to procure or develop corresponding soft- or hardware building blocks realizing the particular element of the case study. If this fails, he tries to create an adequate procedure and add it to the repository at layer 2. If the creation is not possible, he notifies the domain expert who similarly to phase 2 either relaxes the security requirements and steps back to phase 1 or gives up the business process.

Phase 4: If a procedure is used to fulfill the security requirements or a security activity resp. a basic security element of a case study, one has to check which soft- and hardware building blocks are needed to realize the procedure. If not all building blocks are available at layer 1, the security expert is notified. If he cannot add them, the domain expert again has to relax the security activities or give up the business process.

To illustrate the MoSS$_{BP}$-process, in the next section we describe the integration of the security requirements *anonymity* and *authenticity* to a business process performing the handling of tenders in an electronic procurement transaction.

7 EXAMPLE

Business-to-business transactions between companies by means of electronic procurement (e-procurement) get more and more popular and, meanwhile, standards for the transactions exist. For instance, the OBI consortium issued a set of specifications for **O**pen **B**uying on the **I**nternet (OBI, [24]). In this standard, an architecture for electronic procurement of goods and a corresponding business-to-business model are defined. The architecture introduces a buying organization, selling organizations, a payment authority, and a requisitioner. In behalf of the buying organization, the requisitioner carries out orders at the selling organizations and the orders are paid by means of the payment authority. If the buying organization decides to procure a certain good, it provides the requisitioner with seller addresses. Thereafter the requisitioner sends requests for tenders to the sellers, receives tenders, decides about a winning seller based on the tenders, and sends an order to the winning seller. Thereafter the order is fulfilled and paid by means of the paying authority. The tenders and orders are compatible to the EDI standard [7]. An automated realization of an OBI-conform business process is introduced in [18] and a formal-based proof that it fulfills certain access control security requirements is described in [15].

Here, we concentrate on the request for and the delivery of tenders which are realized by a partial business process. This business process shall be modified in order to guarantee that the true identities of the buying organization and the requisitioner cannot be discovered by the selling organizations. In the first step of the corresponding MoSS$_{BP}$-based process, the domain expert creates a UML activity diagram modelling the partial business process as depicted in figure 3. The swimlanes describe the depart-

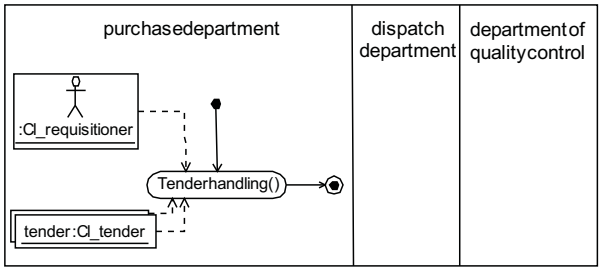

Figure 3: Tender Handling Process

ments which are responsible for the various activities. The requisitioner is an employee of the purchase department which is responsible for the tender handling. In later steps of the e-procurement, the dispatch department will be involved in the reception of the procured goods whereas the department of quality control will control their quality. In our example, we use three activity objects instantiated from the classes `Cl_process-start`, `Cl_tender-handling`, and `Cl_process-end`. The objects are linked by two solid arrows describing the order of the activities. The unnamed object of class `Cl_requisitioner` models the requisitioner who executes the activity `Tender handling`. It is linked with the activity by a dotted arrow of class `execute`. Moreover, the business process deals with tenders which are represented by artifact objects of the class `CL_tender`. The corresponding dotted arrows depict that the tenders are used in the activity `Tender handling`. For simplicity, we model only two tenders being sent from two different selling organizations.

In the following subsections, we will outline the execution of the phases of the MoSS$_{BP}$-process for the security requirements *anonymity* and *authenticity*.

7.1 Phase 1

In this phase, the domain expert has to identify the security requirements of the business process tender handling and we assume that he decides to enforce the requirements *anonymity* and *authenticity*. Moreover, the domain expert has to rate the seriousness of these two requirements and to assign the corresponding security levels (cf. section 3) which are modelled as attributes of the security requirement objects. The seriousness of the tender's authenticity is estimated as low, since a violation may only cost

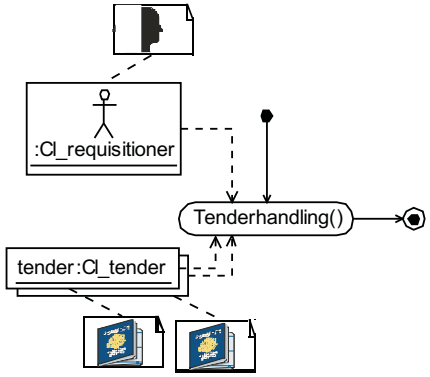

Figure 4: Secure Tender Handling Process (1)

working time but does not damage the selling organization, and the security level 2 is assigned to the attribute `authenticity security level` of the object `tender`. For the anonymity of the buying organization, however, the seriousness is estimated as high, since, for example, the relation with other suppliers of the buying organization may suffer if they get notice of the tender request. Therefore, the security level 6 is assigned to the attribute `anonymity security level` of the unnamed object of class `Cl_requisitioner`.

Figure 4 depicts the business process model after the assignment of the security requirements. Each security requirement is modelled by a node linked to the security object. It is represented by a rectangle with a dog-eard corner including the icon representing the specific security requirement. The passport symbol represents *authenticity*. The shadowy head represents *anonymity*.

After the assignment of the security requirements, a semantic check is performed stating that the assignment of the authenticity object to the `tender` is correct (cf. column 'authenticity' and row 'information' in table 1). The anonymity of the activity `Tender handling`, however, is incorrect (cf. column 'anonymity' and row 'activity' in table 1) and the domain expert is supplied by the following information:

- Explanation of the term *anonymity*,
- information, for which business process elements a correlation with anonymity is possible,
- information, for which security requirements a correlation to the busi-

ness process element *activity* is possible.

The domain expert recognizes that anonymity of the agent is what he really needs and models the process accordingly. Again a semantic check is carried out and the tool notifies the domain expert that more information how to relate the security requirement *anonymity* of the agent to the objects of the business process is needed (cf. section 3):

- Against whom anonymity of the agent is required,
- in which actions anonymity of the agent is required,
- for which information, end product, procedure, resp. information flow in the business process actions *anonymity* of the agent is required.

The domain expert decides to assign the following objects to the *anonymity* of the agent:

- Anonymity against the seller,
- anonymity based on the activity `Tender handling`,
- anonymity based on the whole information transmitted between the buying and the selling organization during the execution of the activity `Tender handling`. In fact, this are the information objects `customer_request` and `tender`.

At the end of phase 1, the domain expert models the secure tender handling process by the model sketched in figure 5[3]. In the following two subsections we will introduce the remaining phases of the MoSS$_{BP}$-process example for the security requirement *anonymity* whereas *authenticity* is shortly sketched afterwards.

7.2 Phase 2 of anonymity

At first, the domain expert checks if for each security object of the tender handling process a soft- or hardware building block (layer 1), a corresponding procedure (layer 2), or a case study (layer 3) exists fulfilling the security requirement *anonymity*. We assume, that suitable elements are currently not contained in the MoSS$_{BP}$-repositories. Thus, the domain expert has

[3]For the sake of clarity only one `tender`, one `customer_request`, and one unnamed object of class `Cl_seller` are represented.

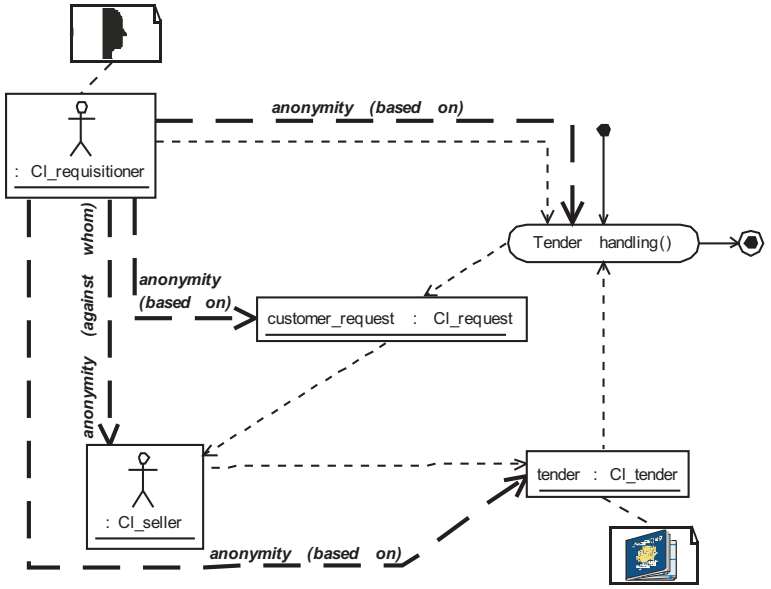

Figure 5: Secure Tender Handling Process (2)

to consult a security expert in order to create a case study for anonymous electronic tender handling. Before the security expert can decide if it is possible to develop a case study and on which techniques the case study should be based, he has to look on the relations between the involved objects since these relations may be the reason for vulnerabilities leading to the disclosure of identities.

In particular, the anonymity of a security object relates to information resp. to actions. If it relates to information, it has a direct link with the identity of the information creator. Of course, this link can be utilized to disclose the creator. If the anonymity relates to actions, it becomes manifest in interactions with the environment during the execution of the actions which again can be exploited for attacks. Interactions with the environment always takes place by participating in a communication (i.e., by sending or receiving messages). With respect to the anonymity of a security object concerning creation, sending and receiving of data, we can distinguish between the protocol control information of a message (e.g., an e-mail header), user data, and documentation data (e.g., a declaration who is the creator of the user data).

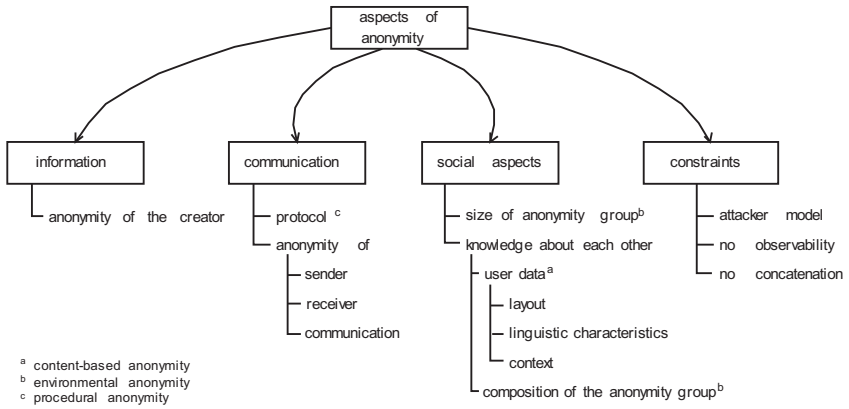

Figure 6: Aspects of anonymity

The various relations between the objects cause a whole spectrum of different anonymity attacks. Gavish and Gerdes (cf. [10]) differentiate three anonymity types which have to be considered in order to prevent disclosures:

Environmental anonymity: A person who wants to act anonymously has to ensure that he is not disclosed by people of his direct environment. In our example, the buying organization has to guarantee that a potential intruder — which could also be an employee — must not be able to observe the activities of the purchase department.

Content-based anonymity: A user acting anonymously has to guarantee that the user data created by himself, do not disclose the own identity. Therefore the identity of the buying organization must not be included in the user and documentation data of a request for tenders. Moreover, it has to be impossible that the identity can be deduced from the user or documentation data.

Procedural anonymity: If a user who wants to act anonymously, communicates electronically, the telecommunication protocol realizing the communication must not contain identity information in the protocol control information. Thus, the protocol realizing the request for tenders (i.e., the HTTP-protocol according to the OBI-standard [24]) must not contain data allowing the deduction of the buying organization's identity.

Another classification of the anonymity aspects was provided by Rubert [30]. As sketched in figure 6, like Gavish and Gerdes this model distinguishes between information and communication aspects. Moreover, it refers to social aspects which are based on so-called anonymity groups. An anonymity group hides an anonymously acting agent in a group of other agents. Intruders from outside the group must not be able to recognize which agent of the group is executing a certain action. Finally, the classification lists constraints which may influence the seriousness of the attacks. The constraints depend on an attacker model specifying the characteristics of a potential attacker (e.g., his computer capacity).

After analyzing the relations between the involved objects and detecting the corresponding vulnerabilities for the anonymity of the buying organization, the security expert has to consider the basic concepts to protect agent's anonymity:

- Encryption of messages or message headers,
- removal of sender identifications,
- randomized transmissions which can be used to prevent traffic analysis of communications by comparing the lengths of messages or checking chronological orders,
- creation of dummy messages in order to prevent traffic analysis, too,
- broadcasting of messages.

Techniques to protect anonymity are based on these basic concepts. Below, we will outline some techniques to guarantee anonymity (cf., e.g, [8, 9, 27, 25]):

Techniques to protect the receiver anonymity: Here, we can apply broadcasting of messages which do not contain the receiver's identity. This method, however, is not scalable to large numbers of potential receivers (e.g., the Internet). Nevertheless, it can be used for relatively small anonymity groups where the message is sent to the group and a copy is delivered to every group member.

Techniques to protect the sender anonymity: In this case, we can use the following basic concepts:

- Sending dummy messages in order to prevent traffic analysis which may exploited to detect the transmitter of the message.

- Removal of the sender identification which may be performed by so-called anonymity servers. The sender transmits a message to the anonymity server which forwards it without the sender identity of the receiver. Moreover, we call special anonymity servers which are able to send the replies to the original sender, as mediators.
- Encryption of messages in order to prevent man-in-the-middle attacks on the sender identity.

Techniques to protect the anonymity of the interconnection:
In this case, only the communication partners should know the existence of a data exchange between the communication partners. To realize this technique, one has to combine various basic concepts. In particular, the sender identification has to be removed since, otherwise, the identity of the sender may be disclosed using protocol control information during the transmission. Moreover, the disclosure of the sender identity is possible by scrutinizing the user data. Therefore, before transmitting the data to the anonymity server, the sender has to end-to-end encrypt the user data. Furthermore, an attack is possible by performing a lexical analysis. Therefore, besides the encryption on the section between the sender and the anonymity server, at least an additional encryption should take place between the last router of the route and the receiver. Finally, one can perform a traffic analysis by observing the incoming and outgoing links of a router resp. anonymity server which can be used conclude the whole route of a message. To prevent traffic analysis, one can either use dummy messages or randomized transmission. Examples for techniques realizing the anonymity of the interconnection comprise MIX [25] and Onion-Routing [33].

Unfortunately, the techniques outlined above do not address disclosures based on content-based anonymity. Thus, the security expert cannot prevent that the buying organization (i.e., the requisitioner) discloses the own identity by including vulnerable data in requests for tenders. For instance, user data created by a word processing tool may contain additional information describing the creator of the data. If it is possible, the administrator of the word processor should turn off such features. Therefore, the example analysis leads also to modifications of the software installation processes in a company.

In our example, the anonymity of the buying organization against the selling organization in a request for tenders is based on the data transmitted

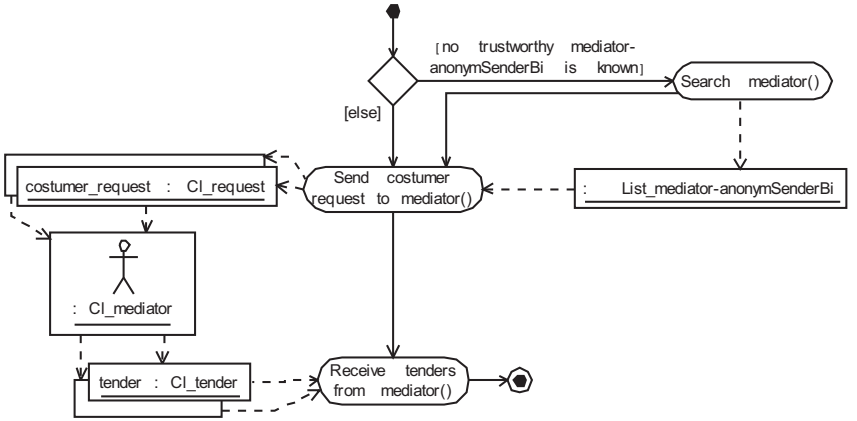

Figure 7: Case Study 'Mediator'

between the requisitioner and the sellers in the activity modelled by the object `Tender handling`. The transferred information is described by the information objects `customer_request` and `tender`.

In order to decide which techniques should be used to protect the buying organization's anonymity, an attacker model is necessary. In our example, potential attackers are current suppliers who want to break the anonymity of tender requests in order to be informed about potential competitors. Moreover, the receivers of the tender requests are also potential intruders since by knowing the true identities of their customers they can try to influence the order decision. Based on this classification, the domain and security experts agree that the likelihood of anonymity attacks by the current supplier or a requested seller based on wiretapping is very low. Therefore, only countermeasures to protect the anonymity of the buyer against analysis of received tender requests are installed. In consequence, the case study developed by the security expert comprises the following countermeasures:

1. The requisitioner has to prevent hints of the buying organization's identity in the user and documentation data of the requests for tenders. Since for this problem no technical support is available, the system administration process has to be modified accordingly.

2. The sender identification has to be removed from the protocol control information by an anonymity server. Since the requisitioner needs to receive the answers of his request, a trustworthy mediator must be con-

sulted who removes the identification data of the buying organization from the protocol control information of the tender request messages. Moreover, the mediator forwards the received tenders to the buying organization.

Figure 7 depicts the functional perspective of the case study. If no trustworthy mediator is yet known, a lookup of suitable mediators is started and the access information of the detected mediators are stored in a list modelled by an unnamed object of class `List_mediator-anonymSenderBi`. If the requisitioner wants to request a tender, he selects a mediator, which is specified by the unnamed object of the class `Cl_mediator`, from the list and transmits the request for tenders to this mediator. Afterwards, the mediator forwards the received tenders to the requisitioner.

7.3 Phases 3 and 4 of anonymity

In phase 3, the domain expert has to check for each security activity and for each basic security element of the case study if a soft- or hardware building block (layer 1) or a corresponding procedure (layer 2) exists. The case study of the security requirement *anonymity* contains no basic security elements and the only security activity is `Send costumer request to mediator`. For this security activity, we assume that no corresponding building blocks exist at the repository of layer 1. Nevertheless, it is very easy for the security expert to create a suitable building block which must only realize a simple service call to the mediator.

Since we do not use any procedures in phase 3, we can omit phase 4. Thus, by realizing the case study developed in phase 2 and applying the building block created in phase 3, we can modify the tender request process in order to fulfill the desired security requirement *anonymity*.

7.4 Phases 2 to 4 of authenticity of information

To ensure that a security object is what it pretends to be, nobody must be able modify the object after its creation. A useful method to guarantee authenticity is public key cryptography. In our example, the tender has to be digitally signed with the private key of the seller. If the check of the digital signature fails, the tender is not authentic and must be removed by the requisitioner. In the repository of layer 3 an adequate case study is stored including the following perspectives:

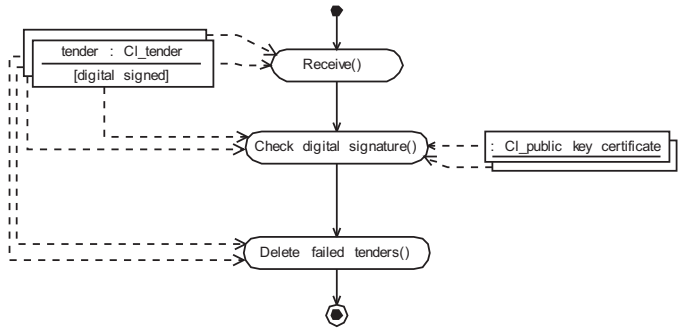

Figure 8: Case Study 'Authenticity of a document'

- In the *functional perspective* an activity `Check digital signature` of the relevant information and an activity `Delete failed tenders`, which will be executed if the check fails, are included.
- The *informational perspective* contains the entity `public key certificate`.

In the repository of layer 2 an ALMO$T procedure (cf. [28]) is stored specifying how to realize the activity `Check digital signature`. In this procedure among other things the construct `RSA.decrypt (pKeyring.get (seller).tender)` is used. It guarantees that the document `tender` has to be decrypted by the method `RSA` using the public key of `seller` which is stored in `pKeyring`. Appropriate software tools to realize the decryption of a document using the method RSA are offered in the repository of layer 1.

Figure 8 shows the case study describing how to modify the functional perspective of the tender handling process according to the security requirement *authenticity* of the tenders. Figure 9 depicts the modified functional perspective of the tender handling process regarding the security requirements *anonymity* and *authenticity*. After sending the costumer request to potential sellers, receiving their tenders via a mediator, and checking the digital signatures of the tenders, the tenders are evaluated and a winning seller is chosen (activity `Decision about tenders`).

8 CONCLUSION

We introduced MOSS$_{BP}$ to facilitate modifications of business processes in order to fulfill security requirements in Internet-based distributed e-

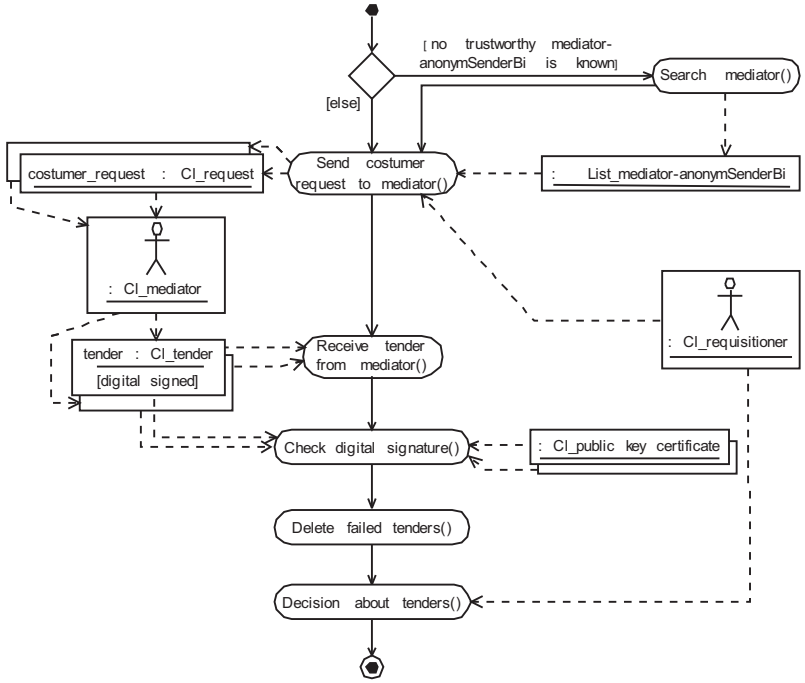

Figure 9: Secure Tender Handling Process (3)

commerce systems. In particular, the approach concentrates on bringing the domain experts which have a deep view into the business processes and the security experts who have a broad knowledge of the security mechanisms together. The modifications of the business processes are performed in a process of four steps.

Unfortunately, however, the modifications may cause other security holes since the added business process components may interfere with each other and with existing business process elements. To check this, one has to perform a security analysis of the business process. A tool for security analysis based on UML object diagrams and on graphic diagram rewriting was introduced in [14, 17]. It is based on the Common Criteria standard [22] and performs asset valuation, vulnerability and threat identification, risk analysis, and countermeasure selection in a highly automated fashion.

Moreover, we can also use the security analysis tool to facilitate the different phases of the MOSS_{BP}-process. Up to now, only phase 1 is supported [16]. An extension facilitating also the other phases is under development.

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

[3] 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.

[4] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, 1999.

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

[6] B. Curtis, M. I. Kellner, and J. Over. Process modeling. *Communications of the ACM*, 35(9):75–90, 1992.

[7] Data Interchange Standards Association. *X12 Standard*, release 4050 edition, Dec. 2001.

[8] T. Dinkhoff, V. Gruhn, T. Demuth, and A. Rieke. Anonym im World Wide Web? Janus — Schutz von Inhaltsanbietern im WWW. *Datenschutz und Datensicherheit*, 22(11):623–627, 1998. In German.

[9] H. Federrath and A. Pfitzmann. Neue Anonymitätstechniken: Eine vergleichende übersicht. *Datenschutz und Datensicherheit*, 22(11):628–632, 1998. In German.

[10] B. Gavish and J. Gerdes. Anonymous mechanisms in group decision support systems communication. *Decision Support Systems*, 23(4):297–328, 1998.

[11] G. Herrmann. *Verläßlichkeit von Geschäftsprozessen — Konzeptionelle Modellbildung und Realisierungsrahmen*. Logos Verlag, 2002. In German.

[12] G. Herrmann and G. Pernul. Towards Security Semantics in Workflow Management. In *Proceedings of the 31st Annual Hawaii International Conference on System Sciences (HICSS-31)*. IEEE Computer Society Press, Jan. 1998.

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

[14] P. 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, Dec. 2001. ACM SIGSAC, IEEE Computer Society Press.

[15] P. Herrmann. Formal Security Policy Verification of Distributed Component-Structured Software. In H. König, M. Heiner, and A. Wolisz, editors, *Proceedings of the 23rd IFIP International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'2003)*, LNCS, Berlin, Sept. 2003. Springer-Verlag.

[16] P. Herrmann and G. Herrmann. Security-Oriented Refinement of Business Processes. In *Proceedings of the 5th International Conference on Electronic Commerce Research (ICECR-5)*. ATSMA, IFIP, 2002.

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

[18] P. Herrmann, L. Wiebusch, and H. Krumm. State-Based Security Policy Enforcement in Component-Based E-Commerce Applications. In *Proceedings of the 2nd IFIP Conference on E-Commerce, E-Business & E-Government (I3E)*, pages 195–209, Lisbon, 2002. Kluwer Academic Publisher.

[19] 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.

[20] 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.

[21] P. C. Hung and K. 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.

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

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

[24] OBI Consortium. *OBI Technical Specifications — Open Buying on the Internet*, draft release v2.1 edition, 1999.

[25] A. Pfitzmann. *Dienstintegierende Kommunikationsnezte mit teilnehmerüberprüfbarem Datenschutz*. Number 234 in Informatik-Fachbereichte. Springer-Verlag, Berlin, 1990. in German.

[26] A. 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.

[27] T. Roessler. Anonymität im Internet. *Datenschutz und Datensicherheit*, 22(11):619–622, 1998. In German.

[28] A. Röhm, G. Herrmann, and G. 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.

[29] A. Röhm and G. Pernul. COPS: A Model and Infrastructure for Secure and Fair Electronic Markets. *Decision Support Systems Journal*, 29(4):343–355, 2000.

[30] M. Rubert. Anonymität als Sicherheitsmerkmal von Geschäftsprozessen. Diploma thesis, Department of Business Administration, University of Essen, 1999. In German.

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

[32] 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.

[33] P. F. Syverson, M. G. Reed, and D. M. Goldschlag. Onion Routing Access Configurations. In *DISCEX 2000: Proceedings of the DARPA Information Survivability Conference and Exposition*, volume 1, pages 34–40, Hilton Head, SC, Jan. 2000. IEEE Computer Society Press.

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

[35] 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.

Gaby Herrmann studied Computer Science at the University of Karlsruhe (diploma in 1991). Afterwards, she worked as a researcher first in the Computer Communication Systems Group and later in the Information Systems Group of the Business Department at the University of Duisburg-Essen (doctorate in 2001). Now she is executive director of the dean's office at the Business Department. Her research interests include security aspects of business processes and workflows.

Peter Herrmann studied Computer Science at the University of Karlsruhe (diploma in 1990). Since then he works as a researcher in the Computer Networks and Distributed Systems Group of the Computer Science Department at the University of Dortmund (doctorate in 1997). His research interests include security aspects of distributed component-structured software as well as formal-based development of distributed applications and hybrid technical systems.