Modelling Security Goals in Business Processes

Christian Wolter[1], Michael Menzel[2], Christoph Meinel[2]

[1]SAP Research, CEC Karlsruhe, {christian.wolter}@sap.com
[2]Hasso-Plattner-Institute, {michael.menzel,meinel}@hpi.uni-potsdam.de

Abstract: Various types of security goals, such as authentication or confidentiality, can be defined as policies for process-aware information systems, typically in a manual fashion. Therefore, we foster a model-driven transformation approach from modelled security goals in the context of process models to concrete security implementations. We argue that specific types of security goals may be expressed in a graphical fashion at the business process modelling level which in turn can be transformed into corresponding access control and security policies for process-aware information systems, for instance based on service-oriented architectures. In this paper we present security policy and policy constraint models. These models are projected onto general enterprise models and enterprise business processes in particular. We further discuss the suitability of this approach based on an example process and outline future work in order to derive security policy implementations out of the process models applicable for service-oriented architectures.

1 Introduction

In the domain of process-aware information systems, security configurations, legal compliance regulations, and risk assessments are all defined as security goals that decide upon the ways in which company assets are protected and the availability of resources is managed. A multitude of access control models, security protocols, security mechanisms, and related implementations have emerged over the last decades enforcing security goals. At the same time, IT-infrastructures have evolved into distributed and loosely coupled enterprise system landscapes, where a company's assets and resources are exposed by business services. Various business services are orchestrated by business domain experts and modelled as business processes with their own business logic in order to adopt faster to market changes and business demands.

While some attributes of business processes, such as control- and data-flows, are directly expressed in the context of the process models itself, related security goals are not expressed in this context. It is evident that business domain experts need to be able to define their security goals at a business process level, while the corresponding access control and security mechanisms need to be created and enforced at the service and resource level of the IT-infrastructure, for instance a service-oriented architecture (SOA). With respect to legal compliance issues and risk assessments, concepts are emerging to annotate process models with related bits of information [LJJ06, SGN07]. On the other hand, at the business process modelling level no currently available process modelling notation appears to have

the ability to capture security configuration related security goals, such as confidentiality, integrity , or attribute-based dynamic authorisation concepts (i.e. history-based separation of duty) [WS07].

Therefore, to leverage the definition of such security configurations into the scope of business process modelling and to support the enforcement of modelled security configurations at runtime, the dependencies between security goals, business processes, and the enterprise landscape need to be understood in order to foster a model-driven generation of security goal implementations for business processes. This approach is similar to the generation of business process execution descriptions, such as BPEL, where process descriptions, executable by a process engine, are generated from a mere visualisation notation [OvdAM+06]. Thus, modelled security goals must be transformed into enforceable security policy implementations, depending on the target environment.

In order to support this model-driven approach:

- We provide a detailed analysis of some basic security goals in terms of authorisation, confidentiality, and integrity as well as their related security constraints. Therefore, we provide a general security policy and various related security constraint models.
- We discuss the dependencies of our security policy and constraint models on the overall enterprise landscape. Thus, we apply our models to the enterprise model layers introduced in [SL06].
- In a next step we project our security policy model to a general business process model in order to directly specify security configurations in the context of business processes.
- As a pragmatic proof of concept we provide an example banking process with added dynamic authorisation and confidentiality annotations to demonstrate the feasibility of our approach.

The rest of this paper is organised as follows. In Section 2 we provide a detailed discussion about some basic security goals that were given in [PP02] and provide conceptual models for them. In Section 3 we outline the dependencies between security goals and the general enterprise architecture model and the business process model in particular. In Section 4 we compare our approach with some related work in the area of security ontologies and model-driven security approaches with respect to the expressible security goals. The last section discusses the potential benefits of our approach and outlines the automatic generation of policy implementations suitable for service-oriented architectures as some future work.

2 Security Goal Modelling Concepts

The conceptual security models must reveal all security aspects in an enterprise landscape and the relationship among affected entities. Therefore, our security model describes basic security goals and outlines the relationship to specific security attributes.

2.1 Specifying Security Goals

The abstract concept of security can be defined precisely by specifying a set of security goals [PP02]. Although these goals can be further specialised, subdivided or combined, we will focus on the basic goals in this paper solely:

1. *Confidentiality* provides protection against the unauthorised notice of stored, processed, or transferred information.
2. *Integrity* ensures the properness (intactness, correctness, and completeness) of information (data integrity) and the correct functioning of a system (system integrity) respectively. Transferred, processed, or stored data must not be modified with proper rights and - in economic terms - modifications must correspond to business values and expectations. A system must act in an expected and proper way at each point in time.
3. *Authentication* ensures the credibility of information - such as a claimed identity - by confirming this information as authentic.
4. *Authorisation* is the process of granting rights to participants to perform an interaction, for instance to access a resource.
5. *Traceability and Auditing* provide verifiability regarding all performed actions in an information processing system. This can be related to simple logging mechanisms, but also to monitoring as real-time auditing e.g. in intrusion detection systems.
6. *Availability* ensures that data, resources and services, which are needed for the proper functioning of a system, are available at each point in time regarding the requested quality of service.

These goals can be related to various entities of a process-aware information system. These relations among security goals and affected entities are typically described by *Constraints* that are composed in a security *Policy* as indicated by Figure 1.

The basic entity in such a model is an *Object*. We define an object as an entity that is capable to participate in an *Interaction* with other objects. This interaction always leads to an *Effect*, which can comprise the provision of information or the change of state in a system. The effect can, but does not need to be related to the object that initiated the interaction. For example, one object could be an application and another object could be a resource, such as a file. The process of accessing this file would be the interaction resulting in the effect that data in the file is changed or some information is returned to the application.

Each object is related to a set of attributes describing its meta information. For instance, if the object represents a user, attributes, such as name, email address, age, etc. will be assigned. Altogether, policy constraints always refer to a set of objects, a particular set of objects' attributes, and optionally a set of interactions and effects that are related to the objects. Based on these relations, specific constraints for particular security goals can

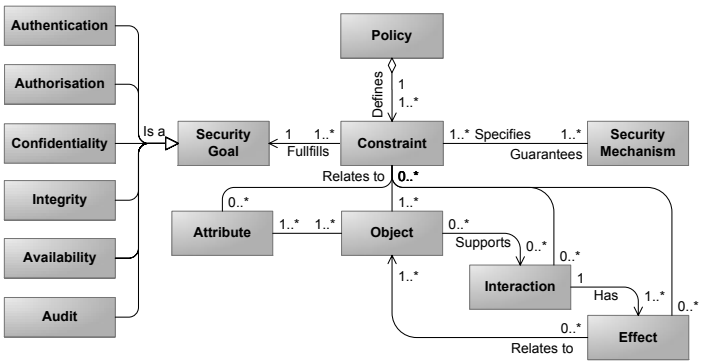

Figure 1: Security Policy Model

be defined. These specific constraints define requirements for associations between the entities with regard to the particular security goals. In the course of this paper four of six basic security goals are modelled and described subsequently. Namely the security goals authorisation, authentication, integrity, and confidentiality. As shown in Figure 1, constraints specify security mechanisms that enforce or guarantee the defined constraint. For instance, a confidentiality policy usually specifies an algorithm (e.g. DES) that must be used to guarantee this requirement.

2.2 Security Mechanisms

In our model a *Security Mechanism* is designed to characterise techniques that are used to enforce security constraints (cf. Figure 2). In general, these mechanisms can be classified as algorithms (e.g. DES) or protocols (e.g. WS-Security). The dependencies between these entities and their relationship to *Interaction* and *Effect* are not visualised in Figure 2. However, it provides the foundation to specify a comprehensive ontology for security mechanisms. Besides security mechanisms, a *Credential* represents another important entity in our model that subsumes evidences used by security mechanisms. A detailed classification of security credentials was presented by Denker *et al.* [DKF+03]. In this work they introduced an ontology that divides credentials in simple credentials (e.g. key, login, certificate) and composed credentials (e.g. Smart Card, SAML, WS-Security Token) that contain a set of simple credentials.

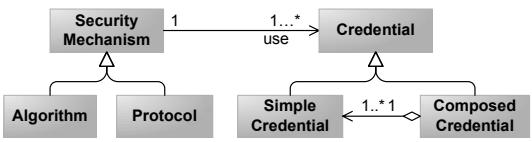

Figure 2: Security Mechanisms Model

2.3 Security Constraint Models

Based on the given security policy model (cf. Figure 1), we define specific types of *Constraints*, each guaranteeing one of the security goals listed above. We will elaborate upon the security goals *Authorisation*, *Authentication*, *Integrity*, and *Confidentiality* during the course of this paper, since these goals have the most significant effect on the modelling of business processes due to their potential relationship to business roles. Boxes with a dashed border in the model refer to entities defined in the aforementioned policy and security mechanism models. Each constraint is related to a specific set of entities and define rules restricting particular associations between those entities. These rules must be enforced by security mechanisms and are visualised in our model using dashed arrows pointing to the restricted associations.

2.3.1 Authorisation Constraint

A broad range of access control models have been developed in the last decades, defining access control constraints based on particular security information such as the user's role (RBAC [SC96]) or the user's team affiliation (TBAC [TS97]). Since all these pieces of information can be considered as attributes of involved objects, the attribute-based access control model (ABAC) can be seen as the most comprehensive access control model, as described in [bSH06].

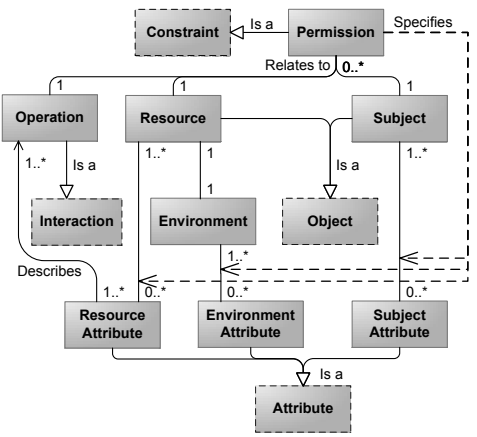

Figure 3: Authorisation Constraint Model

In general, there are three entities involved in an access control decision: The subject that wants to access a resource, the resource itself, and an operation that can be performed on this resource. Subject and resource map to objects in our basic constraint model, while operation specifies the interaction. According to ABAC, the access control decision is made based on subject attributes, resource attributes, and attributes of the resource's environment. Which attributes must be present, is specified by the policy constraint called *Permission* (cf. Figure 3).

2.3.2 Authentication Constraint

Authentication enables the credibility of information and is guaranteed by a credential that can be verified using security mechanisms. In our model, information is represented by a set of attributes and can be authenticated by one or more credentials. Since the credential must be assigned to a subject, it is a subject attribute as well. As shown in Figure 4, the authentication constraint *Claim* specifies the relationship between subject attributes and a set of credentials. For instance, the digital identity of a user can be authenticated by providing a name and password (i.e., the user's credential).

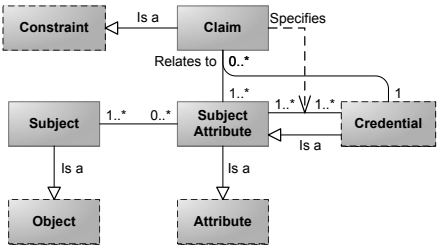

Figure 4: Authentication Constraint Model

2.3.3 Integrity Constraint

Integrity ensures that a system must act in an expected way at each point in time regarding transferred, processed, or stored data and the functioning of the whole system itself. In other words, an interaction must have exactly one effect that is specified by the integrity constraint. Such a constraint is called an *Assurance* in our model (cf. Figure 5). The security mechanisms that need to be defined in the assurance to guarantee the integrity depend on the concrete application. For example, WS-Policy [DLGea05] can be used as a policy language, specifying WS-Security [NKMHB06] assurances to enforce the integrity of SOAP communication.

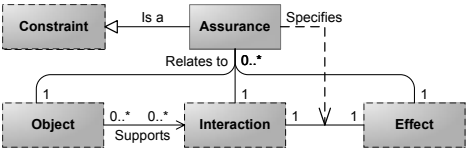

Figure 5: Integrity Constraint Model

2.3.4 Confidentiality Constraint

Since confidentiality ensures that authorised subjects are able to notice stored, processed, or transferred information solely, it is similar to authorisation, but more specific. The access to the information depends on a credential, the *Shared Secret*, which is shared by

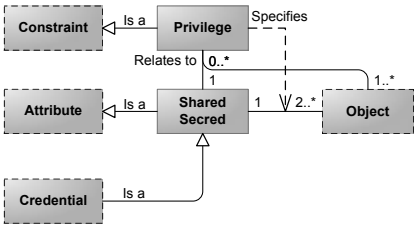

Figure 6: Confidentiality Constraint Model

all authorised subjects, as shown in Figure 6. Confidentiality is guaranteed by a security mechanism using the shared secret. In our model, the confidentiality constraint, named as *Privilege*, specifies the relationship between subject and shared secret.

3 Security Goals in the Business Process Context

In this section we outline how our security goal models are related to the various layers of a general enterprise model, such as proposed by Schreiter *et al.* in [SL06]. In particular, we examine the business process layer.

3.1 Enterprise Architecture Modelling

Enterprise architecture modelling is considered as "the structuring of an organisations processes, their related information systems, personnel, and organisational sub-units, so that they align with the organisations core goals and strategic direction" [SL06].

This modelling approach divides an enterprise into several conceptual layers. The *Integration Layer* is related to the enterprise application systems, their provisioned services, and the presentation of any digital content, e.g. through an enterprise portal. As proposed in [SL06], the *Integration Layer* could be further refined into three detailed layers describing the backend systems and integration applications, their provided services, and a presentation layer for data visualisation. The *Organisational Layer* contains information about the organisational structure of an enterprise in terms of departments, user roles, employees assigned to departments, and teams and roles, including role hierarchies.

The *Business Process Layer* addresses an enterprise's processes and defines dependencies between the business processes and the other layers. These dependencies give insight into the technical implementation aspects of business processes and related enterprise applications. For instance in the context of a service-oriented environment it defines which services must be invoked in order to execute a task defined in the business process. The process layer also defines dependencies to the *Organisational Layer* in terms of which organisational resources, i.e. employees acting in specific roles, are involved in the business process execution. A simplified example is given in Figure 7 visualising some objects of

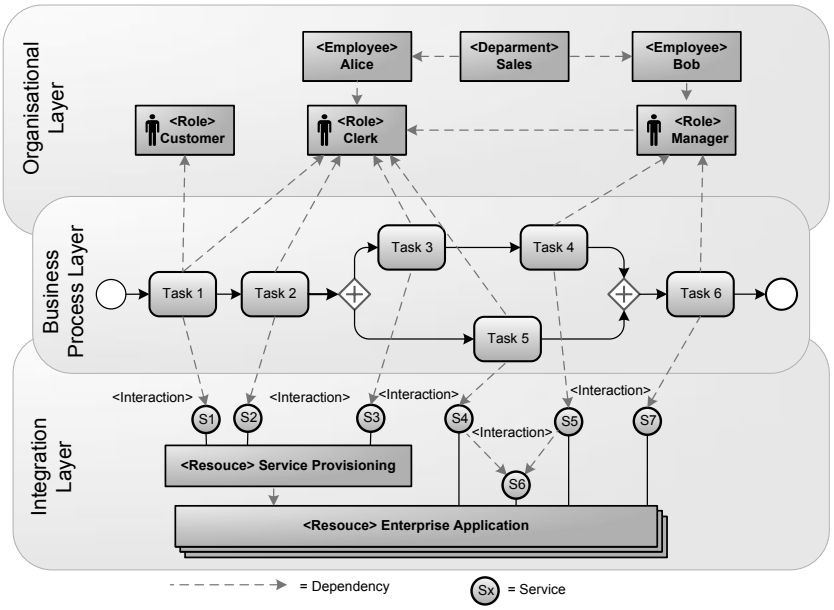

Figure 7: Simplified Enterprise Architecture Layers Example Inspired by [SL06]

the different layers and their dependencies.

The *Business Process Layer* acts as an abstraction layer for both the *Organisational Layer*, as well as the *Integration Layer* and hides details of the other layers behind the modelled processes. Nevertheless, due to the dependencies between the layers we are still able to derive technical and organisational details out of the business process layer. A typical technical example is the generation of BPEL code with concrete technical details about invokable web services out of the modelled business process [OvdAM+06].

Referring to the detailed discussion of the proposed enterprise architecture modelling approach given by Schreiter *et al.* the modelling of security goals is not considered. This leads backs to our initial discussion that security goals are not expressed in the context of business processes and are either defined on a very abstract management level or an implementation level without any relation to affected business processes. Therefore, we propose to include the definition of security goals directly into the *Business Process Layer* hiding the technical details. In this way, a shared base for security goal communication and administration can be provided.

In Section 2 we discussed the concepts of four basic security goals. We argued that security goals and their constraints are related to objects and object attributes, such as subject attributes, resource attributes, and environmental attributes. Figure 8 depicts where the necessary objects and object attributes can be found in an enterprise model, for instance needed to express an authorisation constraint and it also depicts dependencies between various enterprise architecture entities.

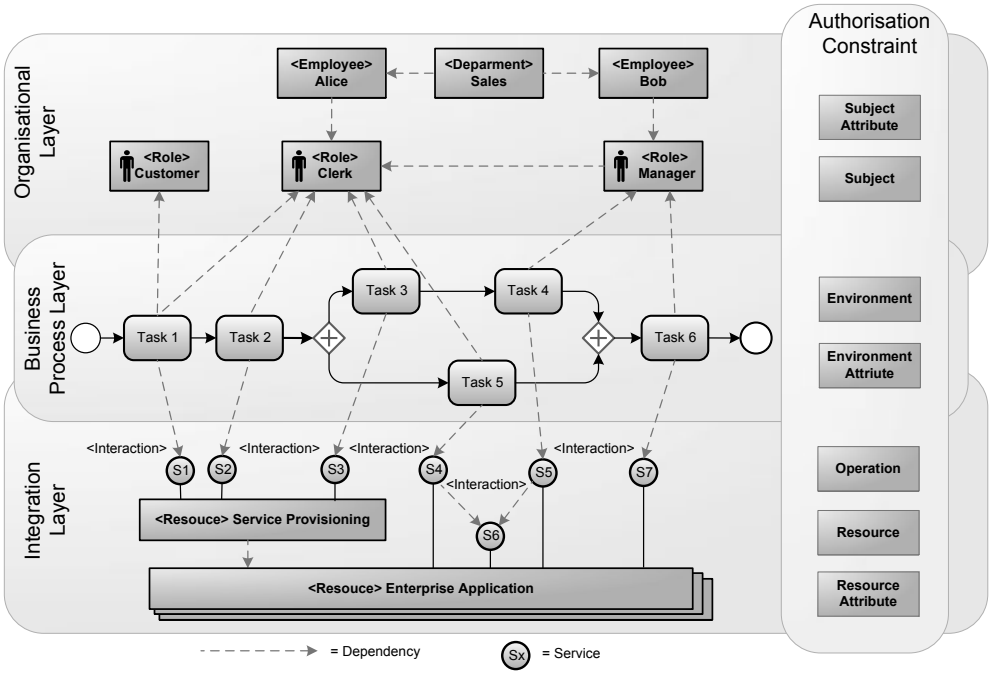

Figure 8: Mapping of Authorisation Constraint Entities

According to [SL06] a dependency between enterprise architecture entities is defined as: "The functionality of entity 1 depends on the proper functionality of entity 2". For instance, a dependency between a business task and a role, stating the fact that human interaction is necessary to fulfil this particular task.

Subject objects and *Subject Attributes* can be derived from the *Organisational Layer*. *Environment* objects and *Environment Attributes* can be derived from the process execution context in the *Business Process Layer*. *Resources*, their *Resource Attributes*, and related *Operations* can be found in the *Integration Layer*. We consider any kind of process-aware information system as a *Resource* object, while their provided services and functions are considered as *Operations*. The overall system state, such as CPU utilisation, storage capacity, and system configuration, like network address, are considered as *Resource Attributes*.

3.2 Extending the Business Process Layer

In the last section we projected entities from the *Authorisation* constraint model to the different enterprise model layers as an example. In research done by Russel *et al.* business process models where extended by a third perspective describing the resource perspective for processes [RvdAtHE05]. They considered resources as either human or non-human. Based on this, we analysed the model of BPMN [Gro06], a standardised process modelling

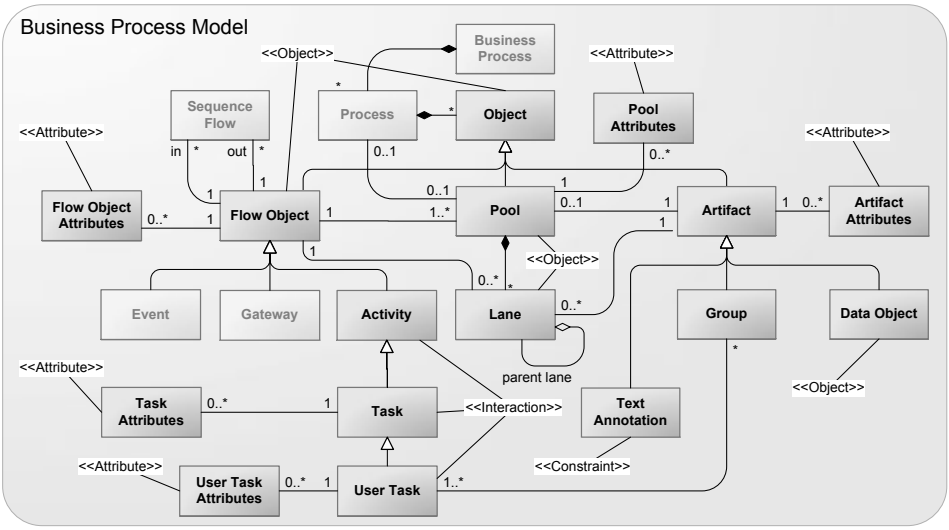

Figure 9: Application of Policy Stereotypes to Process Models

notation and mapped our security policy model entities to this model.

This mapping supports the idea that existing modelling notations provide flexible sets of attributes that can be applied to various entities of the process model, like pools, tasks, or data objects, in order to hold the information needed to express security goals and related constraints. We add stereotypes related to our general policy meta-model to the BPMN model elements creating a link between the entities of both models as shown in Figure 9.

3.3 Modelling of Security Goals in BPMN

In the last section we showed that relevant security policy information, such as objects, their interactions, and related object attribute information are available within the process model. Nevertheless, a suitable visualisation of security goals must be added to the process modelling notation with a specification added to describe the dependencies between the modelled security goals and the attributes of the process model entities.

For the visualisation of security goals, we propose a further enrichment of process modelling notations by security goal annotation elements. Some security goal annotations are given in an example process taken from the banking domain (cf. Figure 10). These annotations are based on an extended and semantically enhanced BPMN model discussed in [WS07]. The example process given in Figure 10 describes the necessary steps for opening an account for a customer.

The process contains several security goal annotations. Authorisation constraints and a role hierarchy are implicitly defined by the utilisation of the lane *Manager* and the nested

lane *Clerk*. Authorisation to access the *Customer Data* business object is implicitly given by the directed association between the tasks *Input Customer Data* and *Open Account in System*. The communication with the external credit bureau is marked as *encrypted* to achieve confidentiality and the contract send to the customer must be *signed* by an employee of the bank in order to guarantee non-repudiation and authentication. In addition, the process model contains groups of tasks that are annotated with Binding or Separation of Duty (4-Eyes-Principle) authorisation constraints. A further detailed discussion about the syntactical expressiveness of the security-related annotations in the context of authorisation constraints is provided in [WS07].

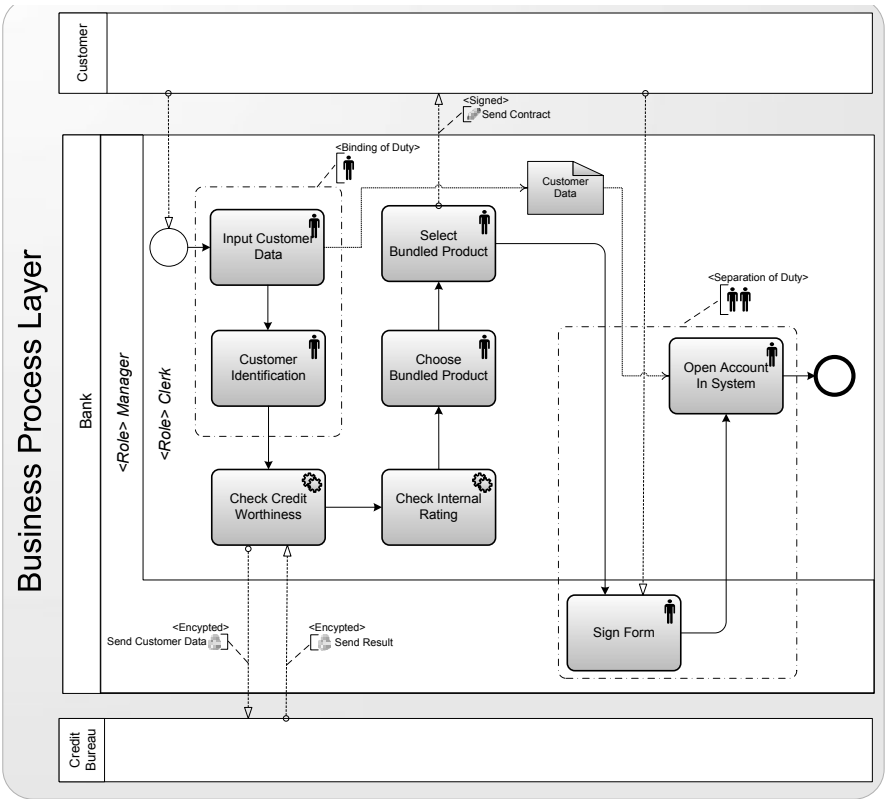

Figure 10: Business Process with Modelled Security Goals

4 Related Work

The domain of model-driven security in the context of business processes is an emerging research area. The need to support the application scenario and hypothesise the related security policies for the affected services on an abstract level is discussed in [TIN04]. We

extended this suggestion by defining security configuration requirements in the context of application scenarios captured on the business process layer. Schreiter *et al.* propose five architectural layers, ranging from high level business process scenarios to platform-specific services and applications and the dependencies between them omitting security related aspects [SL06].

Recent work done by Nagaratnam *et al.* [NNH+05] discusses an approach to overcome this shortage by expressing security requirements in the context of business processes and how to monitor and manage them on the different enterprise architecture levels. This intention, while similar to our concepts regarding the benefits of a modelling approach, does not provide a detailed analysis of security goals, their conceptual models, and their relationship to the business process related entities.

This has been addressed by Rodríguez *et al.* [RFMP06] by defining a meta-model that links security requirement stereotypes to activity elements of a business process and proposed graphical annotation elements to visually enrich the process model with related security requirements [RFMP07, EBA+07]. In contrast to our approach, we provide an additional analysis of the conceptual model of related security requirement stereotypes and a first mapping between the security models and business process models in order to apply the automated generation of security constraints and polices for the underlying enterprise architecture that implements the modelled business process. We analysed the characteristics of several security requirements and discussed their relationships to various access control models and their properties. A concept that also was recently discussed in [EBA+07] with a focus on access control, not considering mappings to other security requirements, such as integrity or confidentiality.

Our security model could be complemented by modelling concepts for compliance rules for business processes [SGN07] as described by Sadiq *et al.*. They propose model annotations with control tags that are mappable to the Formal Contract Language (FCL) focusing on the intended behaviour of the process model in the context of organisational compliance regulations. Their control tags cover order of event, data, and authorisation aspects, but they do not address how to actually derive enforceable compliance rules at runtime.

Enforcing authorisation constraint in workflows is addressed in [kHA99]. SecureFlow implements a Workflow Authorisation Model (WAM). Authorisations can be defined and enforced at runtime for users, roles, and workflow tasks. In contrast to our general security modelling approach they focus on authorisation constraints in centralised workflow management system, without considering other security requirements, such as confidentiality or integrity that are important in a service-oriented environment as well. These authorisation concepts are refined in [Hua05] by a semantic policy-based security framework for business processes identifying two levels of security for business processes. On the task or activity level, security concerns, such as non-repudiation, confidentiality, and data integrity are considered. On the process level, general compliance rules, such as required by Sarbanes-Oxley are defined, but a model connecting security and process aspects is not given.

Model-driven security and the automated generation of security enhanced software artefacts and security configurations has been a topic of interest in recent years. For instance

SecureUML [BDL03] is a model-driven security approach for process-oriented systems focusing on access control. Similar to SecureUML, Jürjens presented the UMLSec extension for UML [Jür02] in order to express security relevant information within a system specification diagram. One focus of UMLSec lies on the modelling of communication-based security goals, such as confidentiality, for software artefacts, while SecureUML describes desired state transitions and access control configurations for server-based applications, both do not leap for establishing the link between business processes and model-driven generation of related security requirements.

Security ontologies for service-oriented architectures is addressed in [KLK05, DKF+03], focusing on the definition of a general security ontology and semantics for web service security, such used protocols, encryption algorithms, and authentication credentials that can be applied in terms of annotations to service descriptions, such as WSDL files. Consideration of business processes as composite services and therefore applying these security annotations into the context of business processes, is not given.

5 Conclusion

Business process modelling represents a cornerstone of process-aware information systems. However, as stated in [TIN04] existing approaches address the specification of security configurations and related policies on a technological level rather at the business level. For example, this applies to the well defined OASIS standardised eXtensible Access Control Markup Language (XACML) and WS-Security. These XML-based notations are very expressive, but policy definition is cumbersome.

Process modelling notations would provide a suitable abstract perspective to specific security goals on a more accessible level, such as shown in [LJJ06, SGN07]. In a similar fashion we provide a modelling extension for security configurations, such as confidentiality, integrity, or authorisation, by providing a related abstract security goal specification. A model-driven approach addresses the difficulties to manage security mechanisms and their seamless integration into process-aware information systems. As a result, these security configurations would be consistent with the affected business processes and result in a decreased error-proneness. While this paper is focusing on the annotation of BPMN models, our annotations do not influence the control-flow and data-flow characteristics of business process models, our approach is more generic and thus could be applied to other process modelling notations, for instance XPDL or jPDL.

In this work, we presented an approach to express security goals at the business process level. The foundation constitutes our generic security model that specifies security goals, policies, and constraints based on a set of basic entities, such as *Objects*, *Attributes*, *Interactions*, and *Effects*. The strength of our model lies in its general description of security goals, and the abstraction from technical details. Thus, the provided models can be mapped to an arbitrary application or technical specification. For example it is possible to map our models to the technical specification of the WS-Security standard, applicable for service-oriented architectures.

Besides the potential mapping to a technical implementation, we addressed the issue of security goal specification, such as authorisation, confidentiality, or integrity in the context of business processes. It has been shown, that existing modelling notations provide the necessary elements or can be easily extended, by additional attributes, to express security goals based on our security policy model.

5.1 Future Work

We stated that our proposed security model is a promising approach to describe and implement security goals in different application domains. In order to fully enable the applicability of our concepts we have to investigate the combination of our security models with modelling concepts to describe legal compliance regulations, such as secure money transfer or auditing as described in [SGN07] and risk assessment concepts [LJJ06], thus enabling the specification, generation, and enforcement of compliant and secure business processes in a service-oriented environment. Another aspect is the involvement of human resources that perform activities in the context of business processes. Recent work [Sch04, NSIC07] shows that unforeseen events, such as illness or workload, could result in a temporary delegation of access rights and permissions from a delegator to a delegatee. We will investigate and extend our concepts in order to model and enforce role and task delegation constraints.

Therefore, existing projections, such as described in [MSSN04], must be extended to translate such fine-grained authorisation constraints into a concrete policy languages (i.e. WS-Policy or XACML [And05]).

A formalised security goal modelling description for business processes is necessary in order to specify and verify the consistency of modelled security goals and to guarantee a syntactic and semantic correct generation of security policy implementations. Some concepts have been recently discussed [TCG04] and we will investigate their applicability to our concepts.

In addition, the relationship between our security model and the OASIS reference model for SOA [MLM+06] must be investigated. Since the definition of the SOA reference model is based on the concept of interaction and effects, a straightforward application should be possible, proving that our models can be used to express security goals and related secure configurations suitable for service-oriented architectures. These topics will be addressed by future work.

References

[And05] Anne Anderson. Core and Hierarchical Role Based Access Control (RBAC) profile of XACML v2.0. OASIS Standard, 2005.

[BDL03] David Basin, Jürgen Doser, and Torsten Lodderstedt. Model Driven Security for Process-Oriented Systems. In *SACMAT '03: Proceedings of the 8th ACM symposium on Access control models and technologies*, pages 100–109, 2003.

[bSH06] Hai bo Shen and Fan Hong. An Attribute-Based Access Control Model for Web Services. In *pdcat*, pages 74–79. IEEE Computer Society, 2006.

[DKF+03] Grit Denker, Lalana Kagal, Timothy W. Finin, Massimo Paolucci, and Katia P. Sycara. Security for DAML Web Services: Annotation and Matchmaking. In *International Semantic Web Conference*, pages 335–350, 2003.

[DLGea05] Giovanni Della-Libera, Martin Gudgin, and et all. Web Services Security Policy Language (WS-SecurityPolicy). Public Draft Specification, Juli 2005.

[EBA+07] Christian Emig, Frank Brandt, Sebastian Abeck, Jurgen Biermann, and Heiko Klarl. An Access Control Metamodel for Web Service-Oriented Architecture. In *ICSEA '07: Proceedings of the International Conference on Software Engineering Advances (ICSEA 2007)*, page 57, Washington, DC, USA, 2007. IEEE Computer Society.

[Gro06] Object Management Group. Business Process Modeling Notation Specification. www.bpmn.org, 2006.

[Hua05] Dong Huang. Semantic Policy-based Security Framework for Business Processes. In *Proc. of the Semantic Web and Policy Workshop*, 2005.

[Jür02] Jan Jürjens. UMLsec: Extending UML for Secure Systems Development. In *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*, pages 412–425, 2002.

[kHA99] Wei kuang Huang and Vijayalakshmi Atluri. SecureFlow: A Secure Web-Enabled Workflow Management System. In *ACM Workshop on Role-Based Access Control*, pages 83–94, 1999.

[KLK05] Anya Kim, Jim Luo, and Myong H. Kang. Security Ontology for Annotating Resources. In *OTM Conferences (2)*, pages 1483–1499, 2005.

[LJJ06] James H. Lambert, Rachel K. Jennings, and Nilesh N. Joshi. Integration of risk identification with business process models. *Syst. Eng.*, 9(3):187–198, 2006.

[MLM+06] Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter Brown, and Rebekah Metz. Reference Model for Service Oriented Architecture 1.0. OASIS Committee Specification, February 2006.

[MSSN04] J. Mendling, M. Strembeck, G. Stermsek, and G. Neumann. An Approach to Extract RBAC Models from BPEL4WS Processes. In *Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)*, Modena, Italy, June 2004.

[NKMHB06] Anthony Nadalin, Chris Kaler, Ronald Monzillo, and Phillip Hallam-Baker. Web Services Security: SOAP Message Security 1.1. OASIS Standard Specification, February 2006.

[NNH+05] N. Nagaratnam, A. Nadalin, M. Hondo, M. McIntosh, and P. Austel. Business-driven application security: From Modeling to Managing Secure Applications. *IBM Systems Journal, Vol 44, No 4*, 2005.

[NSIC07] Tuan-Anh Nguyen, Linying Su, George Inman, and David W. Chadwick. Flexible and Manageable Delegation of Authority in RBAC. In *AINA Workshops (2)*, pages 453–458, 2007.

[OvdAM+06] Chun Ouyang, Wil M.P. van der Aalst, Dumas Marlon, ter Hofstede, and Arthur H.M. Translating BPMN to BPEL. In *BPM Center Report BPM-06-02*, 2006.

[PP02] Charles P. Pfleeger and Shari Lawrence Pfleeger. *Security in Computing*. Prentice Hall Professional Technical Reference, 2002.

[RFMP06] Alfonso Rodríguez, Eduardo Fernández-Medina, and Mario Piattini. Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes. In *TrustBus*, pages 51–61, 2006.

[RFMP07] Alfonso Rodríguez, Eduardo Fernández-Medina, and Mario Piattini. Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases. In *BPM*, pages 408–415, 2007.

[RvdAtHE05] N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In *In Proc. of 17th Int. Conf. on Advanced Information Systems Engineering (CAiSE05)*, 2005.

[SC96] Ravi S. Sandhu and Edward J. Coyne. Role-Based Access Control Models. *IEEE Computer*, 29:3847, 1996.

[Sch04] Andreas Schaad. An Extended Analysis of Delegating Obligations. In *DBSec*, pages 49–64, 2004.

[SGN07] Shazia Wasim Sadiq, Guido Governatori, and Kioumars Namiri. Modeling Control Objectives for Business Process Compliance. In *BPM*, pages 149–164, 2007.

[SL06] Torben Schreiter and Guido Laures. A Business Process-centered Approach for Modeling Enterprise Architectures. In *Proceedings of Methoden, Konzepte und Technologien für die Entwicklung von dienstebasierten Informationssystemen (EMISA)*, 2006.

[TCG04] Kaijun Tan, Jason Crampton, and Carl A. Gunter. The Consistency of Task-Based Authorization Constraints in Workflow Systems. In *CSFW*, pages 155–, 2004.

[TIN04] Michiaki Tatsubori, Takeshi Imamura, and Yuhichi Nakamura. Best-Practice Patterns and Tool Support for Configuring Secure Web Services Messaging. In *ICWS*, pages 244–251, 2004.

[TS97] Roshan K. Thomas and Ravi S. Sandhu. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Autorization Management. In *DBSec*, pages 166–181, 1997.

[WS07] Christian Wolter and Andreas Schaad. Modeling of Authorization Constraints in BPMN. In *BPM '07: Proceedings of the 5th International Conference on Business Process Management*, pages 64–80, 2007.