Incorporating Security Requirements from Legal Regulations into UMLsec model

Shareeful Islam[1], Jan Jürjens [1, 2]
[1]Institut für Informatik, Technische Universität München, Germany
[2] Department of Computing, the Open University, Great Britain
islam@in.tum.de, http://www.jurjens.de/jan

Abstract. Compliance with law, industry standards, and corporate governance regulations are one of the driving factors for discovering security requirements. This paper aims to incorporate constraints from regulations through security requirements at an early stage of development. Constraints are extracted using a pattern based approach from legal texts of information security laws and policies derived from the security standard ISO/IEC 27001:2005. The UML extension UMLsec is then used to address whether the security requirements defined in a UMLsec model implement these constraints successfully.

Keywords: Security requirements, model based security engineering, information security law, UML sec, ISO/IEC 27001:2005.

1. Introduction

For software which processes critical information, security requirement needs to align with prevailing relevant laws and other regulations to control non-compliance issues. Identifying relevant regulations, interpreting key elements from legal texts and addressing these by defining suitable security requirements is challenging: Legal texts contain numerous ambiguity, cross reference, domain specific definition which make it hard to analyze and interpret [11]. Characteristics of information security and data privacy laws are relatively new, unstable and not available from all security domains. When these laws fit with national law then stakeholder make their interpretations based on business goal and objective. This different level of interpretations and tailoring makes legal text more difficult to interpret. Current practices on security requirements emphasize more on security threat on critical asset, risks associate with this threat, impact and mitigation strategies ([9, 10]). These approaches do not offer any guideline how to trace regulations.

This paper aims to integrate legal constraints from information security regulations and policies from control clauses of ISO/IEC 27001:2005[4] into a security requirements engineering process. These legal constraints interpret the law to derive actors, possible actions, objects and purpose [12]. Legal texts from European Commission (EC) information society legislations are considered to derive legal constraints [5]. Finally Model-based Security Engineering with UMLsec is used to trace legal constraints to security requirements [7]. The approach integrates within Requirement Engineering (RE) activities, so that a requirement specification document will include a complete set of security requirements derived from legal constraints.

There has been a lot of previous work towards security modeling, including security design models [13-16] and security requirements elicitation [17,18]; however, this work does not yet seem to have been applied to address legal constraints (with the exception of [3], which however does not trace them through to the design model stage). Our approach explicitly incorporate legal constraints from relevant regulations when security requirements are elicit. These legal constraints are then address by elicit requirements in design phase by using UMLsec.

2. Legal Aspect of Information Security

Rapid technology advancement has transformed the business infrastructure to depend more on software to process, store and transmit confidential information. This infrastructural change implies new conditions for legal regulations to manage this information. At the same time, software systems have become more complex, extensible, and distributed, making protection of the digital information difficult. Despite using the latest security techniques and protocols, most software systems still face many security breaches. These security breaches can cause a breach of regulations by violating policies, or security requirements such as privacy, integrity, non-repudiation etc. Non-compliance can lead to financial penalties, loss of brand reputations, customer dissatisfaction, etc. Thus IT security risk assessment needs to take into account information security regulations such as data privacy, quality of electronic signature, security of processing, etc at the early stage. But regulations relating to information security are relatively new and evolving [11]. Due to different cultural and historical traditions, sometimes even the same legal requirements may be inconsistent between different countries. For instance, legislative and regulatory regime in E-commerce may cause incompatibilities particularly for any cross border transaction.

The current work considered legal text from EC information society directives. These directives are legal guidelines for information security laws. For instance directive 95/46/EC sets up a framework for ensuring protection of personal data and free movement of the data. In the current work, we considered articles from this directive as a case study to derive legal constraints. The legal text from different articles of these directives needs to be interpreted and analyzed to derive constraints. Security requirements need to consider these constraints before proceeding with the next phase of the development.

3. Obtaining Security Requirements

We shortly explain our process for obtaining security requirements. It is an asset-based risk driven approach for identifying security requirements similar to processes that have been previously proposed [2,3,9,10]. The process consists of a set of iterative activities. Due to space limitation, only a summary of the approach is described here, which integrates with the usual RE activities. Thus, first the initial RE phases (requirement elicitation and analysis) provide an understanding of the business context and the identified artifacts (such as a view of the system, its operative

environment, initial use cases, business scenarios, business goal and risk, stakeholder requirements etc.) The security requirements process then starts based on these initial RE artifacts. Initially, assets, services and later also security scenarios (misuse cases, threat models, attack trees etc) can be identified based on these RE artifacts. Relevant information security laws and industry standard (if required) are also identified at this stage. Draft security goals and policies are then defined to protect asset and service for business continuation to meet the business goal. Security goals and policies also need to be driven by the relevant regulations. The next phase identifies threat, vulnerability to asset, and services. Threat identification depends on prior identification of possible vulnerabilities. Security artifacts are then developed based on possible threats and vulnerability. For instance, the initial business use case scenarios can instantiate into misuse cases. Attack trees and abuse cases can be identified based on some interventions that lead to policy violation, non-compliance assumption etc. Details of vulnerabilities for non-compliance and policy violations are used later to derive legal constraints and trace regulations. A complete security risk management is then performed to identify possible risks, analyze their impact, and develop mitigation strategies to control the risk. These mitigation strategies help to revise security goals and policies, and to elicitate the security requirements. At this stage of the security requirements process, legal constraints are derived from identified information security law. Pattern based approach is used to derive legal constraints from legal text of information security law. Relevant policies based on the control clauses from different sections of ISO/IEC 27001:2005 are identified to support legal constraints from the security goal and policy documents. Model-based security engineering with UML and UMLsec is then used to verify how design models extended with security requirements can fulfill constraints from regulations and policies from standards to meet regulatory compliance. Finally, security requirements are reviewed, prioritized, and integrated with other requirements for the software requirement specification document.

4. Model-based Security Engineering for Addressing Regulations

The idea of Model-based Security Engineering (MBSE) is to construct a relatively abstract model for the system. Different artifacts such as system views, business scenarios, stakeholder requirements, misuse cases, attack trees etc can be used to construct the model [6,7]. Finally, this abstract model is used to derive implementations by using automatic code generation or by manually creating the code and generating test-sequence to verify it against the model. Here, we consider the first part of MBSE (from requirements to models) for tracing legal constraints. Security requirements, legal constraints and policies are used as specification elements within UMLsec and UML diagrams that model the system. It is then possible to check whether security requirements properly address constraints from the relevant regulations.

UML offers rich extension mechanism such as stereotypes, tagged values, and constraints in the form of labels, which have been used to define UMLsec [7]. Stereotypes and tags are used to formulate security requirements. Finally, constraints can be attached that have to be satisfied by modeling elements with the particular stereotype. UML diagrams such as use cases, sequence diagrams, deployment

diagrams etc together with the UMLsec stereotypes describe various views in different parts of a security-critical system. For instance, use case diagrams identify security requirements represented in stereotypes attached to interactions between actors and use cases. Deployment diagrams express the physical layer of a system, so that the security requirements at the logical layer can be compared to the security levels provided at the physical layer. Security mechanism, protocols, devices etc can be coordinated using deployment diagrams to analyze the security or the overall system. Sequence diagrams can specify interactions between different parts of the system. In this paper, we focus on use cases and deployment and sequence diagrams because these are particularly relevant during the requirements engineering phase.

We use stereotypes as security requirements, tags for different properties of the security requirement and legal constraints, and policies as constraints in UMLsec. Two different pattern-based approaches (targeted to activities resp. purpose) are used to identify legal constraints [12]. Activity pattern defined as a subject who performs an action (right, obligation) on an object. Activity pattern identified properties of legal constraints such as actors, actor's right and obligation on actions to an object.. Purpose-based pattern describe possible motivation or reason for the action. We use a goal-oriented approach where high level goal are derived for any action. Similarly, policies give guidelines for the actions to comply with control clauses of the relevant standard.

Table 1 shows an example of MBSE for privacy requirements to trace constraints. Actors, rights, obligations, and policies are considered based on article 16 and 17 from directive 95/46/EC for personal data protection. A detailed description of the table is given in the next section through a case study. Use case diagrams can include the <<privacy>> stereotype as a security requirement for different states of sensitive data such as flow, storage, access and process. The stereotypes <<secure links>>, <<secure storage>>, and <<data security>> with associates tagged values contribute to ensuring privacy as well. For instance, <<secure links>> for data flow requires the tagged value {authenticity} with values actor and data. When data transmits through a public network, then the authenticity of an actor is required at the destination end, but this identity must be hidden from the adversary during the transmission. Legal constraints and policies are then defined in order for rights, obligations, and guidelines required for this action to comply with the regulations. Table 1 also shows different technical measure such as security protocols, mechanisms etc that contribute to enforce privacy requirements. The technical measures must take into account security architectures, mechanisms, and other security requirements related to privacy requirements [1]. For instance, privacy requirements must be consistent with identification requirements and authorization requirements. How ever such measure requires consistence among level of security to the risk for breach of privacy and nature of data to be protected with the implementation cost.

5. Case Study

This case study is based on a business scenario which uses the security requirements process sketched above to identify security requirements, derive legal constraints from regulations, and trace constraints to security requirements using MBSE. We do not cover all details of the process due to space limitations. As the business scenario,

Table 1. Extension of UMLsec for tracing privacy requirements

Diagram & stereotype	Tags	Constraints	Tech. security measures
Use case <<privacy>>	{state = {dataflow}, {data storage}, {data access}, {data process}}	right = (access, disclose, process, transmit) obligation = (unauthorized access & disclose, unlawful process, accidental loss or destruction) purpose = (financial, social security, health) policy = (privacy , access control , data classification, data transmission, etc)	Security of data communication (cryptographic algorithm, key length), strengthens of password (length, minimum number of combination, life time) data classification (sensitivity, confidential, public) location of critical data, storage (plain text, encrypted), access control mechanism (mandatory, role based, optional) secure channel, authorization mechanism, etc.
Deployment <<secure links>>	{authenticity = actor, data} {confidentiality} {adversary = {type={read}, {logical condition}}	right = (access, process) obligation = (unauthorized disclosure & access, unlawful process) policy = (data transmission, cryptographic control, etc)	
Deployment <<secure storage>>	{authenticity = actor, data} {authorization = actor, right} {data storage = format} {adversary = {type = {access, process}, {logical condition}}	right = (access, disclose, process) obligation = (unauthorized access, unlawful process, accidental loss, or destruction) policy = (access control, user responsibility, password, data storage, data classification, data backup, etc)	
Interaction <<data security>>	{authenticity = actor, data} {authorization = actor, right} {confidentiality = data} {integrity = {actor, data}} {adversary = {type = {access, process}, {logical condition}}	right = (access, disclose, process, transmit) obligation = (unauthorized access, disclosure, unlawful alternation, controller consent) policy = (data classification, user responsibility, acceptable use of asset, access control, data transmission, cryptographic control, etc)	

we consider an online order processing system. Article 16 and 17 of directive 95/46/EC for protection of personal data is used to identify legal constraints. The privacy requirement is taken as the security requirement at hand, and different relevant policies are also considered in the case study.

Business Scenario

The customer is supposed to carry out the "order item" use case and a business actor is supposed to perform the "delivery item" use case. Product information, selection of product, features, order processing information etc are all available in online.

Relevant legal text

Directive 95/46/EC, Section Viii, Confidentiality and Security of Processing

Article 16, Confidentiality of processing

*Any person acting under the authority of the controller or of the processor, including the processor himself, who has **access** to **(personal data)**, **must not process** them except on instructions from the controller, unless he is required to do so by law.*

Article 17 (partial), Security of processing

*1. Member States **shall** provide that the controller **must** implement appropriate technical and organizational measures to protect **(personal data)** against **accidental or unlawful destruction** or **accidental loss, alteration, unauthorized disclosure or access**, in particular where the **processing** involves the **transmission of data** over a network, and against all other **unlawful forms of processing**. Having regard to the state of the art and the cost of their implementation,*

such measures shall ensure a level of security appropriate to the risks represented by the ***processing*** *and the nature of the data to be protected.*
2. The Member States ***shall*** *provide that the controller* ***must****, where* ***processing*** *is carried out on his behalf, choose a processor providing sufficient guarantees in respect of the technical security measures and organizational measures governing the processing to be carried out, and* ***must*** *ensure compliance with those measures.*

Article 16 and 17 describe confidentiality and security of personal data processing. The legal text is *italicized*, initial requirements key words (must, shall, etc.) are in ***bold***, actors is *underlined*, actions is in ***bold and underlined*** and object is ***(bold, underlined with first brackets)***. Now legal constraints are extracted based on activity and purpose pattern. These patterns are used to derive rights and obligations of actors to an object that govern a variety of practices supported by software systems. Actors are generally legal persons (subjects in legal text) who perform an action on an object. Actions (verbs in legal text) are possible activities on a data object. These actions are sometimes allowed (rights or permissions for the actor) and sometimes denied (obligation or restrictions for the actor). Objects may be pieces of information (personal data such as credit card number, pin code, birth date etc). The purpose is the goal of an action such as that the customer credit card number is used for a financial transaction. When legal text contains subject (actor), verb (action) and object, then pattern based approach try to identified actor's action right or obligation on an object and purpose for the action as legal constraints. Possible actors, activities, objects, and purposes of the legal text are given below.

Possible actors

- Data controller: Natural or legal personal person who determines the purpose and means to process and store data.
- Data processor: Natural or legal person nominated by controller who is responsible to monitor data processing by automated means.
- Data processing operator: Natural or legal person as a user or employee nominated by controller or processor to process, store data.
- Data subject: Natural or legal person to whom the personal data is related.
- Third party: Natural or legal person, agency, or any other body which processes personal data on behalf of controller.

Possible rights and obligations

- Right: Access, disclose, process, and transmit.
- Obligation: Unauthorized access & disclose unlawful process, accidental loss or destruction, controller consent.

Object: Data subject
Purpose: Financial, health care, social security, intentional.

Possible legal constraints

- C1=Data processor or data processing operator can process data subject's personal data when only instructed from data controller or required by law.
- C2=Data controller or third party protect personal data from accidental destruction or unauthorized access or disclose by sufficient technical measure.
- C3= If data processing involve transmission of data over a network then all unlawful from of processing need to control by sufficient technical measure.

- C4=Controller must ensure sufficient technical and organization measure before processing data.

Business scenario analysis
Actors and use cases from business scenario are now identified and a link is established with legal constraints.

Actors
- User: Personnel such as employee or business owner who use the system. Adversary can also be a user who tries to misuse or abuse the system. Data controller, processor, processing operator, third party, etc can be treated as user.
- Customer: Person who buys a product. It can also refer a data subject.

Use case
Use cases are actions by the actor on an object. Legal constraints are applicable to relevant use cases. Possible use cases are for example buying and selling goods, processing of personal data, notification of processing etc. Rights in use cases are for example access, process, disclose and transmit of data by different actors. Obligations in use cases are for example unauthorized access & disclose unlawful process, accidental loss or destruction, controller consent, etc by different actors.

Possible policies considering sections of ISO/IEC 27001:2005
ISO/IEC 27001:2005 specifies requirements for establishing, implementing, operating, monitoring, reviewing, maintaining and improving documented Information Security Management System (ISMS). We identified different sections of the standard such as 4.2.1, 4.2.3, 4.3, 6.a, 6.b, A.11, A12.1, etc relating to security and legal requirements. For instance section 4.2.1.b.2 state to include legal or regulator requirements for ISMS, A.11 specifies requirements for access control, etc and many more. Identified security policy need to derive from the guideline of these sections. Some possible policies we identified based on our business scenarios are access control policy, password policy, user authentication policy, data transmission policy, cryptographic control policy, data storage policy, data classification policy, data backup policy, acceptable use of asset, etc.

Use case diagram
This diagram is used to capture security requirements. Figure 1 shows a use case diagram describing the business scenario. Here the actor, as data subject with rights such as “view” and “process”, can order an item by giving private information. <<privacy>> is included as a stereotype for ensuring privacy requirements of data subject’s personal data (such as the card number, pin code, purchase details, address, etc). Tagged values are required to ensure privacy for different states (such as flow, storage, access, process, etc) of data. Rights, obligations and policies are mention in the figure to ensure privacy. Use case diagram focus on all possible rights, obligations and policies required for this scenario.

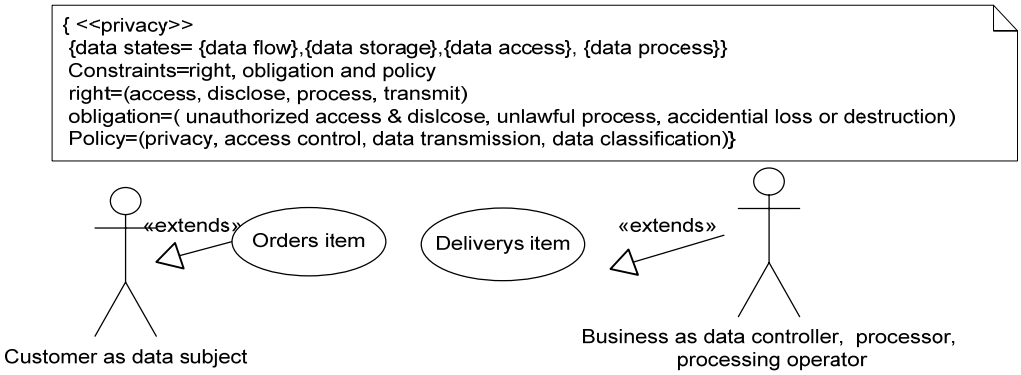

Fig. 1. Use case diagram for business scenario

Deployment diagram

Security requirements represented on the logical level by use case diagrams in fig 1 are now enforced by the level of physical security in deployment diagrams. This level includes the use of security mechanisms and protocols as a technical security measure (shown in table 1) to meet physical security. Continuing with the case study, the data subject's personal information (such as order history, address, etc) requires securing storage. Here we do not consider the credit card information stored at the data controller end. The data store needs to ensure <<secure storage>> in a manner that it is encrypted and that it requires proper authentication and authorization to access and process the data. The actor needs authentication and certain authorization rights to access and view the data. The actor must restrict from unauthorized access, processing and disclosing. The data controller needs to ensure rights and obligations for the actions with sufficient technical and organization measure.. Figure 2 shows the deployment diagram for the data processing operator machine as a client (with the data controller machine as the server). When the processing operator intends to access personal data, then rights and obligations need to ensure <<secure storage>>. For instance, data storage must be in a form (for example encrypted) so that an unauthorized system user cannot view the data. Data can be accessed only subject to an access control policy and legal constraints are enforced through sufficient technical measure.

Sequence diagram

The privacy requirement now extends to the interaction diagram. The interaction between customer and business are elaborated with sequence diagrams shown in figure 3. The data controller and data processor (as business) interact with the data subject (as customer), and store and process the data subject's personal data. This actor is also responsible for receiving and transmitting the data subject's personal data through public network. When a customer processes any order, then certain personal information (such as credit card number, pin code, order details, etc) is transmitted through a public network from the data subject to the data controller. The <<data security>> stereotype with certain tags such as authenticity, confidentiality, and integrity of data is required to protect the data from a possible adversary to meet the privacy requirements. For instance, an adversary might attempt an unauthorized access, disclosure, or processing of personal data through a man-in-the-middle attack

or may abuse the data for other purposes. The data controller responsible for receiving, storing and transmitting this data must take the necessary technical measures such data encryption to establish a secure channel and must follow the transmission policy, cryptographic control policy, data classification policy etc.

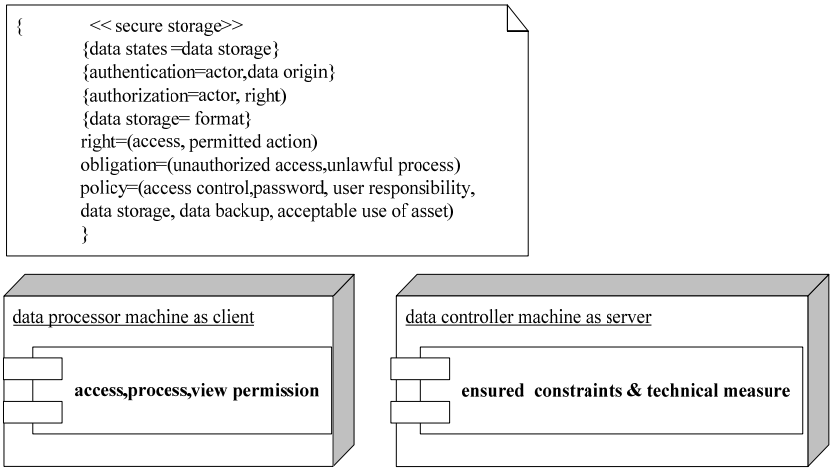

Fig 2. Deployment diagram for secure storage

6. Discussion

Legal text is always difficult to interpret. Our approach systematically interprets legal text from information security law to derive constraints. These constraints are one of the driving factors to elicit security requirements at early stage. Beyond the mentioned directive in the case study, this approach is applicable to interpret legal text for other EC information society directives. How ever some sections such as article 17 of 95/46/EC about sufficient guarantee in respect of the technical security measure is not properly interpret by our approach. Technical measure need to take into account cost, nature of object, risk level, business goal, etc with the security state, architecture, mechanisms, protocol etc. More over complexity, scalability, validity of the approach in an industrial context is not covered by this paper. The approach also need to extend throughout the development phases to address these legal constrains with provide tool support for building legally secure software through to the implementation level. We consider all these issues as future work of the paper.

7. Conclusion

The current paper proposes a security requirements engineering process integrating constraints from relevant regulations, and tracing these constraints through model-based security engineering. The goal is to consider regulatory compliance issues at early stage to build legally secure software.

*The work is partly supported by the German Academic Exchange Service (DAAD).

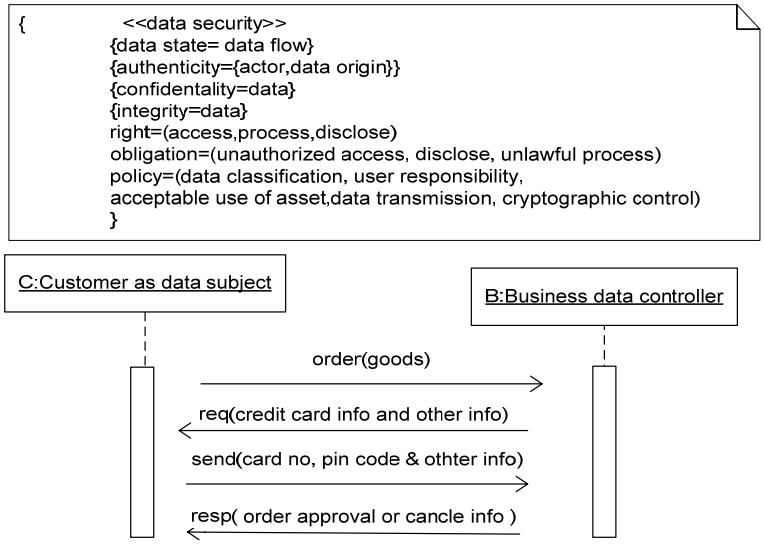

Fig 3. Sequence diagram for data transmission

References

[1] D. Firesmith. Engineering Security Requirements. Journal of Object Technology. Vol.2, No.1, 53-68, January-February 2003.

[2] D. Mellando, E.F. Medina, M. Piattini. A common criteria based security requirements engineering process for the development of secure information systems. Computer Standards & Interfaces 29 (2007), 244-253.

[3] F. Massacci, M. Prest, N. Zannone. Using a Security Requirements Engineering Methodology in Practice: The compliance with the Italian Data Protection Legislation. Technical Report DIT-04-103, 2004.

[4] INCITS/ISO/IEC 27001-2005: Information technology - Security techniques - Information security management systems Requirements, 2005.

[5] Information society, Summary of legislation, European Union.

[6] J. Jürjens. Using UMLsec and Goal Trees for Secure Systems Development, Proceedings of the 2002 ACM Symposium on Applied Computing.

[7] J. Jürjens. Secure System Development with UML. Springer 2004.

[8] J. Viega, G. McGraw, Building Security Software. Addison-Wesley, New York, 2001.

[9] C.B. Haley, R.C. Laney, J.D. Moffett, B. Nuseibeh: Security Requirements Engineering: A Framework for Representation and Analysis. IEEE Trans. Software Eng. 34(1): 133-153 (2008)

[10] N.R. Mead, E. D. Hough and T.R. Stehney. Security Quality Requirements Engineering (SQUARE) Methodology , CMU/SEI-2005-TR-009.

[11]P.N. Otto and Annie I. Antón, Addressing Legal Requirements in Requirements Engineering,, 15th IEEE International R. E. Conference,2007.

[12] T. Breaux, A. Antón. Analyzing Regulator Rules for privacy and Security Requirements, IEEE transactions on software engineering, Vol. 34, No. 1, January-February 2008

[13] S. H. Houmb, G. Georg, R.B. France, J.M. Bieman, J. Jürjens: Cost-Benefit Trade-Off Analysis Using BBN for Aspect-Oriented Risk-Driven Development. ICECCS 2005: p. 195-204.

[14] M. Alam, M. Hafner, and R. Breu, “Model-Driven Security Engineering for Trust Management in SECTET”, Journal of Software 2/1, Feb 2007

[15] J. Whittle, D. Wijesekera, M. Hartong: Executable misuse cases for modeling security concerns. ICSE 2008: 121-130

[16] S.T. Redwine, “Introduction to Modeling Tools for Software Security”, Build Security In body of knowledge, 2007

[17] H. Mouratidis, J. Jürjens, J. Fox: Towards a Comprehensive Framework for Secure Systems Development. CAiSE 2006: 48-62

[18] C.B. Haley, R.C. Laney, J.D. Moffett, B. Nuseibeh: Security Requirements Engineering: A Framework for Representation and Analysis. IEEE Trans. Software Eng. 34(1): 133-153 (2008)