

Security Requirements Specification in Service-oriented Business Process Management

Michael Menzel
Hasso-Plattner-Institute
Prof.-Dr.-Helmert Str. 2-3
14482 Potsdam, Germany
michael.menzel@hpi.uni-potsdam.de

Ivonne Thomas
Hasso-Plattner-Institute
Prof.-Dr.-Helmert Str. 2-3
14482 Potsdam, Germany
ivonne.thomas@hpi.uni-potsdam.de

Christoph Meinel
Hasso-Plattner-Institute
Prof.-Dr.-Helmert Str. 2-3
14482 Potsdam, Germany
christoph.meinel@hpi.uni-potsdam.de

Abstract—Service-oriented Architectures deliver a flexible infrastructure to allow independently developed software components to communicate in a seamless manner. In the scope of organisational workflows, SOA provides a suitable foundation to execute business processes as an orchestration of multiple independent services. Along with the increased connectivity, the corresponding security risks rise exponentially. However, security requirements are usually defined on a technical level, rather than on an organisational level that would provide a comprehensive view on the participants, the assets and their relationships regarding security.

In this paper, we propose an approach to describe security requirements at the business process layer and their translation to concrete security configuration for service-based systems. We introduce security elements for business process modelling which allow to evaluate the trustworthiness of participants based on a rating of enterprise assets and to express security intentions such as confidentiality or integrity on an abstract level.

Our aim is to facilitate the generation of security configurations based on the modelled requirements. For this purpose, we foster a model-driven approach: Information at the modelling layer is gathered and translated to a domain-independent security model. Concrete protocols and security mechanisms are resolved based on a security pattern system that is introduced in the course of this paper.

I. INTRODUCTION

Business Process Modelling gains more and more attention, as it is the foundation to describe, standardise and optimise organisational workflows. A business process model is defined as a set of activities and execution constraints between these activities [1]. It can be used to describe complex interactions between business partners and to indicate related business requirements on an abstract level.

At the same time, IT-infrastructures evolved into distributed and loosely coupled enterprise system landscapes such as Service-oriented Architectures, which expose a company's assets and resources as business services. The independent nature of the services, with respect to operating systems and system architectures, facilitates a composition of different services. The cooperation with business partners demands the utilisation of services across organisational boundaries. In fact, the involvement of independent trust domains constitutes the key aspect regarding security in service-oriented architectures, since the seamless and straightforward integration of cross-organisational services conflicts with the need to secure and

control access. A broad range of security protocols and mechanisms has been specified to address this discrepancy in the scope of SOA. However, these security standards and mechanisms have their focus on the technological level. In order to enforce and guarantee security in an SOA it is not sufficient to protect single endpoints. A comprehensive understanding and evaluation of threats and associated risks is needed.

As we have described in previous work [2], business process modelling offers an appropriate layer to describe security requirements and to evaluate risks. We described an approach to integrate security goals and constraints in business process modelling and a model-driven transformation focusing on authorisation requirements in [3], [4]. Similar approaches exist, for example as presented by Rodríguez [5], that provide extensions for BPMN to express basic security requirements.

However, current security modelling approaches are mostly focused on authorisation, but are not feasible to enable a comprehensive verification of security properties. For instance, it is not sufficient to model a lock at the process layer as an intention to ensure confidentiality for a single connection, since it would not consider the flow of information in the process. In a complex process-driven SOA, information may be passed through multiple intermediaries until it is stored or processed by a service. Therefore, multiple parameters must be considered for the generation of consistent security configurations, such as the information's value, the trustworthiness of participants and the dependencies between modelled entities. We believe that the modelling should not just integrate abstract security intentions – it should include additional security meta information to rate entities facilitating a comprehensive view on security to model, evaluate and verify security requirements.

Based on modelled and verified security requirements, we intend to enable the generation of concrete security configurations for process-aware information systems. However, in the face of the variety of security specifications regarding SOA realised with Web Services, existing implementations come along with incompatibilities and multiple dependencies. Current model-driven approaches enhancing business process models with security intentions do not describe the consistent selection of appropriate security concepts.

Therefore, we foster a model-driven approach in which security intentions and ratings are annotated in business processes that can be translated to consistent security policies. In this paper we provide

- a compilation of organisational security aspects that emphasise the relationship between security threats, security goals and migration strategies in an organisational context.
- a security enhancement for business processes that provides properties and annotations to integrate the revealed organisational security concepts.
- a model-driven approach to generate policies. We introduce a security model to gather security information modelled at the process layer and describe the usage of security patterns to resolve appropriate security protocols.

This paper is structured as follows. Section 2 provides an overview about enterprise security concepts and based on this we introduce our approach to enhance business process modelling with security properties and annotations in the next Section. Section 4 introduces our domain-independent security model to gather security information from the modelling layer and introduces security patterns to resolve appropriate protocols and security mechanisms to enforce modelled security intentions. As a proof of concept, we present a mapping to the Axis2 security configuration in Section 5. In Section 6 we discuss and conclude our approach and outline some suggestions for future work, such as the integration of cross-organisational services.

II. ORGANISATIONAL SECURITY

The central aspect of security engineering is the management of risks that result from potential threats referring to business assets (e.g. information, tasks, etc.). To evaluate the impact of threats, assets must be evaluated to determine its overall importance in an enterprise. Based on the evaluation of enterprises assets, appropriate actions can be identified that face the threats and minimize the risks. Threats and countermeasures can be classified according to related security goals [6]. In this paper we will focus on threats related to the usage of identity information and enforcement of associated rights (*authentication, authorisation, trust*), transferred, processed or stored information (*data confidentiality and data integrity*) and the functioning of a service (*system integrity and availability*).

A. Authentication, Authorisation, Trust

Authentication ensures the credibility of identity information by verifying that a claimed identity is authentic, while *authorisation* is the process of granting rights to participants to perform a task, for instance to access a service. These goals presume a secure management and trustworthy provision of identity information. With regard to a Service-oriented Architecture, the underlying trust relationships must be considered, since the usage and provision of services might not be limited to one trust domain. Trust can be defined as “the extent to which one party is willing to depend on the other party in a given situation with a feeling of relative security, even though

negative consequences are possible.” (McKnight and Chervany [7]) To evaluate the trustworthiness of the authentication and authorisation process, it is important to analyse the underlying trust relationships that can be classified in *Organisational Trust* and *Identity Trust*.

Organisational Trust – Organisational Trust refers to the quality of the trust relationship between the participants of a SOA. When service consumer and service provider are located within the same trust domain, the registration, authentication and management of participants happens under the same administrative control and are, therefore, usually fully trusted. However, with regard to cross-organisational SOA involving services from different organisations, trust between the participants of a SOA is not given per default, but required to allow access to the services of a partner organisation. Models for identity management as Federated Identity Management establish cross-organisational trust by setting up federation agreements and contracts to extend the trust domain of an organisation to the federation. Having a federation or not, whenever organisational borders are crossed by a SOA, the question of whether the partner is trusted arises. Factors as past experience, the minimum trust settings for, for example, registration and authentication of users or the reputation of a company are important attributes to assess the trustworthiness of the potential business partner. Also, the kind of business relationship is an important factor. A B2B relationship is usually much more trustworthy than a B2C relationship due to contracts which manifest certain obligations and procedures of the business partners. All these factors make up the quality of the trust relationship. This quality can be quantified into a value, which we call *Organisational Trust Level* in the following. The assessment of the organisational trust level is in particular important with regard to authorisation and access control which is most often based on identity information. An organisation will only rely on identity information received from outside its own control, when it recognizes the source of information as trustworthy. Therefore, the trustworthiness of this source has to be assessed and put in relation with the damage that might be caused by trusting on malicious information.

Identity Trust – The identity of a subject is important for most systems in order to provide personalized service or to hold us liable in case anything bad happens. Therefore, reliable authentication mechanisms are required. They ensure the credibility of identity information which provide the foundation to perform access control. 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 [8]) or the user’s team affiliation (TBAC [9]). 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 [10]. Within the borders of one organisation, the organisational role of a subject is often the attribute of choice to perform access control, since it can be understood

as "A set of expectations and behaviours associated with a given position in a social system." [11] and therefore implicitly reflects the trust one can put into the correct behaviour of a subject. While this "role behaviour" is predictable within one organisation, it is hard to predict for a subject from a foreign organisation since role definitions are not or insufficiently known. Here, attributes as the affiliation to a company or a person's credit line are more meaningful. In either case, the provision of such trust-related attributes is required to build up trust in the identity of the user and its behaviour. This trust, which we call *identity trust*, is the concept behind all access control models.

Depending on the value of a transaction and the risk associated with it, a transaction has specific trust requirements, which are described as part of access control policies and lead to an expected identity trust level. To establish identity trust, trust-related information as specified in the access control policy is exchanged, often conveyed in security tokens, so called credentials. With regard to authentication, approaches exist [12] that describe how the quality of the authentication effects the trust level.

B. Data Confidentiality and Data Integrity

Confidentiality provides protection against the unauthorised notice of transferred, processed, or stored information, while *Data Integrity* ensures the properness (intactness, correctness, and completeness) of information. Since the enforcement of these security goals might involve the application of complex security mechanisms, there is always the trade-off between the desired level of security and performance. The required security level depends on the information's value that influence the implementation (type of protocols, algorithms, etc.) of these goals. Moreover, these goals can refer to transferred, processed, or stored data. Data should be secured if it is transferred over unsecured connections or if it is processed or stored by untrustworthy participants. Compliance requirements might be an additional reason that require the application of data confidentiality.

C. System Integrity and Availability

System Integrity ensures the correct functioning of a system to guarantee that a system acts in an expected and proper way at each point in time. *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. Availability depends on system integrity since availability can not be guaranteed if a system's integrity is compromised. However, the quality of service requirements for availability might require additional technical solutions, e.g. mechanisms for load balancing, to ensure availability.

To ensure the correct functioning of a system, services must be protected from various threats and attacks that can be classified in two categories [13], [14]:

- 1) *malicious content based attacks* – The purpose of this class of attacks (e.g. data with viruses, injection attacks,

recursive/oversized XML documents) is to exploit the service by sending data or messages with malicious content. Countermeasures can be applied by using filters for content inspection.

- 2) *protocol misuse* – Attacks based on the misuse of protocols (e.g. WSDL Scanning, WSDL parameter tampering or error interface probing) intend to gain information or to bypass authorisations. Intrusion detection systems can help to recognize this class of attacks.

The necessity to scan transferred data in a process-aware information system depends on the trustworthiness of the involved participants in a communication process and the importance of involved tasks and services. In general, scanning mechanisms are necessary if borders of security domains are passed, or if the sender of a message is less trustworthy.

III. MODELLING SECURITY IN BUSINESS PROCESSES

The previous Section provided an overview about organisational security concepts that address various security threats in SOA. We outlined that the required level of security in a process-aware information system depends on metrics that determine the value of enterprise assets and assign trust level to each participant. These values determine the risk that is associated with each asset. Appropriate security measures can be applied for each asset to reduce risks. A process-aware information system can be considered as secure, if the asset's risk complies with its business value. To facilitate a specification of security requirements and the determination of risks at the business process layer, we propose a set of security modelling enhancements for BPMN that is described in this section. Figure 1 illustrates a simple example for an order process using BPMN enhanced with security elements.

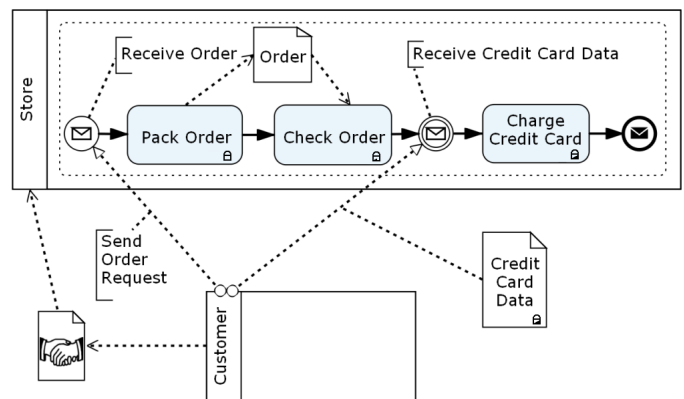








Fig. 1. Security Enhanced BPMN Example

A. Modelling Enhancements for the Evaluation of Assets

Enterprise assets in the scope of process models are represented by performed tasks or data objects that are passed between tasks and participants. To rate these assets we added a property called *AssetRating* (visualised as a small lock, cf. Figure 1) to task and data objects that is assigned a rating as listed in Table I. This classification is based on an generic

TABLE I
OVERALL ASSET VALUE SCALE

	Rating	Description
	Extreme	Endangering human life or threatening enterprise
	Very High	Severe financial or security consequences
	High	Impact on customer services and reputation
	Medium	Affect the enterprises mission
	Low	Minor financial damage and little business impact
	Negligible	No security relevance

approach for asset valuation described by Markus Schumacher et al. [15]. They determine the asset’s overall rating based on three partial values: The *security value* represents the importance the organisation places on guaranteeing the assets value, the *financial value* quantifies the monetary value for the enterprise and the *business value* determines the impact on the business. Each partial value is based on a rating with six predefined categories.

The property *AssetRating* of tasks and data object is visualised as a small lock as shown in Figure 1. In addition the graphical representation also indicates the rating of the asset. In this example the task ‘Pack Order’ has a low rating, while the data object credit card has the highest rating.

B. Modelling Enhancements for the Description of Trust

In the previous Section we identified organisational trust and identity trust as the basic concepts to enable a trustworthy interaction of participants. Organisational trust constitute a prerequisite for interactions by defining a trust relationship between two participants. To express organisational trust in BPMN, we defined an artifact called *Organisational Trust* that can be connected to two or more pools (representing participants) to express this trust relationship, cf. Table III and Figure 1. The organisational trust level quantifying the relationship between two participants is determined by parameters that are added as properties to the artifact. Although various parameters can affect the organisational trust value, we decided to consider three basic parameters for modelling:



The *InitialTrust* parameter represents an initial trust value based on how a trust relationship between organisations has been established. Table II lists four categories that is based on an overview about business security patterns provided by IBM [16]. Two participants can have unlimited trust (*operational* – e.g. they belong to the same organisation), they can trust each other based on contracts (*B2B* – e.g. identity federation) or they can trust each other in a limited way (*B2C* – e.g. another organisation serves as OpenId-Provider).

The *MinAuthenticationTrust* and the *MinRegistrationTrust* value represent the minimum trust that can be put in the process of user registration and authentication. For instance, if an organisation acts as an OpenId identity provider that can be used by users with a valid email-address and that authenticates users based on a user-definable password, then the corresponding trust values will be quite low. We defined an approach to measure authentication trust in [12].

TABLE II
INITIAL TRUST RATING

Type	Trust	Known by
Operational	absolute	same organisation
B2B	high	contract
B2C	low	reputation
WebPresence	none	unknown

TABLE III
SECURITY ENHANCED BPMN ELEMENTS

Name	Organizational Trust	
Base Class	Artifact	
Description	Specifies the trust relationship between two or more participants	
Constrains	The element must be connected to two or more pools	
Properties	<i>Security Rating</i> InitialTrust <i>Integer</i> MinAuthenticationTrust <i>Integer</i> MinRegistrationTrust	
Name	Security Group	
Base Class	Artifact	
Description	Specifies security intentions for a group of task, artifacts or pools	
Constrains	This element groups an arbitrary set of BPMN elements with the same security intentions	
Properties	<i>Security Rating</i> Confidentiality <i>Security Rating</i> DataIntegrity <i>Security Rating</i> SystemIntegrity	

In contrast to organisational trust, identity trust is established dynamically during service access. Hence it does not require a representation in the process model.

C. Modelling Enhancements to express Security Intentions

The security intentions introduced in the previous section, such as confidentiality and integrity, identify countermeasures that should be applied to reduce certain risks. These risks are not solely related to a single entity such as a connection, they usually arise from complex dependencies and interactions. Therefore, our approach is to specify security goals and associated requirements within the broader scope of a group of activities or pools instead of assigning intentions to single elements. We introduced the new artifact *Security Group* to BPMN containing the properties *Confidentiality*, *DataIntegrity* and *SystemIntegrity* representing security intentions, as listed in Table III. The value of these properties represent a security level with the same value scale as listed in Table I. A security intention of a group has to be enforced for contained assets, if the asset’s value exceeds the security level of the intention.

IV. TRANSLATING SECURITY REQUIREMENTS

As discussed in the previous section, our security enhancement for BPMN should enable business process experts and security experts to express security intention in an abstract model. It should facilitate a verification of high level security requirements and trust relationships at the business process layer to detect risks. In addition, this model should enable a generation of security configurations according to the model-driven paradigm. Therefore, we defined an platform-independent security model abstracting from concrete implementation platforms and their provided security features. This

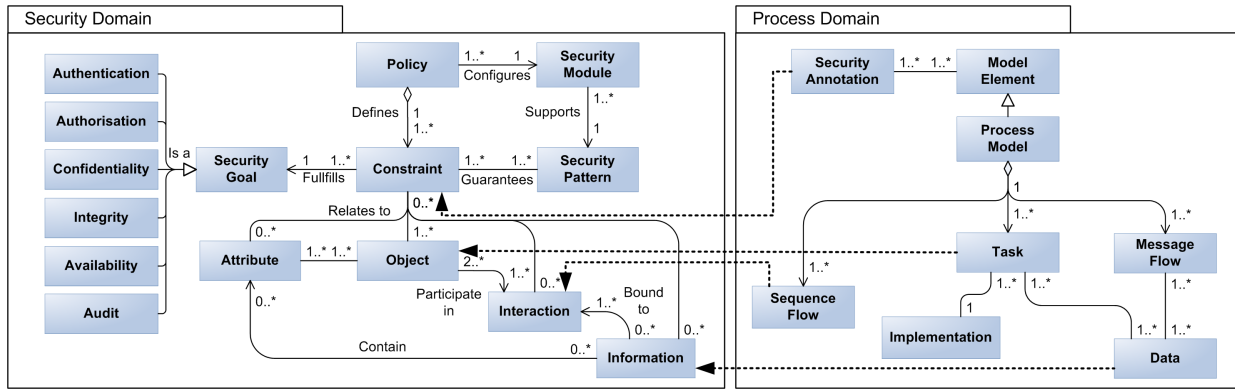


Fig. 2. Security Policy Model

model is used to collect and refine security requirements from the business process model and serves as a common model to generate platform specific security configurations in different languages.

A. Domain-independent Security Model

To express, compare and verify security requirements in a technically and policy language independent way, an abstract security layer has been introduced in previous work [17]. The conception of this layer is close to the OASIS reference model for SOA [18] and enables a straight mapping to the business process layer. The security layer is designed to reveal all security aspects in an SOA landscape and the relationship among affected entities. Therefore, our model describes basic security goals and outlines the relationship to specific security attributes and mechanisms. The relations among security goals and affected entities are described by *Constraints* that are composed in a *Security Policy* as indicated by Figure 2.

These policy constraints always refer to a set of *Objects* (e.g. a service or a participant), *Information* and *Interactions*, which are the basic entities in our security model. We define an object as an entity that is capable of participating in an interaction with other objects. This interaction may involve a set of information that is exchanged. Each object is related to a set of attributes describing its meta information that are derived from the modelling layer.

Furthermore, security constraints are related to a security level that describes the strength of this requirement. This value is assigned during the transformation to the domain-independent security model based on asset evaluation and the context of an security intention at the business layer.

As shown in Figure 2, policies are interpreted and enforced by a *Security Module* that support specific *Security Patterns* to guarantee the defined constraints.

B. Security Pattern

The aforementioned security model represents a set of basic information based on modelled intentions at the process layer. However, the information is not sufficient to generate concrete security configurations since further knowledge is still needed. Expertise knowledge is required to determine an

appropriate strategy to secure a service orchestration, since multiple solutions might exist to satisfy a security goal. For example, confidentiality can be implemented by securing a channel using SSL or by securing parts of transferred messages using WS-Security. The strategy to satisfy a modelled security intention can be determined based on a set of conditions referring to the entities in our security model (e.g. secure channel is applicable when information in transit must be secured and information is not passed through untrustworthy intermediaries).

To describe these strategies and their preconditions in a standardised way, we foster the usage of security patterns. Security patterns have been introduced by Yoder and Barcalow [19] in 1997 and are based on the idea of design patterns as described by Christopher Alexander 'A pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that pattern' [20]. In general, security patterns are defined in an informal way, usually in the natural language, to enable programmer and system designer to adapt the solution described by the pattern to their own specific problem in a particular implementation context.

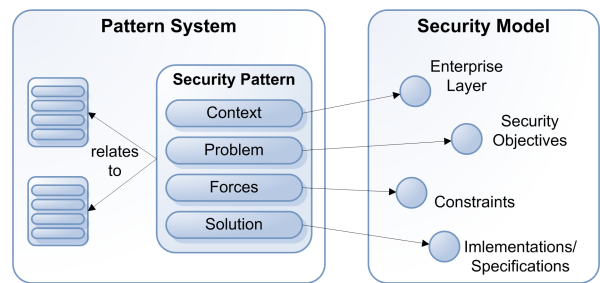


Fig. 3. Security Pattern

However, our intention is to enable an automated selection of appropriate patterns to gather information for the generation of security configurations. Therefore, a formal pattern specification is needed as described by Markus Schumacher [21] to enable a reasoning on a set of security patterns. Figure 3 illustrates the structure of a pattern and its relationship to our

security model. A pattern is composed of

- **problem** – The problems that are addressed in the context of security are threats. As described in Section 2 a group of threats can be related to a security goal. Since this level of abstraction is adequate for our model-driven approach, the problem refers to a security goal.
- **context** – The context describes the enterprise layer or life-cycle phase a pattern is referring to. Buschmann et al. provided a classification of patterns and identified three main categories: *Architectural Pattern*, *Design Pattern* and *Idioms* [22]. Since we are referring to design patterns solely in this paper, this field is less important.
- **forces** – Forces describe the conditions under which a pattern can be applied. These preconditions have to be matched with information provided by our security model to determine appropriate patterns.
- **solution** – A Pattern describes a strategy to solve a problem that is adapted by concrete security mechanisms or security protocols. In our approach, a pattern solution identifies these protocols.

Various pattern might exist that specify different solutions for the same security goal. Moreover, dependencies between patterns might exist. As described by Zimmer [23] there are three basic dependencies that might occur between security patterns: *Usage*, *Refinement* or *Conflict*.

Based on previous work in the field of security patterns [24], we defined a pattern system that describes patterns for each security goal and their relationship. Figure 4 shows an example for the security goals integrity and confidentiality. Two patterns are illustrated: *SecurePipe* to secure data exchanged over an insecure channel, and *MessageConfidentiality* and *MessageIntegrity* to secure data exchanged with messaging. Each pattern refer to particular security goals and identifies appropriate security protocols.

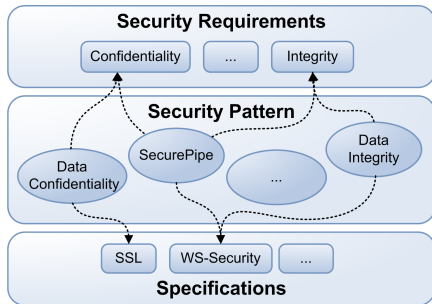


Fig. 4. Confidentiality Pattern

V. MODELL-DRIVEN GENERATION OF SECURITY POLICIES

In the previous sections we presented a model to aggregate security requirements in a domain-independent security model. Security patterns have been introduced as a possibility to choose appropriate security protocols and mechanisms. Therefore, it is necessary to define a mapping for each defined security pattern to a specific security policy. In this section we will describe the generation of security policies based on

the resolved information and present an example to generate policies for a service realized with the Apache Axis 2 Web Service Stack [25].

The Apache Rampart Module [26] is used to apply security to the ingoing and outgoing messages of web service calls. Rampart is configured by the Rampart configuration parameters [26] or the WS-Policy [27] language. The native Rampart configuration uses a flat list of constraints stated in XML with implementation specific information, while WS-Policy provides a grammar to express and group policy assertions describing a broad range of requirements on a more abstract level. Since WS-Policy is not specific to a problem domain, security assertions are defined in the WS-SecurityPolicy specification providing an implementation independent approach to express constraints for WS-Security, WS-Trust and WS-Secure Conversation [28].

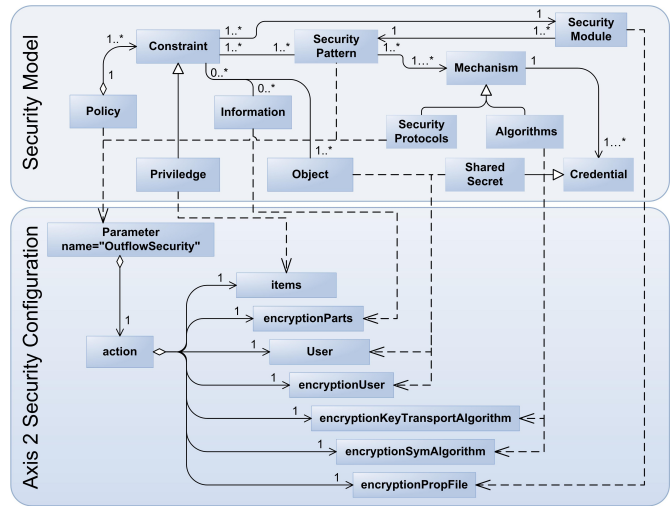


Fig. 5. Mapping to the Apache Rampart Security Configuration

In the scope of the native Rampart configuration, a policy is represented by the element *parameter* with the attribute *name='OutflowSecurity' | 'InflowSecurity'*. The attribute controls whether the security settings are applied to ingoing or outgoing messages. The policy element *parameter* contains the element *action* that encapsulates several elements to configure Rampart. Figure 5 shows the mapping to the Rampart configuration for outgoing messages regarding the security goal confidentiality. Since Rampart operates on the exchanged Web Service messages, this transformation as well as the generated policy are associated with the security pattern *message confidentiality*.

Confidentiality is enforced by Rampart containing an *item* element with the *encrypt* value. In the example (cf. Figure 1) we defined the inflow security configuration for the service implementation referenced by the task *Charge Credit Card* that is receiving customer data. Parts of the resulting XML configuration file is shown in Listing 1. In this simple example always the whole message body is encrypted. Therefore the value of *items* is *'Encrypt'*. *User* and *encryptionUser* refer to the involved service endpoints - these elements reference

the name of the involved objects to resolve the associated keys for encryption. Implementation specific details stating how to resolve the keys are specified in a Java property file that is referenced by *encryptionPropFile*. Furthermore, which information has to be encrypted is defined by *encryptionParts*. Finally, *encryptionSymAlgorithm* and *encryptionKeyTransportAlgorithm* specify the employed algorithms.

```

1 <!-- Body Encryption for calling service DebitOrder-->
2 <parameter name="InflowSecurity">
3   <action>
4     <items>Encrypt</items>
5     <user>Customer</user>
6     <encryptionUser>OnlineStore.DebitOrder</
7       encryptionUser>
8     <encryptionPropFile>OnlineStore.DebitOrder.
9       properties</encryptionPropFile>
10    <encryptionParts>{{}}Body</encryptionParts>
11    <encryptionSymAlgorithm>
12      http://www.w3.org/2001/04/xmlenc#tripleDES-cbc
13    </encryptionSymAlgorithm>
14    <encryptionKeyTransportAlgorithm>
15      http://www.w3.org/2001/04/xmlenc#rsa-1_5
16    </encryptionKeyTransportAlgorithm>
17  </action>
18 </parameter>

```

Listing 1. Rampart Inflow Encryption

VI. RELATED WORK

The domain of model-driven security in the context of business processes is an emerging research area. The need to describe an application scenario related security policies on an abstract level is discussed in [29]. Recent work done by Nagaratnam *et al.* [30] discusses an approach to express security requirements in the context of business processes and how to monitor and manage them on the different enterprise architecture levels. This intention does not provide a detailed analysis of security goals, their conceptual models, and their relationship to the business process related entities.

This issue has been addressed by Rodriguez *et al.* [31], [32] 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. Although they support several security intentions, they do not provide a comprehensive security model based on the evaluation of assets considering authentication and trust. A model-driven scenario based on their annotations is considered as future work.

Our security model could be complemented by modelling concepts for compliance rules for business processes [33] 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 [34]. SecureFlow implements a Workflow Authorisation Model to define and enforce authorisations at runtime for users, roles, and workflow tasks. In [35], Crook *et al.* proposed a framework to model and verify access control policies based on the derivation of roles from their organisational context. In contrast to our general security modelling concept these approaches focus on authorisation, without considering the relation to other security requirements.

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 [36] is a model-driven security approach for process-oriented systems focusing on access control. Similar to SecureUML, Jürjens presented the UMLSec extension for UML [37] 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.

VII. CONCLUSION

Process modelling notations provide a suitable abstract perspective to specific security goals on a more accessible level, as we have shown in [2]. In particular, our previous work has been focused on the description and transformation of authorisation requirements [17]. To enable an overall evaluation of security requirements based on the value of enterprise assets and the trustworthiness of participants, we provided a compilation of organisational security aspects in this paper and outlined their relationships. Based on this overview, we introduced an enhancement for business process modelling to express trust, confidentiality and integrity requirements on an abstract level.

We presented a model-driven approach addressing the difficulty to generate security configurations for a process-aware information system. 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*, and *Interactions*. The strength of our model lies in its general description of security goals, and the abstraction from technical details. Security intentions and related requirements defined at the process layer can be mapped to this model. To resolve concrete security protocols and mechanisms, a security pattern system has been described that resolves appropriate security protocols with regard to specific preconditions.

The gathered information can be mapped to an arbitrary application or technical specification. As an example we introduced a mapping to the configuration of the Axis2 Rampart module. As a result, these security configurations would be

consistent with the affected business processes and result in a decreased error-proneness.

A. Future Work

We stated that our proposed security extension for BPMN is a promising approach to describe, verify and translate security requirements at an abstract level. While we described a model-driven approach to translate security intentions to concrete security configuration, we just outlined the possibility to perform the verification. In the next step, we will address the process of verification in detail. In addition, we will illustrate the benefits of our model-driven approach and the applicability of the verification with a detailed case study.

Moreover, 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 [33] and risk assessment concepts [38], thus enabling the specification, generation, and enforcement of compliant and secure business processes in a service-oriented environment.

Another important aspect is the consumption and provision of services and service compositions across trust domains [39]. To enable a model-driven approach regarding cross-organisational service compositions, it must be considered that federation partners state their own security requirements that must be considered as well as compatibility issues. In addition, it is important to reveal dependencies and contradictions between requirements from different service providers that would prevent a secure or compatible service provisioning. This information can also be used to provide feedback at the modelling layer.

REFERENCES

- [1] M. Weske, *Business Process Management*. Springer, 2007.
- [2] C. Wolter, M. Menzel, A. Schaad, P. Miseldine, and C. Meinel, "Model-driven business process security requirement specification," *Journal of Systems Architecture Special Issue on Secure Web Services*, 2008.
- [3] C. Wolter, M. Menzel, and C. Meinel, "Modelling security goals in business processes," in *Proc. GI Modellierung 2008*, no. ISBN 978-3-88579-221-5. GI LNI, Berlin, Germany, 1008.
- [4] C. Wolter and A. Schaad, "Modeling of task-based authorization constraints in bpmn," in *BPM*, 2007, pp. 64–79.
- [5] A. Rodríguez, E. Fernández-Medina, and M. Piattini, "A bpmn extension for the modeling of security requirements in business processes," *IEICE Transactions*, vol. 90-D, no. 4, pp. 745–752, 2007.
- [6] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*. Prentice Hall Professional Technical Reference, 2002.
- [7] D. H. McKnight and N. L. Chervany, "The meanings of trust," *Technical Report, University of Minnesota*, 1996.
- [8] R. S. Sandhu and E. J. Coyne, "Role-based access control models," *IEEE Computer*, vol. 29, pp. 38–47, 1996.
- [9] R. K. Thomas and R. S. Sandhu, "Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management," in *DBSec*, 1997, pp. 166–181.
- [10] H. bo Shen and F. Hong, "An attribute-based access control model for web services," in *pdcat*. IEEE Computer Society, 2006, pp. 74–79.
- [11] D. Pugh, "Role activation conflict: A study of industrial inspection," *American Sociological Review*, vol. 31:835-42, 1966.
- [12] I. Thomas, M. Menzel, and C. Meinel, "Using quantified trust level to describe authentication requirements in federated identity management," in *Proc SWS*, 2008.
- [13] P. Lindstrom, "Attacking and defending web services, a spire research report," 2004. [Online]. Available: http://forumsystems.com/papers/Attacking_and_Defending_WS.pdf
- [14] "Eege project. grid and web service security vulnerabilities and threads analysis and model," 2005. [Online]. Available: <https://edms.cern.ch/documents/632020/>
- [15] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns - Integrating Security and System Engineering*. John Wiley & Sons, Ltd, 2006.
- [16] IBM, "Introduction to business security pattern," 2004.
- [17] C. Wolter, M. Menzel, and C. Meinel, "Modelling security goals in business processes," in *Proc. GI Modellierung 2008*, no. ISBN 978-3-88579-221-5. GI LNI, Berlin, Germany, 1008.
- [18] M. MacKenzie, K. Laskey, F. McCabe, P. Brown, and R. Metz, "Reference model for service oriented architecture 1.0," OASIS Committee Specification, February 2006.
- [19] J. Yoder and J. Barcalow, "Architectural patterns for enabling application security," in *PLoP*, 1997.
- [20] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobsen, I. Fiksdahl-King, and S. Angel, *A Pattern Language: Towns - Buildings - Construction*. Oxford University Press, 1977.
- [21] M. Schumacher, *Security Engineering with Patterns - Origins, Theoretical Model, and New Applications*. Springer, Berlin, 2003, no. ISBN 3-540-40731-6.
- [22] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Pattern*. John Wiley & Sons, Ltd, 1996.
- [23] W. Zimmer, "Relationships between design patterns," *Pattern languages of program design*, pp. 345–364, 1995.
- [24] N. Yoshioka, H. Washizaki, and K. Maruyama, "A survey on security patterns," *Progress in Informatics*, vol. 5, pp. 35–47, 2008.
- [25] S. Perera, C. Herath, J. Ekanayake, E. Chinthaka, A. Ranabahu, D. Jayasinghe, S. Weerawarana, and G. Daniels, "Axis2, middleware for next generation web services," in *ICWS*. IEEE Computer Society, 2006.
- [26] S. A. et. al., "Apache rampart : Ws-security module for axis2," 2008. [Online]. Available: http://ws.apache.org/axis2/modules/rampart/1_0/security-module.html
- [27] S. Bajaj, D. Box, and et. al., "Web services policy 1.2 - framework (ws-policy)," Public Draft Specification, April 2005. [Online]. Available: <http://www.w3.org/Submission/WS-Policy/>
- [28] J. Rosenberg and D. Remy, *Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. Pearson Higher Education, 2004.
- [29] M. Tatsubori, T. Imamura, and Y. Nakamura, "Best-practice patterns and tool support for configuring secure web services messaging," in *ICWS*, 2004, pp. 244–251.
- [30] 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.
- [31] A. Rodríguez, E. Fernández-Medina, and M. Piattini, "Towards a uml 2.0 extension for the modeling of security requirements in business processes," in *TrustBus*, 2006, pp. 51–61.
- [32] A. Rodriguez, E. Fernandez-Medina, and M. Piattini, "Towards cim to pim transformation: From secure business processes defined in bpmn to use-cases," in *BPM*, 2007, pp. 408–415.
- [33] S. W. Sadiq, G. Governatori, and K. Namiri, "Modeling control objectives for business process compliance," in *BPM*, 2007, pp. 149–164.
- [34] W. kuang Huang and V. Atluri, "Secureflow: A secure web-enabled workflow management system," in *ACM Workshop on Role-Based Access Control*, 1999, pp. 83–94.
- [35] R. Crook, D. C. Ince, and B. Nuseibeh, "Modelling access policies using roles in requirements engineering," *Information & Software Technology*, vol. 45, no. 14, pp. 979–991, 2003.
- [36] D. Basin, J. Doser, and T. Lodderstedt, "Model Driven Security for Process-Oriented Systems," in *SACMAT '03: Proceedings of the 8th ACM symposium on Access control models and technologies*, 2003.
- [37] J. Juerjens, "UMLsec: Extending UML for Secure Systems Development," in *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*, 2002, pp. 412–425.
- [38] J. H. Lambert, R. K. Jennings, and N. N. Joshi, "Integration of risk identification with business process models," *Syst. Eng.*, vol. 9, no. 3, pp. 187–198, 2006.
- [39] M. Menzel, C. Wolter, and C. Meinel, "Towards the aggregation of security requirements in cross-organisational service compositions," in *Proc. 11th BIS*, no. ISBN: 978-3-540-79396-3. Springer LNCS, Innsbruck, Austria, May 2008.