# A Framework for Seamless and Compliant Service Consumption in Outsourcing Scenarios

A. Schaad, P. L. Miseldine, U. Flegel, C. Wolter
SAP Research, Vincenz-Priessnitz Str. 1, 76131 Karlsruhe

**Abstract.** The ability to support the outsourcing of parts of a business process has become a major requirement within enterprise information systems (EIS). If, however, certain procedural or technical constraints derived from compliance regulations have been specified on parts of such a process we face the problem of maintaining such constraints in outsourcing scenarios as well. In this paper we describe a technique and architectural support for flexible outsourcing of process steps at runtime, at the same time supporting the service provider in delivering the required evidence regarding his compliant handling of the outsourced artefacts. We demonstrate the augmentation of business processes with compliance constraints for later model-driven and automated generation of security and configuration policies. An architecture is then presented detailing the working of a compliance-centric service gateway, supporting the outsourcing of constrained activities irrespective of the service provider's EIS landscape.

**Keywords:** Compliance, Security, Business Process, Service

## 1 Introduction

Compliance regulations exist to ensure that the assets and operations of a business are correctly managed according to ethical and social norms as well as legal regulations. As a result, businesses must set objectives that control their operations [1], [2] with the goal of reaching compliance to appropriate regulations. These objectives are assessed by auditors [3] to prove the correct management of the business. With many enterprises reliant on complex IT solutions to manage their day to day operations, such as an ERP, it follows that many of these control objectives relate to operational, and behavioural, functionality of software. Precisely defining the behaviour of complex systems however is notoriously difficult. Similarly, enforcing corrective behaviours on non-compliant software, is a difficult and taxing issue.

In an attempt to provide flexibility in software, essential business activities are commonly abstracted into business processes, with behaviours of software exposed and encapsulated as services [4]. Business processes orchestrate these services to provide the business activity, but are defined in a flexible fashion, such that the process can be manipulated when either the activity needs refinement, or the services change. In relation to compliance therefore, if a software-based realisation of a business activity is assessed as being non-compliant and in need of correction, business processes provide a layer in which such changes can be made that can affect services. Similarly, the need to convey and relate compliance concerns to actual services can be in part realised through constraints placed on the business process.

### 1.1 Problem Scope

When outsourcing parts of business activities we face three key requirements with respect to the enterprise information system supporting the realisation of the overall processes in question:

1) Traceability of compliance constraints must be supported at the business process level.
2) There should be no impact on the technical environment of the business process.
3) Compliance demands must be met by the service provider and observable by the consumer.

Looking at current technical approaches all three requirements are not realistically met. The topic of addressing risk and compliance at a business process-level has only been recently addressed e.g. [5], with the serious shortcoming of not relating to platform dependent enforcement, ie. generation of appropriate service configurations and constrained invocation of activities and supporting services.

We thus demonstrate an approach to specifying compliance constraints at a business process level. Being model-driven in its nature we can transform such constraints into selected business process execution environments (eg. handling of separation of duty constraints at process runtime [6]) or middleware components (eg. using Axis Rampart for ensuring Business Object Security [7]).

Regarding the replacing of services we face the problem of doing this in a seamless fashion considering our supporting infrastructure. As soon as we do exchange an internal service associated to a step in a business process (e.g. a task in BPEL or BPMN [8]), we have to regenerate service bindings and proxy configurations wrt the new service identified as part of the outsourcing process. This situation is even worse when considering possible instances of a process and the effect of exchanging services at runtime.

Thus, in order to avoid any costly reconfiguration or redeployment wrt running process instances we introduce the concept of a compliant-centric service gateway (CCSG) that will interpret compliance constraints on the business process and seamlesly invoke the outsourced activities. In essence, reconfiguration of redeployment will not be required even for running process instances.

Compliance constraints specified on the business process level as well as any model-driven realisation and enforcement at a procedural or infrastructure level should be equally supported by the service provider.

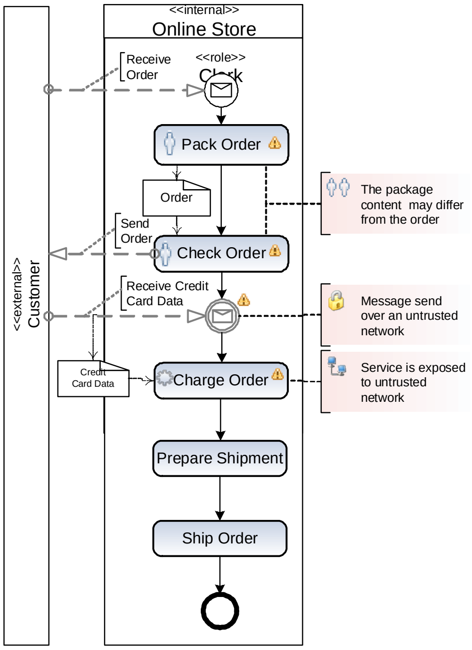

Figure 1: Compliance-driven Process Annotation

The CCSG equally supports that any compliance constraints on outsourced process steps are relayed to the service provider as part of the business objects subject to the outsourcing scenario.

### 1.2 Paper Structure

Accordingly, the paper is structured as follows. In section 2 we focus on the representation of compliance constraints in business processes. Section 3 will then describe the proposed protocol and architecture supporting the outsourcing of such annotated processes, focusing on the seamless service consumption through the proposed compliant-centric service gateway (CCSG). This includes a discussion on the functioning of our proposed relaying protocol. Section 4 sketches how we envision service providers to provide evidence on having met any constraints on the outsourced processes, before providing a discussion of our approach and related work in section 5.

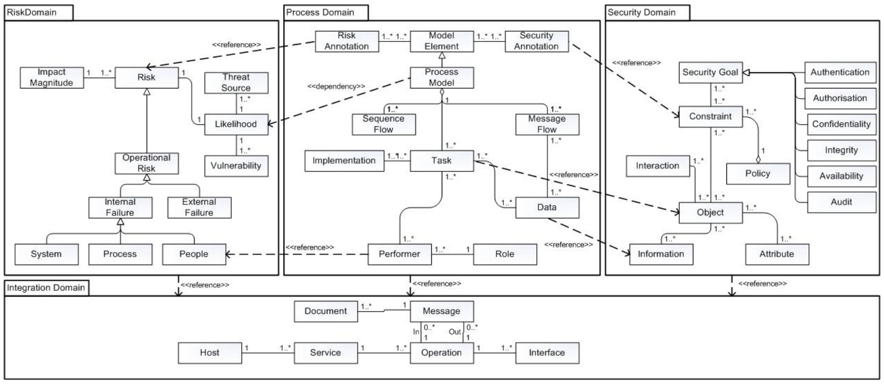

Figure 2: Crosscutting Compliance Dependencies

## 2 Model-driven Compliance Modelling

### 2.1 Risk and Security Modelling

In [9], Zachman discussed the concept of enterprise models that describe and classify the logical structure of an enterprise to aid in the management of enterprise system development. One of these modelling concepts defines process models that formalise and annotate the complex behaviour of isolated applications and services, while effectively merging technology, functional, and behavioural aspects of an enterprise into a single model. Being well considered in both research and industry, process management has led to a number of process modelling standardisation efforts, such as BPEL4WS [8], BPMN [8], XPDL [8], WS-CDL [10], or UML 2.0 AD [11]. Nowadays, most process modelling standards have an appealing graphical notation. The basic elements of such models are easy to understand and grasp. Security domain experts who are not very familiar with the details of process modelling are assumed to easily understand such diagrams.

Many compliance-related operational risks are tightly coupled to functional and behavioural aspects of processes, providing an explicit relationship between them. Thus, it is important to position and consume the existing risk knowledge in the wider context of these process models. It stands to reason therefore, that by consuming knowledge of existing risks and interweaving them with process models, it is possible to query and detect risk related to service orchestrations from a process-based point of view, rather than from the low abstraction level of a technical viewpoint. As an example, consider the simple online purchasing scenario shown in Figure 1.

This process model contains informal textual operational risk annotations in business processes derived from a suitable risk knowledge base, such as presented in [12]. This scenario describes a simple purchasing process initiated by a customer sending an order to a store's clerk. The core process consists of five steps, including manual packing of the order, manual checking if the package fits the original order and and automated charging by credit card when the customer submits his credit card data before final shipment.

A people-based operational risk is related to the manual tasks of packing and checking the order, because the package may not contain the ordered items either by purpose or by accident. This example also contains a network related risk in terms of message confidentiality, e.g. credit card data eavesdropping, and a system-based failure in terms of availability due to a potential DoS attack.

### 2.2 Risk and Security Annotations

To provide risk and security awareness for process management, defined risks and security policies must be linked to process model entities and their semantics. In the domain of service-oriented architectures the service integration domain represents the fundamental layer other enterprise domain concepts build on. For instance the process domain describes the choreography of former isolated services to business processes, referencing service operations as tasks. Similar, a risk model relates the assets it describes to services, hosting system, or messages exchange. But there exist also dependencies to other domains, such as the process domain. In this case the likelihood of some risks occurring is influenced by the process model definition itself. Consider a people related risk based on committing fraud, if two service operations can be performed by the same person. If the process model is designed in such a way, that it is not possible that the same person can perform both tasks in a single process at runtime the likelihood is reduced. These dependencies are sketched out in Figure 2. As a fourth domain security is part of this Figure. Also this domain relates to entities of the other domains such that interacting objects can be related to service operations, process performers, or data exchanged. These domain models are independent from a concrete platform implementation and on purpose abstract to support later model-driven generation of system configurations and policies.

### 2.3 Security Policy Generation

According to the model-driven paradigm our extended modelling environment should support business process experts and security experts to work collaboratively on an abstract model. As discussed, process modelling notations are applicable by providing a general look-and-feel that makes it easy to be understood by diagram modellers and any viewer of the diagram [8]. Also, most of the modelling notations provide extensibility features, such as text artefacts, to add additional information bits into the diagram.

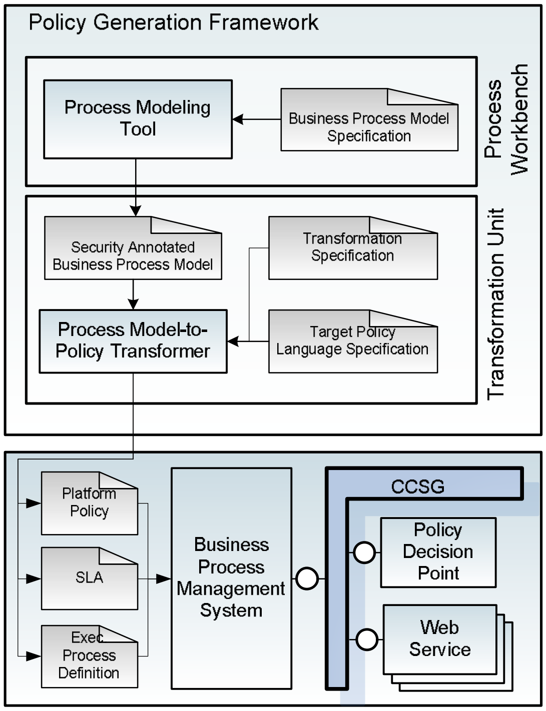
Policy Generation Framework
Process Workbench
Process Modeling Tool
Business Process Model Specification
Transformation Unit
Security Annotated Business Process Model
Transformation Specification
Process Model-to-Policy Transformer
Target Policy Language Specification
Platform Policy
SLA
Exec Process Definition
Business Process Management System
CCSG
Policy Decision Point
Web Service

Figure 3: Policy Generation Framework and Architecture

In Figure 2 we defined platform-independent security domain models abstracting from concrete implementation platforms and their provided security features. In order to generate a concrete policy (Figure 3) we must link the information derived from the process model, its security annotations and their contained constraints to a platform specific model. In the domain of service-oriented architectures, established standards such as WS-Security, XML Encryption, SAML or XACML are suitable [13].

To translate the security annotation into a platform specific policy, a mapping must be defined. This mapping is based on elements that have been identified as referring to the same set of security goals, for example confidentiality. In a simple way, entities of different models must be referring to the same real world concept, so there is a bijective correspondence between instances of the two entities. Therefore, in our current prototypes these transformations are based on a set of transformation rules that translates elements into a concrete security policy by applying transformation patterns, for example XSLT scripts [14]. In general, transformation rules will specify the set of elements in the process model which match a particular pattern that then will be transformed into instances of another pattern in the concrete security policy model.

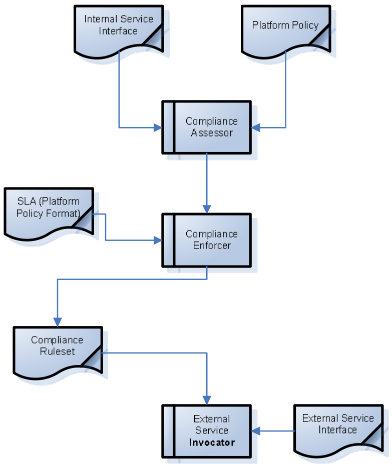
Internal Service Interface
Platform Policy
Compliance Assessor
SLA (Platform Policy Format)
Compliance Enforcer
Compliance Ruleset
External Service **Invocator**
External Service Interface

Figure 4: Compliance-centric Service Gateway (CCSG) Protocol

## 3 Protocol

When choosing to outsource a business activity, new constraints will often be required as new compliance regulations will now be relevant to the activity. For example, data once originally treated as internal, when sent to an external, third party, could be required to be anonymised. As described previously, such constraints can be represented on the business process itself, and mapped to platform-specific policies. In the case of service outsourcing, only the policies applicable to the outsourced activity should be exposed, and this exposure must be complete such that the outsourcer is aware of all constraints governing its usage. Thus, rather than choosing to expose the whole platform-specific policy, a filter of relevant aspects must be made. To communicate the platform-specific policies towards the service level, the services utilised by the business processes will require interface changes that result from the additional constraints. These interface changes are needed as all constraints governing the service and data sent to the service must be fully conveyed to the outsourcer so that it can plan appropriate mechanisms for processing this data, and provide evidence back to the consumer. When outsourcing a business activity, the encapsulated behaviour of a service on a business processing level often remains the same; the same behaviours of the internal service are simply now required to be hosted externally. As such, changes on the business process workflow are mainly non-functional in nature, and do not typically introduce new tasks or sequences. Instead, the interface of the outsourced service will consist of a different service interface than that of the internal service, though represent the same behaviour. Much work exists that allows such relationships to be made, such as the behaviour reification method CA-SPA [16], or semantic-based descriptions [17].

Changes to service interfaces however, require that the business process service bindings be adapted to suit the new service. This can be problematic, especially considering that long running processes must be restarted if the underlying process definition is changed. Thus, a mechanism that can support enrichment of a service call with compliance related constraints, whilst maintaining the same service interface, will minimise the impact of choosing outsourced services from the aspect of compliance, and business process management. To this end, in this section we introduce a methodology for seamlessly replacing an internal service with a gateway service that keeps the same interface though conveys the constraints described on the business process to the outsourcer.

### 3.1 Compliance-Centric Service Gateway

A service interface commonly consists of a set of methods, with each containing a set of parameters. When presented with two different services that encapsulate the same behaviour, relational mappings between each can be made, as defined in established research on the subject. For our purposes however, we require a service that also includes compliance constraints that have been mapped and transformed to a policy. It is therefore required that a service gateway be defined that can expose itself using the same interface defined by an internal service to be outsourced, though locate the policies specific to this service, and convey these to the outsourced service.

Figure 4 shows a base architecture and protocol of such a gateway (based on positioning of the CCSG in Figure 3). As input from the internal system, it requires internal service interface in the form of a WSDL description, and a platform policy derived from the business process. This platform process is analysed via the compliance assessor through identifying references made within the platform policy to the service defined in the service interface. The compliance assessor produces a refinement of the policy that it then forwards to the compliance enforcer component. This component interprets the policy, and represents it in a format suited for the outsourcer, the compliance ruleset. This format is defined via an agreed Service Level Agreement that specifies how the outsourcer expects security constraints to be provided to it. The final component, the external service invocator, exposes the service using as input the internal service interface, and marshals the compliance ruleset and this call to the service defined by the outsourcer, or the external service interface. In this regard, a service gateway is created specific for each service required to be outsourced. It is exposed using the same interface as the internal service; however it calls the external, outsourced service, attributing the necessary constraints. This requires that when the platform-specific policies are generated from business processes, they are stored in a repository that can be queried by the compliance enforcer component.

The use of this gateway has numerous benefits. If the outsourced service interface changes, or its requirements for constraints changes (via SLA), then no changes are required to the business process that is used to invoke the service, as all calls are routed through the service gateway which itself updates its behaviour. Similarly, if new constraints are added to the business process, the repository will receive an updated policy, which is used for subsequent calls to the outsourcer via interpretation of the compliance ruleset.

## 3. Use Case

In this section, the operation of the proposed CCSG is illustrated by elaborating the use case of the online purchasing scenario from Section 2. In that scenario we identified several candidate process tasks suitable for outsourcing. First, we may outsource the tasks of packing and checking an order. Second, the handling of credit card payments are frequently outsourced to specialised payment providers. Third, the delivery of the goods is usually turned over to a shipping company.

For each of these tasks we can examine various compliance requirements, with discussion starting on the explicitly modelled compliance requirement on the human-related risk of fraud when packing and checking orders. As specified in the process domain an order may not be checked by the same person that as packed the goods for the order. The rationale behind this is raising the bar for fraud, where a person may omit goods from the package for their own profit, or to add goods to packages for their own orders. Such fraud could not be detected, if the same person were to check the fraudulent package, leading to a requirement for Separation of Duties (SoD) to be specified. When outsourcing these tasks of the online purchasing scenario the service provider needs to be informed that their customer, the online store, requires the tasks to be subject to SoD rules. To this end, the outsourced tasks Pack Order and Check Order are invoked via the CCSG ensuring proper argument mediation. Before calling the remote task the Compliance Enforcer component of the CCSG enriches the call with a ruleset expressing the SoD requirement in the terms of the service provider (plus further rulesets for compliance with requirements not explained here).

The packaging service provider must be prepared to interpret the SoD ruleset and provide adequate resources for implementing them. In this case, this implies that sufficient personnel are available for different people to package, and checking them. Since personnel that pack and check orders authenticate themselves to the system before receiving order item lists, the system can generate evidence attesting to which human task was performed by which person. Generating such evidence is subject to additional policies provided by the invoking party, and is encapsulated within objects that represent data such as order item lists. The invoked service (packaging service) then interprets these policies to generate the requested evidence and relays it back to the invoking party (online store) as part of the modified data objects [15].

This scenario is limited in that it only makes some of the compliance requirements explicit. In the following we consider compliance of the discussed process to privacy law within the European Union.

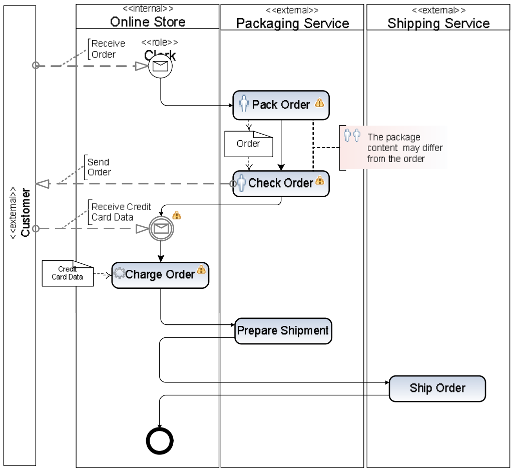

Figure 6: Use Case Scenario

We base our assumptions on the German implementation of the European Directive 95/46/EG into national law. As a summary of this law, we consider it to protect the way in which personal data may be handled to provide a service on a contractual basis. Being the owner of the personal data, here the customer, the data subject needs to be informed about the data used in the process and how it may be used by the data controller, here the Online Store and its subcontractors. This knowledge is on a strict need-to-know basis, strictly for the purposes where it is necessary for service provision. For this example we restrict ourselves to the following data: order number, ordered items, customer address, and customer payment details. It is obvious that the combination of some of the data, such as customer address and ordered items, can be clearly undesirable, depending on the nature of the goods and the recipients of the data. As an example, certain political, religious or philosophical beliefs are outlawed in some countries and ordering books or DVDs expressing such beliefs may be dangerous to the customer if in violation to national regulations. Another example would be goods giving away the medical condition of the customer, who would not want his employer to be informed about his condition.

For packing and checking orders, only the ordered items list and possibly the order number are strictly required and used as parameters when invoking the respective tasks. For logistical reasons however, the packaging provider will also hand over the package to the shipping provider, and in doing so requires the customer address. Here, the CCSG will enrich the invocation of the shipping preparation task with a ruleset expressing that the address may not be used in conjunction with the order items list. Additionally, the data objects are supplemented with evidence generation rules, expressing to generate evidence on access to the order items list, and the customer address. When the evidence is relayed back to the Online Store it can be checked whether the order items list and the customer address were access in the same context, giving rise to compliance verification [14].

## 4. Discussion

Our work relates to accepted model-driven approaches [18, 20, 21], however specifying compliance artefacts at the business process (BPMN) level has only been recently described in [19] with a description of a specific transformation of separation of duty constraints [1, 2, 6] into XACML policies [14]. This gave rise to the question of this paper on how to address outsourcing of parts of such constrained processes. Being aware of hard technical limitations such as the absence of trusted policy enforcement by service providers [15], we thus concentrated on the behaviour and actual protocol of a Compliance-centric Service Gateway.

The approach presented in this work has numerous benefits, as well as numerous drawbacks. Whilst the use of the CCSG and the accompanying architecture can minimise the impact of outsourcing on the business process, this implies that clear encapsulation of behaviour was indeed mapped to the service level. If the behaviour of outsourced service differs, the business process will require adaptation in any case. The approach also assumes the isolation of compliance regulations relevant to a business process is made, and thus allows the inference of compliance rules applicable to the service. By their nature, compliance objectives are high level and business centric as they must relate to the business objectives they aim protect and assure. Accordingly, they do not explicitly relate to low-level, technical implementation, and require an assessment to be performed in collaboration between business experts, and technical experts. Whilst solving these issues is out of scope for this paper, major efforts are underway to simplify this process [22].

Future work will now focus on an implementation of our design in the context of a business process management system based on BPMN [23]. The model-driven software architecture of this system as well as its ability to directly execute BPMN make it an ideal candidate.

## 5. Related Literature

1. Taylor, H. “The Joy of SoX- Why Sarbanes-Oxley and Services Oriented Architecture may be the best thing that ever happened to you.” Wiley & Sons, 2006
2. Schaad, A. “A Framework for Organisational Control Principles”, PhD Thesis. Department of Computer Science, University of York, 2003
3. Menzies, C. (Hrsg.) (2004): “Sarbanes-Oxley Act. Professionelles Management interner Kontrollen.” Schäffer-Poeschel, Stuttgart 2004.
4. Gustavo Alonso, Fabio Casati, Harumi A. Kuno, Vijay Machiraju: “Web Services - Concepts, Architectures and Applications” Springer 2004
5. Muehlen, M. zur; Rosemann, M.: “Integrating Risks in Business Process Models.” Proceedings of the 2005 Australasian Conference on Information Systems (ACIS 2005). Manly, Sydney, Australia, 29. Nov.–2. Dec. 2005. Manly, Sydney, Australia.

6. A. Schaad, K. Sohr and M. Drouineaud "A Workflow-based Model-checking Approach to Inter- and Intra-analysis of Organisational Controls in Service-oriented Business Processes" Journal of Information Assurance and Security, 2007
7. Securing SOAP Messages with Rampart - http://ws.apache.org/axis2/modules/rampart/1_0/security-module.html (last access 30/07/2008)
8. Weske, M. “Business Process Management.” Springer, Berlin, 2007
9. Zachmann, J. “A framework for information systems architecture” IBM Systems Journal Volume 26 , Issue 3 (1987)
10. Gero Decker, Hagen Overdick, Johannes Maria Zaha: On the Suitability of WS-CDL for Choreography Modeling. EMISA 2006: 21-33
11. Nick Russell and Wil M. P. van der Aalst and Arthur H. M. ter Hofstede and Petia Wohed „On the suitability of UML 2.0 activity diagrams for business process modelling” Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling - Volume 53, 2006
12. Chonawee Supatgiat, Chris Kenyon, and Lucas S. Heusler. Cause-to-Effect Operational Risk Quantification. In IBM Research Report. IBM Research Division, January 2006.
13. Mohammad Ashiqur Rahaman, Andreas Schaad "SOAP-based Secure Conversation and Collaboration", ICWS 2007, Salt Lake City, Utah
14. Wolter, C., Schaad, A. "A TRANSFORMATION APPROACH FOR SECURITY ENHANCED BUSINESS PROCESSES", SE 2008, Insbruck, Austria 2008
15. Miseldine, P, Flegel, U., Schaad, A. ”Supporting Evidence-Based Compliance Evaluation for Partial Business Process Outsourcing Scenarios” First International Workshop on Requirements Engineering and Law (RELAW), Barcelona, 2008
16. Miseldine, P, Taleb-Bendiab, A. “CA-SPA: Balancing the Crosscutting Concerns of Governance and Autonomy in Trusted Software”. 20th International Conference on Advanced Information Networking and Applications, 2006.
17. Xu, M., Chen, J., Peng, Y., Mei, X., and Liu, C. “A Dynamic Semantic Association-Based Web Service Composition Method” In Proceedings of the 2006 IEEE/WIC/ACM Conference on Web intelligence December, 2006.
18. Jan Jürjens. UMLSec: “Extending uml for secure systems development.” In UML ’02: Proceedings of the 5th International Conference on The Unified Modeling Language, pages 412–425, London, UK, 2002. Springer-Verlag.
19. Christian Wolter, Andreas Schaad "Modeling of Task-Based Authorization in BPMN", 5th International Conference on Business Process Management, 2007
20. David Basin, Jürgen Doser, and Torsten Lodderstedt. “Model driven security for process-oriented systems.” In SACMAT ’03: Proceedings of the 8th ACM Symposium on Access control Models and Technologies, pages 100–109, New York, NY, USA, 2003. ACM.
21. Alam, M. and Hafner, M. and Breu, R. (2008) “Constraint Based Role Based Access Control in the SECTET-Framework - a Model-driven Approach.” Journal of Computer Security, 16 (2). ISSN 0926-227X
22. Project MASTER: Managing Assurance, Security, and Trust for Services. EU funded public project FP7-216917. http://www.master-fp7.eu
23. Niemann, F. “Projekt Galaxy: SAP baut Netweaver zur SOA-Middleware aus“ http://www.computerwoche.de/knowledge_center/erp/1856279/