Model Driven Security for Inter-Organizational Workflows in e-Government

Ruth Breu[1], Michael Hafner[1], Barbara Weber[1], Andrea Novak[2]

[1] Universität Innsbruck, Institut für Informatik, Innsbruck, 6020, Österreich
{ruth.breu, m.hafner, barbara.weber}@uibk.ac.at

[2] Austrian Research Center Seibersdorf, Seibersdorf, 2444, Österreich
andrea.novak@arcs.ac.at

Abstract. Model Driven Architecture is an approach to increase the quality of complex software systems by creating high-level system models and automatically generating system architectures and components out of these models. We show how this paradigm can be applied to what we call Model Driven Security for inter-organizational workflows in e-government. Our focus is on the realization of security-critical inter-organizational workflows in the context of web services and web service orchestration. Security requirements are specified at an abstract level using UML diagrams. Out of this specification security relevant artifacts are created for the target reference architecture based on upcoming web service security standards.

1 Introduction

E-government refers to the use of the Internet and other electronic media to improve the collaboration within public agencies and to include citizens and companies in administrative processes. A core aim of e-government is to bring about a digital administration in order to enhance quality of service (e.g., additional online information or service offerings) as well as efficiency (e.g., reduced case processing times, fewer errors or using fewer resources to accomplish the same task).

The implementation of e-government solutions is a very complex task that can only succeed if IT-experts and domain experts co-operate with each other at a high level of abstraction right from the beginning. Security issues rooted in provisions and regulations play a very critical role. These include security requirements of public law (i.e., Austrian Signature Act [1] and the Austrian E-Government Act [2] as well as the Federal Act concerning the Protection of Personal Data [3]), the Austrian Security Manual [4], the OECD Guidelines for the Security of Information Systems and Networks [5] and internal security requirements of the municipalities.

Security requirements must not be considered as an isolated aspect, but during all stages of the software development cycle [6], [7]. As the engineering of security into the overall software design is often neglected, different approaches for integrating security in the system development cycle have been proposed [8], [9]. Nevertheless, they do not yet exploit the potential of a model driven approach.

Model driven software development is particularly appealing in the area of security as many security requirements adhere to certain categories (e.g., integrity) and can be described in implementation-independent models. In most cases, the development of security-critical systems is based on a set of well-known counteractive measures (i.e., protocols, algorithms) for which the correctness has been proved.

In this paper we give an overview of our approach to the model driven realization of security-critical inter-organizational workflows in the context of web services and web service orchestration. The description of security requirements is performed at a high level of abstraction. Security relevant artifacts are generated for a target architecture. A detailed description of the different aspects can be found in [10], [11].

Our approach provides a specification framework for the design of collaborating systems in the context of the platform-independent web service technology. It also supports the systematic transition from security requirements, via the generation of security artifacts, to a secure solution based on a web services platform. The specification of security requirements is performed in a platform-independent way and can thus be applied by domain experts without in-depth technical knowledge.

The structure of the subsequent sections is as follows. After providing an overview on web services composition, web services security and Model Driven Architecture in Section 2, we present a case study in Section 3, and describe our model driven approach in Section 4. Finally, section 5 gives an overview of related work and section 6 draws the conclusion.

2 Backgrounds

2.1 Web Services Technology

The emergence of web services technologies together with workflow composition languages allows for an easier and platform-independent collaboration between partners (.e.g.: governmental and local authorities). WS-BPEL [12] is a workflow composition language for web services and provides support for abstract business protocols and executable business processes. A business protocol specifies the public message exchange between parties and abstracts from how they are internally processed, while an executable business process models the behavior of a partner in a specific business interaction. WS-BPEL is an appropriate top layer standard to the web services protocol stack, including WSDL [13], SOAP, UDDI, WS-Transactions [14] and related standards. An alternative standard to describe business protocols is WSCI [15]. BPML [16] can be used to model executable business processes.

2.2 Web Services Security

As web services are often composed to carry out complex business transactions, not only the web service itself has to be secured, but also the message exchange between different web services. WS-Security [17] specifies a mechanism for signing and encrypting SOAP messages, and is used to implement message integrity and

confidentiality. It also supports the propagation of the authentication information in the form of security tokens (e.g., Kerberos tickets or X.509 certificates). XACML [18] provides access control mechanisms and policies within documents, while SAML [19] represents authentication and authorization decisions in XML format and is used to exchange this information over the internet (e.g., to support single sign-on).

2.3 Model Driven Architecture and Security

Model Driven Architecture (MDA) is an approach for the design and the implementation of applications that aims at cost reduction and application quality improvement [20]. At the very core of MDA is the concept of a model (i.e., abstraction of the target system). MDA defines two types of models, a Platform Independent Model (PIM), describing the system independently from the intended platform, and a Platform Specific Model (PSM) describing the system on its intended platform (e.g., J2EE or .NET). The process of converting a PIM into a PSM is called transformation. Models are described using a well-defined modeling language such as UML. Model Driven Security is based upon MDA in the sense that security requirements are integrated into design models, leading to security design models. Transformation rules of MDA are extended to generate security infrastructures [20].

3 Case Study

Our methodology for the systematic design and realization of security-critical inter-organizational workflows is illustrated by a portion of a workflow drawn from the use case "Processing of an Annual Statement" (Figure 1) describing the interaction between a business agent (the Tax Advisor) and a public service provider (the Municipality).

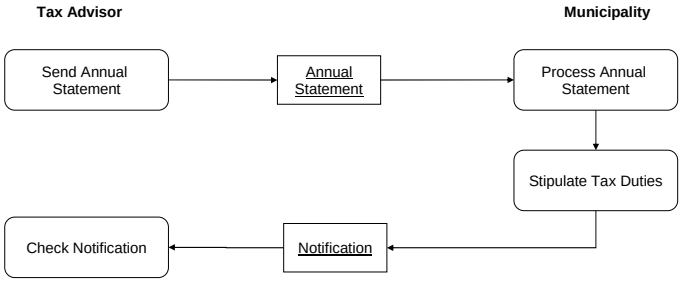

Fig. 1. Processing of an Annual Statement (Portion of Workflow)

The use case was elaborated within the project SECTINO, a joint research effort between the research group Quality Engineering at the University of Innsbruck and the Austrian Research Center Seibersdorf. It is based on a case study involving a major Austrian municipality. The project aims at the development of a framework supporting the systematic and efficient realization and management of innovative e-government related workflows with a special focus on security requirements.

In Austria, all wages and salaries paid to employees of an enterprise are subject to the municipal tax. Businesses have to send the annual tax statement via their tax advisor to the municipality which is responsible for collecting the tax by the end of March of the following year. The municipality checks the declaration of the annual statement and calculates the tax duties. A notification with the amount of tax duties is then sent to the tax advisor by mail. Ultimately, the workflow should allow the declaration of the municipal tax via the internet.

One of the project goals is to analyze security issues that may stem from the migration of the workflow to an e-government based solution and create the necessary run-time artifacts for the target architecture through model transformation.

4 Model Driven Security for Inter-Organizational Workflows

In this section we present our approach to the management of security related aspects within the development process (Figure 2).

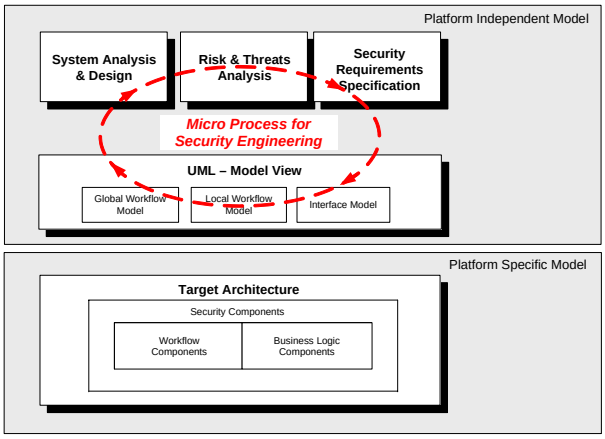

Fig. 2. Model Driven Security for Inter-Organizational Workflows

The development of security-critical inter-organizational workflows starts with the analysis and the design of the workflow, followed by a risk and threats analysis, and the security requirements specification (Section 4.1). Security requirements are then modeled in a platform-independent way at different levels of abstraction (Section 4.2). These four steps are executed iteratively, following a five step approach for security analysis called Micro Process for Security Engineering. The requirements are finally transformed into run-time artifacts for the target architecture (Section 4.3).

4.1 Security Analysis

Security related aspects within the development of inter-organizational workflows are tackled by a five step approach as illustrated in Figure 3 [9].

Table 1. Sample Scenario of a Security Analysis at the Global Level

1. The data exchange within the "Processing of an Annual Statement" has to comply with the requirements of integrity and confidentiality.
2. This workflow is open to the threat that a third unauthorized party may try to read and to modify the exchanged data.
3. The probability of occurrence is estimated as medium, the possible damage is estimated as substantial.
4. The measures to counter the threats involve encryption and digital signatures.
5. The proposed measures are checked. There remains the requirement that the two partners have to authenticate each other.

The Micro Process for Security Analysis is performed at different levels of abstraction (i.e., at the global, the local and the component level). Requirements and measures are explored and described at the appropriate level of detail based on given artifacts (e.g., the global workflow model). Table 1 illustrates the security analysis process using a sample scenario (Section 3).

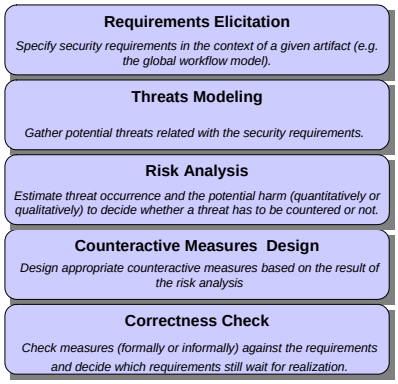

Fig. 3. The Micro Process for Security Analysis

In the early phases of design, security requirements are expressed in a textual way (e.g., by a security relevant section within the use case specification). In the context of the UML notation we provide extended notation techniques. Security requirements are related to each other so that they can be traced from one level of abstraction to the next (i.e., each requirement is transformed into one or several requirements or into some counteractive measures at the abstraction level underneath).

4.2 Model View

In the context of this paper a *workflow* describes a network of partners cooperating in a controlled way by calling services and exchanging documents. Our method of designing security-critical inter-organizational workflows is based on two orthogonal views: the *interface view* and the *workflow view*. The latter is further divided into the

global workflow model describing the message exchange between cooperating partners, and the local workflow model describing the behavior of each partner. The application of these orthogonal perspectives allows us to combine the design of components offering services that may be called in different contexts.

This paper focuses on *programming in the large* [21] and assumes the business logic itself to be given. As there is no central control of the process, the inter-organizational workflow is designed by representatives of the partners involved in the workflow. Actions are allocated to specific partners in the global workflow model. Every action corresponds to some business logic implemented at a partner node.

Very often partners have already implemented some kind of application logic, maybe even made it accessible to customers as a web service. In this case, the development of an inter-organizational workflow requires an inside-out proceeding. The interface of the application logic is projected onto the interface model. The interface model of every partner's node describes the public part of the local application logic, which is accessible to the inter-organizational workflow and conforms to a uniform technical, syntactical and semantic specification the partners agreed upon. If for example, the partners agree to implement the global workflow based on web-services, some partners will have to provide a web services wrapper for their application logic; they may decide on parameter formats, interaction protocols, operation semantics or run-time constraints specification, information is typically published in WSDL files and technical Models (tModels) of UDDI Registries. Accordingly, in an outside-in proceeding, the interface model represents a specification of the functional requirements the partner has to implement at its node.

From a security perspective, the interface model deals with security requirements from the components' point of view, while the workflow model deals with the secure exchange of documents between different partners. In the sequel we briefly sketch the different models, followed by an explanation of the inter-model dependencies.

Global Workflow Model. The global workflow model describes an integrated abstract view of the workflow involving partners in autonomous organizations. The global workflow describes the interaction of partners abstracting from internal processing steps and does not contain any connection to the business logic.

The global workflow is modeled as an UML 2.0 activity diagram [22] describing sequences of actions, which represent web services. For each action, we specify which partner is calling the service, which partner offers the services and which documents are exchanged. We enrich the activity diagram of the global workflow by a qualification of the document exchange with security requirements. Figure 4 shows the document exchange between two public service providers requiring compliance to the security requirements of confidentiality and integrity. Security requirements are modeled in compliance with the UML 2.0 Metamodel by attaching a constraint, to the *ObjectNode*. The *ValueSpecification* of the constraint consists of attributes assigned a set of element nodes corresponding to the document parts to be encrypted and signed.

In our example, every part of the document sent from the Tax Advisor to the Municipality is meant to comply with confidentiality and integrity. At runtime, the Policy Enforcement Point, acting as a security gateway, will have to sign and encrypt the document at the company's boundary according to a security policy configuration file containing the above mentioned requirements (section 4.3).

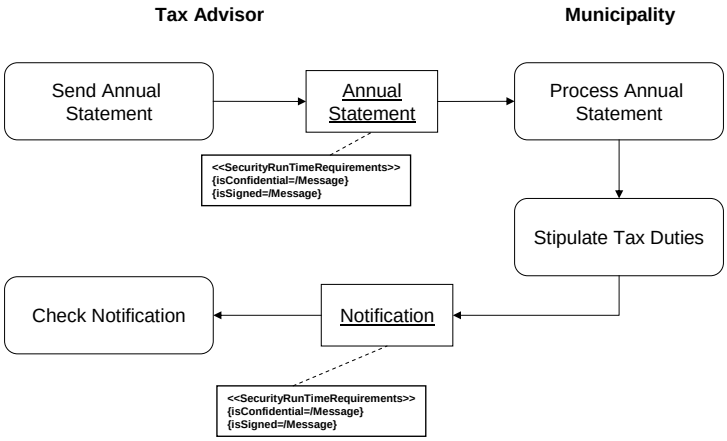

Fig. 4. A Sample Document Flow with Security Requirements (Global Workflow Model)

Local Workflow Model. The local workflow models define the portion of the global workflow each partner is responsible for. They are developed for each partner type. The local workflow is an executable process description that considers service calls from the outside, and contains internal actions as well as connections to the business logic. It is a direct input for a local workflow management system and is typically developed internally by partners. Referring to the sample process, the global workflow model captures the protocol between the online municipal tax component and the involved partners like the Municipality and the Tax Advisor, while the local workflow model describes the sequence in which the component accepts and processes incoming messages based on the services described in the interface model. The local workflow model describes the necessary processing steps to calculate the tax duties. These steps are performed internally and are invisible to the outside.

Interface Model. The interface model describes a component offering a set of services with given properties and permissions. Security requirements at this level of abstraction involve the support of a role model and the specification of access rights for particular web service operations. We describe access rights formally and platform-independently using OCL, a predicative sublanguage of UML [23]. The OCL specification is then transformed into an XACML-policy file via automatic generation. A more detailed description of the interface model can be found in [10].

Model Dependencies. Security requirements specified in the global workflow model have to be mapped in a consistent way to the local workflows of all cooperating partners, which reflect the business logic in their local environment.

Partner A in Figure 5 is responsible for the implementation of the business logic covering Actions 1, 2, 5 and 6 in the global workflow model. This can be seen as an abstract functional specification of the application logic a partner has to contribute to the global workflow. All the partners together agree on the signature format and naming conventions for the interfaces they provide to each other. These interfaces are visible to all partners and represent entry or exit points for data, messages or

documents, either entering the local workflow for further processing or leaving it after processing (e.g., OP_1 (Msg1):= Msg4 in Figure 5).

In a second step, the partners map the interfaces of their local business logic to operations in the Interface Model (e.g., $LocalOP_B$ (LocalMsg1) in Figure 5). They are not visible to the partners and are used during the execution of their own local workflows in order to perform additional workflow actions.

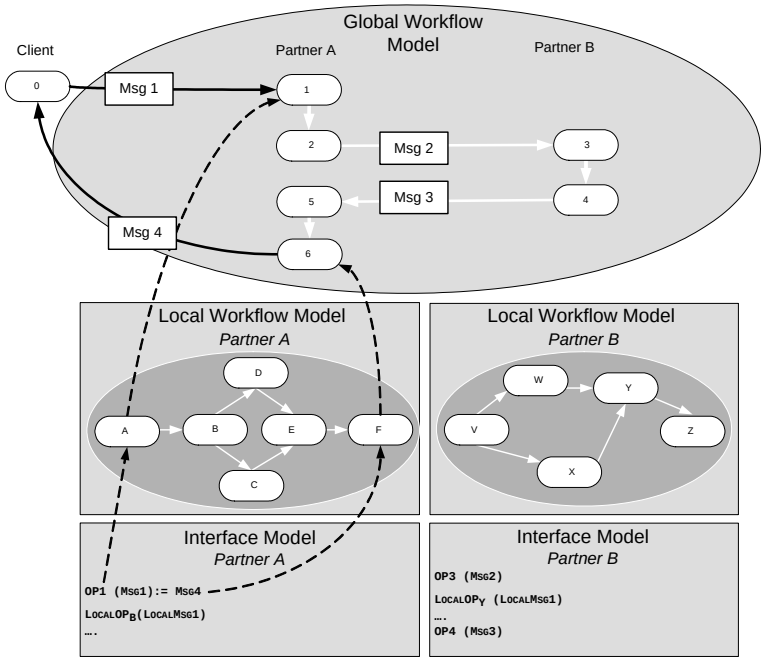

Fig. 5. Interface, Global and Local Workflow Model

In the global workflow model, either one or two actions are mapped to an operation in the interface model depending on whether the message exchange is asynchronous or synchronous. Van der Aalst et al. [24] present a formal approach based on Petri nets for the design of inter-organizational workflows guaranteeing local autonomy without compromising the consistency of the overall process. In our terms, this means that - in a peer-to-peer fashion – the local workflows should exactly realize the behavior as specified in the global workflow.

3.3 Target Software Architecture

In this section we present our target architecture for a partner which offers a portion of a distributed workflow. The architecture is based on the data-flow model of XACML as described in [18]. Figure 6 shows the software architecture in the view of a partner who implements his portion of the global workflow as a local workflow and offers an interface to its partners.

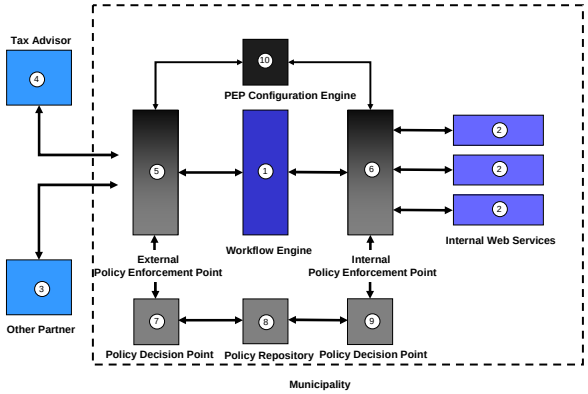

Fig. 6. Target Software Architecture

The core component is the workflow engine (1), which implements a choreography language such as WS-BPEL [12] or BPML [16] and aggregates and controls the sequence of existing Web services (2) to a composition that may be offered as a web service of its own to external business partners (3) and (4).

Table 2. Security Objectives and their Implementation in the PEP Component

Security Requirement	Security Component	Provided Functionality	Used Technologies & Standards
Authentication	External PEP Policy Decision Point Policy Repository	SOAP Firewall (Message Structure Processing) SOAP Firewall (Auhtorization Policies) Policy Archive	SAML, WS-Sec, XML-Encr.,XML-Sign, PKI, WSS4J, XML XACML, XML XACML, XML
Authorization	Internal / External PEP Policy Decision Point Policy Repository	SOAP Firewall (State-Dependent Permission Check,Mapping Global to Local Access Rights) SOAP Firewall (Auhtorization Policies) Policy Archive	SAML, WS-Sec, XML-Encr.,XML-Sign, PKI, WSS4J, XML XACML, XML XACML, XML
Confidentiality	External PEP PEP Configuration Engine	SOAP Firewall Check of Complimnace to Security Requirements	SAML, WS-Sec, XML-Encr.,XML-Sign, PKI, WSS4J, XML PKI, WSS4J, XML
Integrity	*refer to "Confidentiality"*	*refer to "Confidentiality"*	*refer to "Confidentiality"*
Non Repudiation	*refer to "Confidentiality"*	*refer to "Confidentiality"*	*refer to "Confidentiality"*

SAML	Security Assertion Markup Language (SAML)
XACML	Extensible Access Control Markup Language (XACML)
Sign	XML Digital Signature
Encr	XML Encryption
WS-Sec	Web-Services Security Specification
XSD	XML Schema Definition
WSS4J	Web-Services Security for Java
XML	Extensible Mark up Language
PKI	Public Key Infrastructure

Our prototypical generator is based on WS-BPEL, a workflow composition language for Web services. WS-BPEL is an appropriate top layer standard to the web services protocol stack, including WSDL [13], SOAP, UDDI, WS-Transactions [14] and a multitude of related standards. We use BPWS4J as a BPEL engine [25].

In order to provide a trusted domain atomic and composite web services are wrapped by security components. The Policy Enforcement Points (PEP) act as security gateways. The external PEP (5) implements security objectives like user authentication, confidentiality and integrity regarding data exchange with external

partners, whereas the internal PEP (6) enforces access rights. It checks invocation requests from workflow partners to exposed services and then forwards requests to Policy Decision Points (PDP) (7) and (9) which check the requests according to some policy stored in the Policy Repository (PR) (8). The PEP Configuration Engine (10) supports the configuration of the security components acting as a repository for XML files that provide specific instructions to the Policy Enforcement Points.

Security requirements in the workflow models are implemented through basic concepts like keys, encryption, signatures and certificates based on XML and SOAP. Access control requirements are expressed in XACML, which is an XML based OASIS standard for a policy and access control decision language [19]. Table 2 gives an overview of the security requirements, the security components, the functionality the components provide and the underlying standards and technologies.

5 Related Work

Related work can be found in several areas. A number of approaches deal with secure document exchange and workflow management in a centrally organized environment. Among these are the Author-X system [26], PERMIS [27], and Akenti [28]. Often a central control is appropriate, but there are also many application domains requiring a local organization.

A whole community deals with inter-organizational workflow management systems [24], [29], [30], [31], [32], [33]. We do not aim to contribute a novel approach to this field. Instead, we rely on UML models for modeling workflows and existing workflow management systems based on web services technology. Security extensions at a low level of abstraction for workflow management systems are treated in [34], [35], [36], [37].

Model driven approaches that are close to the idea of our framework are [38], [39], [40]. Lodderstedt [20] introduced the notion of Model Driven Security for a software development process that allows for the integration of security requirements into system models and supports the generation of security infrastructures. These approaches deal with business logic, our approach deals with workflow management.

6 Conclusions and Further Studies

In this paper we have given an overview of our approach to Model Driven Security of inter-organizational workflows. Our framework is based on the idea of specifying security requirements at the abstract level of UML models and generating security components in the context of web services, web service orchestration and the upcoming related security standards.

At this stage, we have finished specifying the underlying concepts and evaluating adequate standards. Extension efforts are heading for several directions. We are working on the transformation functions and on an implementation of the core of the code generator for a run time architecture based on web services technology. We aim at the extension of the list of security requirements that can be expressed within our

syntactic framework by considering additional features for basic security requirements (e.g., the distinction of documents signed by actors or by systems according to various legal requirements) and by introducing new types of complex, domain specific security requirements (e.g., transactional security requirements for electronic banking). Our case studies in the field of e-government show us an increasing demand for high-level development of secure workflow realizations.

References

1. Austrian Signature Act (Signaturgesetz - SigG), Art. 1 of the Act published in the Austrian Federal Law Gazette, part I, Nr. 190/1999.
2. Federal Act on Provisions Facilitating Electronic Communications with Public Bodies (E-Government Gesetz - E-GovG), Art. 1 of the Act published in the Austrian Federal Law Gazette, part I, Nr. 10/2004, entered into force on 1 March 2004.
3. Federal Act concerning the Protection of Personal Data (Datenschutzgesetz - DSG2000), published in the Austrian Federal Law Gazette, part I No. 165/1999, on 17. August 1999.
4. Austrian Security Manual, http://www.cio.gv.at/securenetworks/sihb/
5. OECD Guidelines for the Security of Information Systems and Networks, http://www.ftc.gov/bcp/conline/edcams/infosecurity/popups/OECD_guidelines.pdf.
6. P. Devanbu, S. Stubblebine: Software engineering for security: a roadmap. In: A. Finkelstein (ed.): The Future of Software Engineering, ACM Press 2000, pp. 227-239.
7. E. Ferrari, B. Thuraisingham: Secure Database Systems. In: M. Piattini and O. Díaz (eds.): Advanced Databases: Technology Design, Artech House, London 2000.
8. A. Hall, R. Chapman: Correctness by construction developing a commercial secure system. In: IEEE Software 19 (2002) 1, pp. 18-25.
9. The authors: Towards a Systematic Development of Secure Systems. In: Information Systems Security 13 (2004) 3.
10. The authors: Towards Model Driven Security of Inter-Organizational Workflows. Accepted for SAPS 2004.
11. The authors: Modeling and Realizing Security-Critical Inter-Organizational Workflows. In: W. Dosch, N. Debnath (Eds.): Proceedings IASSE 2004, ISCA, ISBN 1-880843-52-X, 2004.
12. BEA, IBM, Microsoft, SAP AG, Siebel Systems: Specification: Business Process Execution Language for Web Services Version 1.1, May 2003, http://www.ibm.com/developerworks/library/ws-bpel.
13. E. Christensen, F. Curbera, G. Meredith, S. Weerawarana: Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl.
14. BEA, IBM, Microsoft: Web services Transaction (WS-Transaction), http://www-6.ibm.com/developerworks/webservices/library/ws-transpec/
15. BEA, Intalio, Sun Microsystems, SAP: Web Service Choreography Interface (WSCI) 1.0, http://www.w3.org/TR/wsci/
16. A. Arkin: Business Process Modeling Language. San Mateo, CA: BPMI.org, 2002. Proposed Final Draft.
17. IBM, Microsoft, VeriSign: Web services Security (WS-Security), http://www-106.ibm.com/developerworks/webservices/library/ws-secure/
18. S. Godik, T. Moses: eXtensible Access Control Markup Language (XACML) Version 1.0 3 OASIS Standard, 18 February 2003, http://www.oasis-open.org/committees/xacml/repository
19. S. Cantor, J. Kemp, E. Maler: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, Last-Call Working Draft 17, 13 July 2004,

http://www.oasis-open.org/committees/download.php/7737/sstc-saml-core-2.0-draft-17.pdf
20. T. Lodderstedt: Model Driven Security: from UML Models to Access Control Architectures. Dissertation, Univ. of Freiburg 2003.
21. F. Leyman, D. Roller: Production Workflow: Concepts and Techniques. Prentice-Hall 2000.
22. UML: http://www.uml.org.
23. OMG, UML 2.0 Superstructure Specification, http://www.omg.org/docs/ptc/03-08-02.pdf.
24. W.M.P. van der Aalst and M. Weske: The P2P appraoch to Interorganizational Workflows. In K.R. Dittrich, A. Geppert, and M.C. Norrie (eds.): Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01), Springer, Berlin, pp. 140-156.
25. IBM: BPWS4J, http://www.alphaworks.ibm.com/tech/bpws4j.
26. E. Bertino, S. Castano, E. Ferrari: Securing XML Documents with Author X. In: IEEE Internet Computing 5 (2001) 3, pp. 21-31.
27. D. W. Chadwick: RBAC Policies in XML for X.509 Based Privilege Management. In: Proceedings of the IFIP TC11 17th International Conference on Information Security: Visions and Perspectives 2002, pp. 39-54.
28. M. Thompson, A. Essiari, S. Mudumbai: Certificate-based Authorization Policy in a PKI Environment. In: ACM Transactions on Information and System Security 6 (2003) 4, pp. 566-588.
29. W.M.P. van der Aalst.: Loosely Coupled Interorganizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries. In: Information and Management 37 (2000) 2, pp. 67-75.
30. W.M.P. van der Aalst: Process-oriented Architectures for Electronic Commerce and Interorganizational Workflow. In: Information Systems 24 (1999) 8, pp. 639-671.
31. Z. Luo, A. Shet, K. Kochut, J. Miller: Exception Handling in Workflow Systems. In: Applied Intelligence 13 (2000) 2, pp. 125-147.
32. P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig: CrossFlow: cross-organizational workflow management in dynamic virtual enterprises. In: International Journal of Computer Systems Science & Engineering 15 (2000) 5, pp. 277-290.
33. 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.
34. V. Atluri, W.K. Huang: Enforcing Mandatory and Discretionary Security in Workflow Management Systems. In: Proceedings of the 5th European Symposium on Research in Computer Security 1996.
35. 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.
36. 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.
37. J. Wainer, P. Barthelmess and A. Kumar: W-RBAC – A Workflow Security Model Incorporating Controlled Overriding of Constraints. In: International Journal of Cooperative Information Systems. 12 (2003) 4, pp. 455-485.
38. 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.
39. U. Lang.: Access Policies for Middleware. PhD Thesis, University of Cambridge 2003.
40. T. Lodderstedt, D. Basin, J. Doser: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: J.-M. Jézéquel, H. Hussmann, S. Cook (eds.): Proceedings of the 5th International Conference on the Unified Modeling Language, Springer 2002, pp. 426-441.