SECTET – An Extensible Framework for the Realization of Secure Inter-Organizational Workflows

Michael Hafner[1], Ruth Breu[1], Berthold Agreiter[1], Andrea Nowak[2]

[1] Universität Innsbruck, Institut für Informatik, Innsbruck, 6020, Österreich

[2] Austrian Research Center Seibersdorf Research GmbH, Seibersdorf, 2444, Österreich

Abstract. SECTET is an extensible framework for the model-driven realization of security-critical, inter-organizational workflows. The framework is based on a methodology that focuses on the correct implementation of security-requirements and consists of a suite of tools that facilitates the cost-efficient realization and management of decentralized, security-critical workflows. After giving a description of the framework, we show how it can be adapted to incorporate advanced security patterns like the Qualified Signature, which implements a legal requirement specific to e-government. It extends the concept of digital signature by requiring that the signatory be a natural person.

1 Introduction

SECTET is a framework for the high-level development and management of security-critical, inter-organizational workflows based on Web services. The framework supports business partners during the design, the realization and the management of a common *Global Workflow* - a decentralized collaboration between actors across domain boundaries [1]. Realizing the paradigm of Model Driven Security [2], the framework targets the correct technical implementation of business-level security-patterns which are integrated at the abstract level into the specification of the Global Workflow.

A Global Workflow specifies the message flow between partners in a distributed environment with no central control, by means of UML 2.0 diagrams. The models of this virtual process are translated into executable configuration files for Workflow Management Systems (WFMS) and security components of target architectures located at every partner's node based on XML and Web services standards and technologies [3]. A target architecture encapsulates a set of core Web services, which may access back-end services, orchestrates them through a WFMS, and guards them by imposing specified security policies to inbound and outbound service calls.

In this paper we present the core components of our framework and illustrate the main concepts of our methodology for the systematic design and realization of security-critical inter-organizational workflows with a portion of a workflow-scenario drawn from e-government. We additionally show how the framework can be adapted to incorporate advanced security patterns like the *Qualified Signature,* which extends the concept of digital signature by requiring a natural person to sign [4].

Section 2 sketches the technical background of our framework and refers to related

Hafner M., Breu R., Agreiter B. and Nowak A. (2006).
SECTET – An Extensible Framework for the Realization of Secure Inter-Organizational Workflows.
In *Proceedings of the 4th International Workshop on Security in Information Systems*, pages 47-57

work. Section 3 introduces a motivating scenario and shows how to realize a secure inter-organizational workflow by applying the concepts of our framework to a specific case. Section 4 shows how the framework can be extended according to specific business requirements. For a comprehensive introduction to the SECTET framework or a detailed presentation of single aspects, please refer to a series of accompanying papers (e.g., [5], [6] and [7]).

2 Background

Workflow Standards. The Business Process Execution Language for Web Services (WS-BPEL [8]) is an XML-based language to compose workflows on top of atomic Web services. It provides mechanisms to define executable business processes and, with limitations, abstract business protocols. Collaboration protocols like BPSS [9] and WS-CDL [10] provide the means to formally specify collaborations in distributed environments by offering a global view on collaborating services. BPML [11] and ebXML [10] are alternative languages to specify executable processes. BPML is quite similar to WS-BPEL as it supports Web services standards, but it is considered as semantically weaker. ebXML comprises a powerful set of standards for the specification of B2B protocols but it is not compatible to the Web services concept. Since we strongly focus on Web services technologies, which is the most widespread technology with strong vendor support, we consider WS-BPEL as the appropriate top-layer standard to model local workflow processes in our context.

Web Services Security. Currently a comprehensive set of Web services security standards is emerging. OASIS has proposed a security extension built on top of the SOAP Protocol [12]. The extension uses the XML encryption and signature mechanism to add security features to SOAP messages ([13], [14]). This way, security mechanisms can be integrated into the header and the body of a SOAP message, and be sent via any transport channel without compromising security. Beside transport level security extensions, a variety of standards provides means to manage and exchange security policies. XACML [15] is a standard to define access control for resources in a system. Sun has proposed a specific profile for XACML – called Web Services Policy Language - to define the reconcilement of access rights between partners. SAML [16] is a standard for the exchange of security tokens. WS-Policy [17] allows for the definition of protocol level security requirements.

Related Work. Many approaches deal with secure document exchange and workflow management in centrally organized environments, e.g., the Author-X system [18], Akenti [19] and the EU-financed project TrustCom [20]. A big community is working on issues related to inter-organizational workflow management systems [1], [21], [22], and [23]. We do not aim to contribute a novel approach to this field. Instead, we rely on UML models for modeling workflow and security into workflow management systems based on Web services technology. Security extensions for workflow management systems are treated in [24], [25] and [26] although at a quite technical level. A model driven approach that is close to the idea of our framework is [2]. It introduces the concept of Model Driven Security for a software development process that allows for the integration of security requirements through system models and supports the generation of security infrastructures. But

this approach focuses exclusively on business logic, whereas we concentrate on inter-organizational workflow management.

3 The SECTET - Framework

In this section we introduce the conceptual foundation (3.1), we then present the components of the SECTET Framework (3.2) and finally show how the framework is applied to a real-life case study, by modeling a Global Workflow that complies to three basic security requirements of the concept *Secure Document Flow* (3.3).

3.1 Conceptual Framework

3.1.1 Global and Local Workflows

We define a *Global Workflow* as a network of partners cooperating in a controlled way by calling services and exchanging documents. In order to guarantee loose coupling and design autonomy at the local level without compromising interoperability, we make the assumption that there is no central control of the inter-organizational workflow. This means that there is no central *Workflow Management System* (WfMS) or document repository.

The Global Workflow can be thought of as virtual process that emerges through peer-to-peer interaction of executable *Local Workflows*, which traditionally are located in different domains. Through their collaboration they exactly realize the behavior as specified in the Global Workflow. This kind of decentralized application is especially suited to scenarios where central management is not desirable, may it be for social, political or competitive reasons (e.g., public procurement, e-government).

In contrast to the Global Workflow, a Local Workflow is executed on a WfMS. This kind of process accesses back-end functionality by calling local services and orchestrates these services according to some workflow logic. In our approach we focus on the Global Workflow and assume that partners already have implemented the application functionality they agreed to contribute to the Global Workflow.

3.1.2 Model Views

Our approach is based on two orthogonal views: the *Interface View* and the *Workflow View*. The latter is further divided into the *Global Workflow Model* (GWfM) specifying the message exchange between cooperating partners, and the *Local Workflow Model* (LWfM) that describes the application and the workflow logic which is local to each partner. The Interface View describes the interface of every partner independently of the components' usage scenario. The application of orthogonal perspectives allows us to combine the components that provide the services, into various global Workflows, each one realizing a particular usage scenario.

3.1.3 Interface View

The Interface View represents the contractual agreement between the parties to provide a set of services. It specifies the minimum set of technical and domain level constraints and thereby links the GWfM to the LWfM. It describes the interface of

every partner's services independently of their usage scenario and consists of four sub-models:

The *Document Model* is a UML class diagram that describes the data type view of the partner. We talk of documents because we do not interpret this class diagram in the usual object oriented setting but in the context of XML schema. The *Interface Model* contains a set of abstract (UML-) operations representing services the component offers to its clients. The types of the parameters are either basic types or classes in the Document Model. Additionally, pre- and post-conditions (in OCL style) may specify the behaviour of the abstract services. The *Role Model* describes the roles having access to the services and finally the *Access Model* describes the conditions under which a certain role has the permission to call a service. The permissions are written in SECTET-PL [27] in a predicative style over the structures of the Document Model. We provide an in-depth view on model dependencies in [5].

In most cases, when parties agree to realize a Global Workflow, the Interface View or some of its sub-models already exist as parties may already make some of there services accessible to the outside world. Very often the Document Model, which corresponds to the information model and the Interface Model, consisting of method signatures already exist.

3.1.4 Security in Global Workflows

The security requirements are modeled at the design level and integrated as security patterns into the models of the Workflow and the Interface View. They are then translated into executable security components or configuration artifacts for target architectures. Our framework currently supports the following security patterns:

A. Secure Document Flow (Module SECTINO). This pattern allows the specification of a secure document exchange satisfying *"End-to-End Security"*, which means that the requirements are satisfied even in case of being routed via intermediaries. Documents or parts can be qualified with the requirement of confidentiality, which is implemented with the help of public key encryption, Integrity, which means a kind of system signature or non-repudiation of sender or receiver, which triggers signaling at the protocol layer through the exchange of signed timestamps.

B. Context Dependent Access Constraints (Module SECTET-PL). We specify conditions under which a specific role has the right to access services in the Access Model with the help of an extended OCL-style predicate logic [27]. The right to call an operation of a specific Web service may depend either on the caller's role or on parameters that may depend on the system's environment (e.g., time, IP-Address etc.) or that are sent together with the service call.

C. Application Domain Specific Security (Module SECTET-Extensions). Many scenarios have to integrate complex security patterns that satisfy complex legal or business-driven requirements. In most cases they are based on the basic requirements of confidentiality, integrity or non-repudiation. The *Qualified Signature* is an e-government specific requirement that extends the concept of the system signature, which is used to guarantee integrity to a legal entity (e.g., a citizen). In public procurement anonymity of bidders is guaranteed by a specific security protocol and in most cases requires a trusted third party.

3.2 SECTET-Framework Components

3.2.1 Modelling Component

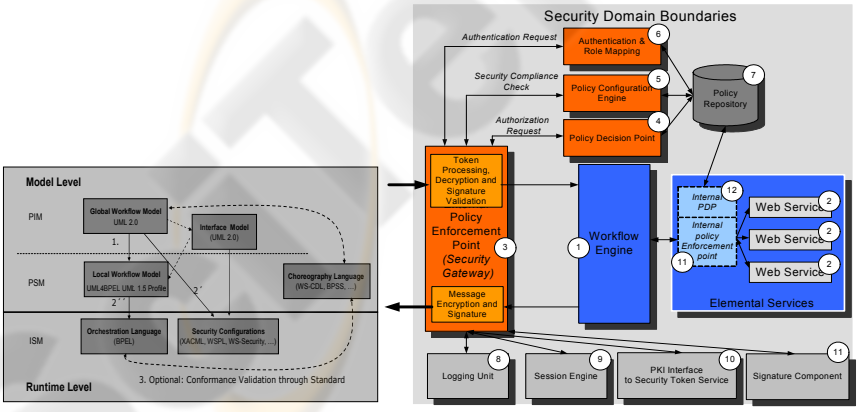
The modeling component comprises a set of tools that supports the collaborative modeling of the Global Workflow and its security-requirements at the application domain level in a platform independent context. According to OMG's paradigm of Model Driven Architectures (MDA) we specify three levels of abstraction (Fig. 1a):

The **Platform Independent Model** (PIM) captures the domain level knowledge and abstracts from implementation details of the target architecture in two respects. The global process is defined independently of platform technology at partner nodes (J2EE, DotNet, Corba etc.) and independently of workflow standards like BPEL or BPSS that may be used to implement local workflow logic. We model the Global Workflow using UML 2.0 activity diagrams. Security requirements are integrated as constraints associated to object nodes, which act as a logical container for documents flowing from partner to partner. Documents and interface signatures are modeled as class diagrams in the Document and the Interface Model respectively.

The **Platform Specific Model** (PSM) describes the system on its intended platform (e.g. BPEL4WS) by integrating platform specific syntax and semantics. Parts of the global workflow logic are translated into executable stubs in a specific workflow standard for WfMS at partner nodes. The translation component (Sect. 4.2) generates the interfaces of the Local Workflows that are accessed by the Global Workflow.

The **Implementation Specific Model** (ISM) represents the reference architecture that acts as the runtime environment at local partner nodes. Currently our framework targets a Web services based reference architecture.

Applying the MDA approach means the transformation of a PIM into a PSM and / or an ISM. We extend the MDA approach towards Model Driven Security in the sense that we integrate security requirements at the abstract level into the PIM. The PIM and the PSM are mapped onto each other and finally translated into configuration artefacts for the runtime environment.



(a) (b)

Fig. 1. Model Dependencies (a) and Reference Architecture (b).

3.2.2 Reference Architecture

The target architecture as depicted in Figure 1 represents the runtime environment for the Local Workflow and its back-end services at the partner node. The architecture is based on the data-flow model of XACML [15]. The components implement a set of XML- and Web services Technologies and standards.

Workflow Component. A workflow engine (1), based on an XML-based Workflow Language like BPEL orchestrates the sequence of local Web services (2) calls as specified in the LWfM. The engine bundles the services to a composition that may be offered to a service of its own. As described above, Global and Local Workflows are modeled in the PIM and PSM as UML 2.0 diagrams and translated into runtime artifacts (BPEL files for the Workflow Engine, WSDL-Files for Web services description) for the target architectures.

Workflow Security. The workflow engine and the Web services in the back-end are wrapped by security components. So-called Policy Enforcement Points (PEP) act as security gateways. We differentiate between the external (3) and the internal PEP (11). The external PEP is the single point of entry into the domain. He is in charge of implementing requirements related to message integrity, confidentiality and non-repudiation for all external communication. To this end, it checks the correct signatures and decrypts incoming requests or response. Correspondingly, the gateway signs outgoing requests or responses and encrypts them as specified in the global workflow model. Both, the security gateway and the workflow engine implement the requirements by configuration. The configuration data is generated from the respective models views. After receiving a service request the PEP authenticates the caller with support of the Authentication and Role Mapping Unit (6), checks the compliance of the incoming message according to signed and encrypted elements (5) and finally queries the Policy Decision Point (PDP) (4) for access rights. After successful completion of these three steps, the PEP forwards the request to the workflow engine (1) which then performs the service orchestration. Optionally, an internal PEP, that merely acts as a role mapping unit may map the caller's global role to some internal role representation required by the back-end applications.

3.2.3 Model Transformation and Code Generation Component

This component has three tasks:

A. Mapping Global to Local Workflows. Those parts of the GWfM that correspond to interfaces that local process nodes should implement are translated into stubs of executable process code. In our case study the partners imported the stubs into Oracle's Process Manager [28] and added service calls to back end functionality.

B. Generation of Security Artefacts. The security requirements in the PIM are translated into configuration artefacts for the target architectures. We made the assumption that every partner wrapped his local node with the security components specified in the reference architecture.

C. Import and Export of Global Workflows. A choreography which was specified in a wide-spread standard like WS-CDL or BPSS can be imported into the framework [29]. The intention is the representation of a global workflow with a standard that is completely independent of the technical platform. WS-CDL is only suited to representation of choreographies, that were designed to run on a Web services based platform. The same can be said about BPSS and ebXML.

3.3 Scenario

The example captures an inter-organizational process in e-government. It is drawn from a case that was elaborated within the project SECTINO [5]. The project's vision was defined as the development of a framework supporting the systematic realization of e-government related workflows.

The workflow “Municipal Tax Collection" describes a Web services based interaction between three participants: a tax-payer (the Client), a business agent (the Tax Advisor) and a public service provider (the Municipality). In Austria, wages paid to employees of an enterprise are subject to the municipal tax. According to the traditional process, corporations have to send an annual statement via their tax advisor to the municipality. The latter is responsible for collecting the tax. It checks the declaration of the annual statement calculates the tax duties and returns a tax assessment notice to the tax advisor. In our case the stakeholders in this public administration process agreed to implement a new online service, which enables citizens and companies to submit their annual tax statements via internet. Due to legal considerations, the process had to be realized in a peer-to-peer fashion and should integrate security requirements like integrity, confidentiality and non-repudiation and ultimately advanced security patterns like the Qualified Signature.

3.3.1 Modelling the Global Process

Table 1. Informal Description of Global Workflow and Security Requirements.

1. Client sends annual statement to Tax Advisor *2. Tax Advisor does internal processing on the document* **3. The Tax Advisor forwards the processed annual statement** *4. Municipality calculates tax duties and* **5. Municipality returns a notification to Tax Advisor** *6. Tax Advisor processes notification* **7. Tax Advisor sends tax information to Client** *Internal Processing Steps* Document flow Interaction Activity	**A. Integrity**: All exchanged documents have to be signed by the sending party with a “System Signature” when leaving the domain boundaries in order to guarantee message integrity. **B. Confidentiality**: The `annual income` and the `clientId` of the `annual statement` and the tax `notification` are confidential, and should only be readable to the Municipality **C. Non-repudiation**: Receipt of the `annual statement` and the `notification` must not be deniable.

Modeling the global process requires two steps. In a first step the partners had to agree on a global process scenario they wanted to implement. Table 1 informally summarizes the Global Workflow. The partners identified three roles (Client, Tax Advisor and Municipality), four interactions (steps 1, 3, 5 and 7) and four documents flowing between the partners (annual statement, processed annual statement, notification and information).

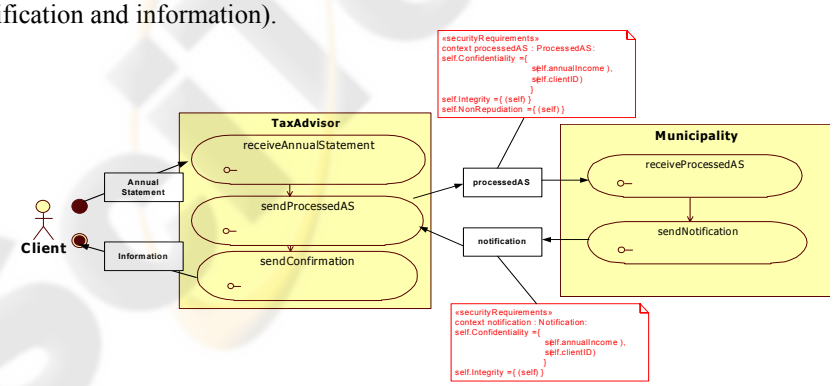


Fig. 2. Global Workflow Model as UML 2.0 Activity Diagram.

Figure 2 shows the result of step 2: the Global Workflow is specified as a UML 2.0 activity diagram. It describes the collaboration of the three roles in terms of the interactions in which the participating parties engage. Model information is confined to "observable behaviour", corresponding to the message flow between the participants, the interaction logic and the control flow between the elementary actions. The graphical representation of the Global Workflow as a UML 2.0 activity diagram allows formalizing the process description in a very intuitive way.

3.3.2 The Transformation Process

The first part of the translation process generates those parts of the Local Workflow Model which represent the interfaces to the GWfM (e.g., the Interaction-Activities `receiveAnnualStatement` and `SendProcessedAS`) for every partner-role (Fig. 3a). Those partners who are assigned a specific role in the GWfM take the BPEL code stubs, import them in the BPEL design tool of their choice (e.g., [28]) and complement service calls to back-end application. The models can then be translated into BPEL and WSDL files for the workflow engine with existing tools (e.g., [30]).

```
<process name="Sectino_TaxAdvisor_LWfM">
  <partnerLinks>
    <partnerLink name="MS_Provider" partnerLinkType="Municipality_LWfM"
     partnerRole="Municipality_LWfM"/>
    <partnerLink name="TS_Provider" partnerLinkType="TaxAdvisor_LWfM"
    partnerRole="TaxAdvisor_LWfMRequester"
    myRole="TaxAdvisor_LWfMProvider"/>
  </partnerLinks>
  <variables>
    <variable name="input" messageType="AnnualStatement"/>
    <variable name="output" messageType="Confirmation"/>
    <variable name="input_MU" messageType="ProcessedAS"/>
    <variable name="output_MU" messageType="Notification"/>
  </variables>
  <sequence name="main">
   <receive name="receiveInput" partnerLink="TS_Provider"
    portType="TaxAdvisor_LWfM"
    operation="sendAnnualStatement" variable="input" createInstance="yes"/>
   <invoke // !!! INSERT CALLS TO LOCAL SERVICES !!! //>
   <invoke partnerLink="MS_Provider" portType="Municipality_LWfM"
    operation="sendProcessedAS" inputVariable="input_MU"
    outputVariable="output_MU" name="sendProcessedAS"/>
   <invoke // !!! INSERT CALLS TO LOCAL SERVICES !!! //>
   <invoke name="callbackClient" partnerLink="TS_Provider"
    portType="TaxAdvisor_LWfMCallback"
    operation="onResult" inputVariable="output"/>
  </sequence>
</process>
```

(a)

```
PolicySet {(target=<AnnualStatement>)
 PolicySet { target=<outbound>
  PolicySet {(target=<processedAS>)

   Policy (Aspect = "Confidentiality") {
     Rule {
       Signature-Algorithm = "RSA-SHA1",
       Node1 = "/self/annualIncome",
       Node2 = "/self/clientID",
       Recipient = "Municipality"} }
   Policy (Aspect = "Integrity") {
     Rule {
       Signature-Algorithm = "RSA-SHA1",
       Node1 = "/self/",
       Recipient = "Municipality"}
   }
 PolicySet { target=<inbound>
  PolicySet {(target=<processedAS>)

    Policy (Aspect = "Qualified Sign") {
     Rule {
       Signature-Algorithm = "RSA-SHA1",
       Node1 = "/self/",
       Source = "Municipality"
       Signatories = 2}
}}}}
```

(b)

Fig. 3. BPEL File (a) and Security File (b) for Role TaxAdvisor.

Security Components. The second part generates the configuration files for the security components in the architecture (Fig. 3b). The security requirements in the constraint box in GWfM (Fig. 2) are translated into an XACML file stored in the Policy-Repository (7) and loaded into the PCE (5) at runtime in order to determine the compliance of inbound messages and to identify security mechanisms to be applied to outbound messages. Figure 4b) shows how the security requirements are translated into a configuration file for the role `TaxAdvisor`. Figure 4 shows the dependency between the security requirements in the GWfM and the Document-Model of the Interface View.

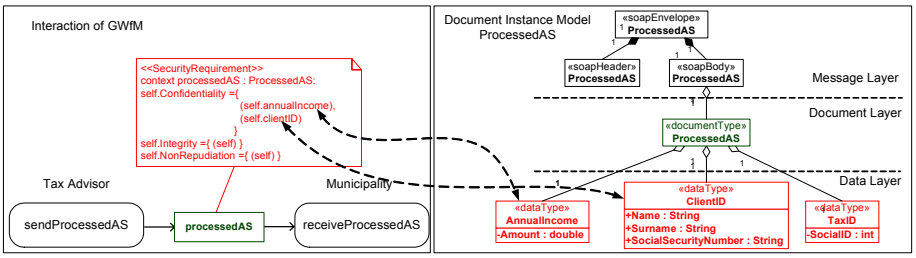


Fig. 4. Dependency of Security Requirements to Elements of the Document Model.

4 Framework Extensions

In many e-government applications a technical signature is not sufficient. In our case, the partners additionally specified that the `notification` sent by the Municipality have to be signed personally by at least two clerks (which corresponds to a "Qualified Signature" according to the Austrian E-Government Law [4]). In this section we show how the framework is extended to incorporate this domain specific security requirement. Specifically we show how the new requirement is integrated into the meta-model of the PIM (4.1), how it is visualized in the PIM and how it is realized in the Target Architecture (4.2). We close the section with a brief conclusion (4.3).

4.1 Platform Independent Model – The Meta-model Integration

The requirement is expressed as an OCL expression associated to the Document Model of the Interface View. The meta-model of SECTINO-UML has been extended to fit this new requirement. `DocumentSecurityRequirements` (Fig. 5) was added as a new class of security requirements. It links a `DocumentType` to a `LocalRole`. A `LocalRole` inherits from a `DomainRole` which represents an actor in the GWfM. The hierarchy of local roles is opaque to the Global Workflow.

We decided to make the security requirement a category of its own instead of associating it to, because the category `WorkflowSecurityRequirements` refers to security related to the Global Workflow, whereas the Qualified Signature by its very nature is a requirement that must be associated of the Local Workflow as – for example - it requires `LocalRoles` to sign a document.

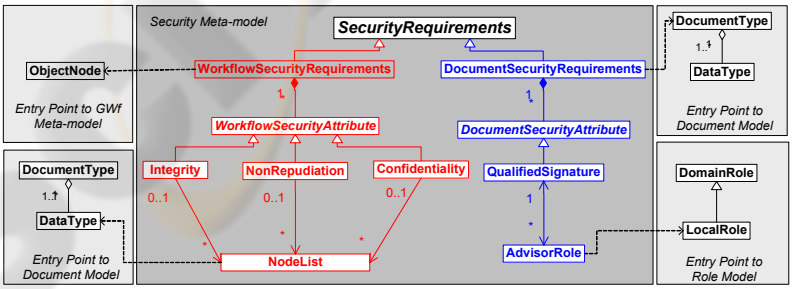


Fig. 5. Security Meta-model and its Links to the GWfM and Sub-models of the Interface View.

4.2 Platform Independent Model and Reference Architecture

Figure 6 shows how the security requirement – namely that the `Notification` has to be signed twice - is represented at PIM-level (once by the `LocalRole SeniorClerk` and once by a `ClericalAssistant`). The requirement Qualified Signature in the Document Model is translated into a corresponding entry into the XACML policy file (Figure 3b) and is implemented through an additional component in the Reference Architecture. In case of an appropriate requirement for an outbound message, the PEP buffers the message, identifies, notifies authenticates the signatory and lets him sign the message via an applet popping up in his work-place.

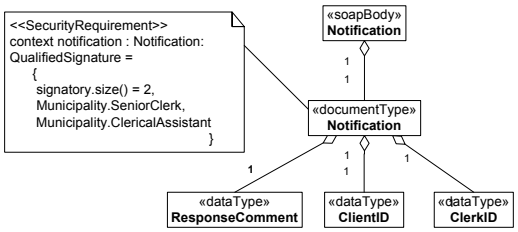


Fig. 6. Document Model "Notification" with Security Requirement.

4.3 Conclusion

SECTET implements a comprehensive approach for the model-driven realization and management of inter-organizational workflows. Future work has to be done in several directions. The set of supported security requirements has to be extended (e.g., rights delegation). Additionally, we are currently implementing a testing environment and working out requirements for efficient change management of inter-organizational workflows. Positive results in pilot applications with industrial partners encourage us to further steps.

References

1. W.M.P. van der Aalst.: Loosely Coupled Inter-organizational Workflows. In: Information and Management 37 (2000) 2, pp. 67-75.
2. D. Basin, J. Doser, T. Lodderstedt: Model Driven Security for Process-Oriented Systems. In 8th ACM Symposium on Access Control Models and Technologies. ACM Press, 2003.
3. IBM and Microsoft. 2002: Security in a Web Services World. A joint security whitepaper from IBM Corporation and Microsoft Corporation. April 2002, Version 1.0.
4. Austrian Signature Act (Signaturgesetz - SigG), Art. 1 of the Act published in the Austrian Federal Law Gazette, part I, Nr. 190/1999.
5. M. Hafner et al.: Model Driven Security for Inter-Org. Workflows in E-Gov. To Appear in: A. Mitrakas et al. (Eds.) : Secure E-Government Web Services. Idea Group, Inc, 2006.
6. R. Breu et al. : Web Service Engineering - Advancing a New Software Engineering Discipline. In: D. Lowe, M. Gaedke (Eds.): Web Engineering, ICWE 2005, Sydney, Australia, July 2005, Proc. LNCS 3579 Springer 2005, ISBN 3-540-27996-2.

7. M. Hafner et al.: Modeling Inter-org. Workflow Security in a Peer-to-Peer Environment. In: R. Bilof (Ed.): Proc. of the 2005 IEEE International Conference on Web Services, ICWS 2005, Orlando, USA, 2005, IEEE Conference Publishing Services, ISBN 0-7695-2409-5.
8. T. Andrews et al.: Business Process Execution Language for Web Services, Version 1.1. , BEA, IBM, Microsoft, SAP, Siebel (2003).
9. OASIS: ebXML Business Process Specification Schema Version 1.01. OASIS, 2001.
10. M. Bernauer et al., "Comparing WSDL-based and ebXML-based Approaches for B2B Protocol Specification". In: Proc. of the 1st Int. Conf. on Service-Oriented Computing (ICSOC), Trento, 2003.
11. A. Arkin: Business Process Modeling Language - BPML1.0, Working Draft, 2002.
12. N. Mitra, N.: SOAP Version 1.2 Part 1: Messaging Framework. W3C Rec. 24/06/2003.
13. Eastlake, D. et al., (eds.): XML-Signature Syntax and Processing. W3C Rec. 14/02/2002
14. Eastlake, D. et al. (eds.): XML Encryption Syntax and Processing. W3C Rec. 10/12/2002.
15. T. Moses et al. (eds.): XACML Profile for Web-Services. XACML Working draft, v. 4/09/2003.
16. P. Mishra et al. (eds.): Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. Committee Draft 02, 24/09/2004.
17. S. Bajaj, et al., :Web Services Policy Framework (WS-Policy). 09/2004.
18. E. Bertino, S. Castano, E. Ferrari: Securing XML Documents with Author X. In: IEEE Internet Computing 5 (2001) 3, pp. 21-31.
19. D. W. Chadwick: RBAC Policies in XML for X.509 Based Privilege Management. In: Proc. of the 17th Int. Conf. on Information Security: Visions and Perspectives 2002
20. http://www.eu-trustcom.com/
21. P. Grefen et al.: CrossFlow: Cross-org. Workflow Man. in Dynamic Virtual Enterprises. In: Int. Journal of Computer Systems Science & Engineering 15 (2000) 5, pp. 277-290.
22. F. Casati and M. Shan: Event-based Interaction Management for Composite E-Services in eFlow. In: Information Systems Frontiers 4 (2002) 1, pp. 19-31.
23. M. Dijkman, M. Dumas: Service-Oriented Design: A Multi-Viewpoint Approach. Int. Journal of Cooperative Information Systems 13(4): 337-368 (2004).
24. E. Gudes, M. Olivier, R. van de Riet.: Modelling, Specifying and Implementing Workflow Security in Cyberspace. In: Journal of Computer Security 7 (1999) 4, pp. 287-315.
25. W.K. Huang, V. Atluri: SecureFlow: A secure Web-enabled Workflow Management System. In: ACM Workshop on Role-Based Access Control 1999, p. 83-94.
26. J. Wainer et al.: W-RBAC – A Workflow Security Model Incorporating Controlled Overriding of Constraints. In: International Journal of Cooperative Information Systems. 12 (2003) 4, pp. 455-485.
27. M. Alam et al.: Model Driven Security for Web Services. In: Proc. of the 8th International Multi-topic Conference (INMIC 2004), IEEE, Lahore, 2004.
28. http://www.oracle.com/technology/products/ias/bpel/index.html
29. J. Mendling, M. Hafner: From Inter-Organizational Workflows to Process Execution: Generating BPEL from WS-CDL. In conf. and workshop proc. MIOS 2005 . R. Meersmann et al. (Eds.): LNCS 3762, pp. 305-315, 2005.
30. K. Mantell: From UML to BPEL. IBM-developerWorks, 2003.