

Developing Secure Networked Web-Based Systems Using Model-based Risk Assessment and UMLsec

Siv Hilde Houmb*

Norwegian University of Science and Technology

Jan Jürjens†

Software & Systems Engineering, TU Munich

Abstract

Despite a growing awareness of security issues in networked computing systems, most development processes used today still do not take security aspects into account. To address this problem, we designed a process for developing secure networked systems based on the extension of the Unified Modeling Language (UML) for secure systems development UMLsec and on the concept of model-based risk assessment (MBRA). Enterprise information such as security policies, business goals, policies and processes are supported through activities in the model-based integrated development process. These are then refined to security requirements at a more technical level, which can be expressed using UMLsec, and which can be analysed mechanically using the tool-support for UMLsec.

Keywords: *Security of web-based applications, e-commerce, risk assessment, enterprise modelling, model-driven design (MDA)*

1 Introduction

After many years of successful research in computer security technology, many security-critical networked systems developed in practise still turn out to be insecure. Part of the reason is that most often, security is not an integrated part of the system development process. While functional requirements are carefully analysed during system development, non-functional requirements, such as security, are often considered only after the fact. In e-commerce systems and other web-based services, security issues are of particular importance, due to the high risk of losing customers in a dynamic marketplace after security incidents. Since

such services are often developed and maintained in a networked enterprise setting, security issues need to be taken into consideration both at a technical and an enterprise level. In addition one should also emphasise cost issues and support the achievement of correct levels of security at the right time for the right price.

The IST-project CORAS [COR02] is based on the concept of model based risk assessment (MBRA) and have developed an integrated system development and risk management process aiming at security critical systems. The process is based on the Australian standard for risk management AS/NZS 4360:1999 [43699], RUP [Kru99] and RM-ODP [Put00] and focuses on handling security issues throughout the whole development process. In this article we extend the integrated process with two sub processes in order to support specification of the relationship between the enterprises involved in a networked enterprise setting. We also support specification of security requirements at a technical level by extending the CORAS approach with the use of a UML extension for secure systems development, UMLsec [Jür02, Jür03b]. The approach provides a precise way for specifying security requirements using UML [RJB99] and provides tool-support for a mechanical analysis of such requirements (see Section 6).

The usability of the CORAS approach has been evaluated during a set of six trials, three within the Telemedicine domain and three within the e-commerce domain. To validate the usability of the extended approach we have established a prototype lab at NTNU in Norway, where we create security critical systems using Lego Mindstorm, AIBO robots and different communication channels and computers with Windows operating systems (Windows XP at the moment). We plan to extend the lab with Intrusion Detect System (IDS) servers, Honeynet (<http://www.honeynet.org/>) and computers with other operating systems. However, in this article we will not discuss the result from this

*siv.hilde.houmb@idi.ntnu.no

†<http://www4.in.tum.de/~juerjens>

Stereotype	Base Class	Tags	Constraints	Description
secrecy	dependency			assumes secrecy
integrity	dependency			assumes integrity
critical	object, subsystem	secrecy, integrity		critical object
secure links	subsystem		dependency security matched by links	enforces secure communication links
secure dependency	subsystem		« call », « send » respect	structural interaction
data security	subsystem		data security provides data sec.	data security basic datasec requirements
fair exchange	subsystem	start,stop	after start eventually reach stop	enforce fair exchange

Figure 1. Some UMLsec stereotypes

project since they are rather preliminary, but rather focus on discussing the usability of the approach using an e-commerce example.

In Section 2 we describe the used fragment of UMLsec, while we in Section 3 describes model-based risk assessment and the CORAS approach. In Section 4 we presents the extended MBRA for networked enterprises, while we in Section 5 illustrates the approach using a small e-commerce example. Section 6 presents related work and Section 7 sums up the main issues from the paper. And finally, the appendix presents terminology for MBRA for networked enterprises.

2 UMLsec

We recall the fragment of UMLsec needed in our context. More details can be found in [Jür02, Jür03b] (see also [PJWB03] in this volume). UMLsec allows one to express security-related information within the diagrams in a UML system specification. The extension is given in form of a UML profile using the standard UML extension mechanisms. *Stereotypes* are used together with *tags* to formulate security requirements and assumptions on the system environment; *constraints* give criteria that determine whether the requirements are met by the system design.

Stereotypes define new types of modelling elements extending the semantics of existing types or classes in the UML metamodel. Their notation consists of the name of the stereotype written in double angle brackets « *»*, attached to the extended model element. This model element is then interpreted according to the meaning ascribed to the stereotype.

One way of explicitly defining a property is by attaching a *tagged value* to a model element. A tagged value is a name-value pair, where the name is referred to as the *tag*. The corresponding notation is {*tag=value*} with the tag name *tag* and a correspond-

ing *value* to be assigned to the tag.

Another way of adding information to a model element is by attaching *Constraints* to refine its semantics. Stereotypes can be used to attach tagged values and constraints as pseudo-attributes of the stereotyped model elements.

In Table 1 we give the relevant fragment of the list of stereotypes from UMLsec, together with their tags and constraints.

We shortly explain the use of the stereotypes and tags given in Table 1. More information can be found in [Jür02, Jür03b].

critical This stereotype labels objects that are critical in some way, which is specified in more detail using the corresponding tags. The tags are {**secrecy**} and {**integrity**}. The values of the first two are the names of expressions or variables (that is, attributes or message arguments) of the current object the secrecy (resp. integrity) of which is supposed to be protected.

secure links This stereotype on subsystems containing deployment diagrams is used to ensure that security requirements on the communication are met by the physical layer.

secure dependency This stereotype on subsystems containing static structure diagrams ensures that the «*call*» and «*send*» dependencies between objects or subsystems respect the security requirements on the data that may be communicated across them, as given by the tags {**secrecy**} and {**integrity**} of the stereotype «*critical*».

fair exchange This stereotype of (instances of) subsystems has associated tags *start* and *stop* taking names of states as values. The associated constraint requires that, whenever a *start* state in the contained activity diagram is reached, then eventually a *stop* state will be reached.

<p>Sub-process 1: Identify Context</p> <ul style="list-style-type: none"> • Activity 1.1: Identify areas of relevance • Activity 1.2: Identify and value assets • Activity 1.3: Identify policies and evaluation criteria • Activity 1.4: Approval <p>Sub-process 2: Identify Risks</p> <ul style="list-style-type: none"> • Activity 2.1: Identify threats to assets • Activity 2.2: Identify vulnerabilities of assets • Activity 2.3: Document unwanted incidents 	<p>Sub-process 3: Analyse Risks</p> <ul style="list-style-type: none"> • Activity 3.1: Consequence evaluation • Activity 3.2: Frequency evaluation <p>Sub-process 4: Risk Evaluation</p> <ul style="list-style-type: none"> • Activity 4.1: Determine level of risk • Activity 4.2: Prioritise risks • Activity 4.3: Categorise risks • Activity 4.4: Determine interrelationships among risk themes • Activity 4.5: Prioritise the resulting risk themes and risks <p>Sub-process 5: Risk Treatment</p> <ul style="list-style-type: none"> • Activity 5.1: Identify treatment options • Activity 5.2: Assess alternative treatment approaches
---	--

Table 1. Sub-processes and activities in the CORAS risk management process [HdBLS02]

3 Model-based risk assessment

Model-based risk assessment (MBRA) has been a research topic since the early 80-ies [KM87, GO84], and builds on the concept of applying system modelling when specifying and describing the systems to be assessed as an integrated part of the risk assessment. CORAS has based themselves on this concept and employs modelling methodology for three main purposes: (1) To describe the target of evaluation at the right level of abstraction, (2) As a medium for communication and interaction between different groups of stakeholders involved in a risk assessment, and (3) To document risk assessment results and the assumptions on which these results depend [HdBLS02, DRR⁺02].

Table 1 outlines the sub process and activities contained in the CORAS risk management process, which is a refinement of AS/NZS 4360:1999. Further information on the CORAS risk management process can be found in [HdBLS02].

The CORAS integrated system development and risk management process is based on the CORAS risk management process, RM-ODP and RUP. RUP structures system development according to four phases: (1) Inception, (2) Elaboration, (3) Construction, and (4) Transition. There are several iterations in each of the phases in RUP. In each of these phases security requirements using UMLsec are specified and analysed mechanically using tool-support for UMLsec, while risk is identified and evaluated using standard risk analysis methods supported by the CORAS framework. The identified risks in one iteration provides input to the security requirements specification in the next iteration.

In each of the iterations one assesses a particular part of the system, or the whole system at a particular viewpoint according to RM-ODP. One also identifies threats and propose alternative treatment for these. Treatments are evaluated according to a cost-benefit strategy.

4 Model-based risk assessment for networked enterprises

Risk assessment involves both technicians and non-technicians and one needs to provide easily understandable models along with precise and analysable descriptions. This is of particular importance in networked enterprise development settings.

In addition to providing support for specifying and analysing security requirements and identifying risks we need a precise definition of the terminology used within risk assessment of security critical systems. This ensures communication as well as preventing misconceptions among system developers, users, analysers and others involved in the development process. The ontology for model-based risk assessment for networked enterprises is based on the standards AS/NZS 4360 [43699], ISO/IEC 13335 [13300], ISO 17799 [17700] and IEEE 1471 [14700]. The most important terms in the context of networked enterprises will be explained in the following. Definition of the terms used are provided in Appendix.

The relationship between society, enterprise, networked enterprise, information system, target of assessment (ToA) and context is depicted in Figure 3.

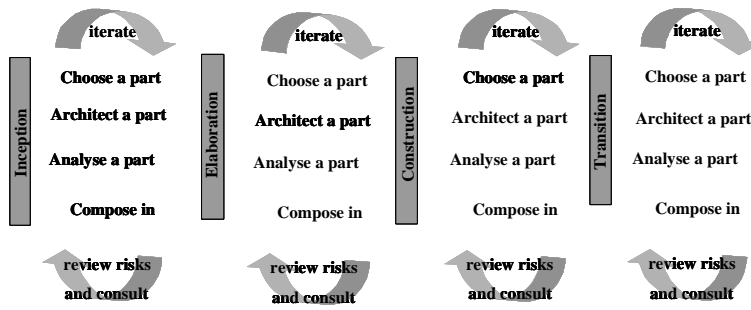


Figure 2. The integrated system development and risk management process

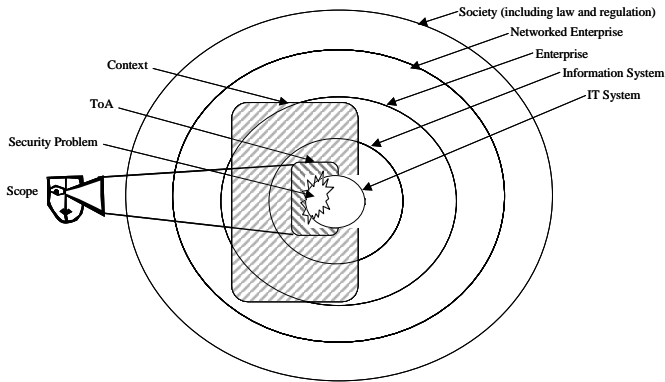


Figure 3. The relationship between ToA, context, enterprise, networked enterprise and society

The society represents the environment in which the networked enterprise is situated, including the law and regulations that each enterprise is controlled by. Enterprises exist within a society and relates to regulations enforced upon them by the society. Each enterprise represents an administrative unit within a society. Within one enterprise there may exist several information systems consisting of technical components, its related stakeholders and other conceptual components. The target of assessment relates to a subset of the components and stakeholders contained in the information system for which influences the target of assessment either directly or indirectly.

The risk management process for networked enterprises has been extended with two additional sub-processes: (1) Identify enterprises, and (2) Identify relationship between enterprises. The change is made to support specification of relations between different enterprises in a networked enterprise setting and to support the specification and analysis of security require-

ments and security mechanism using UMLsec. The activities in the two new sub-processes is as following.

Sub-process 1: Identify Enterprises

- Activity 1.1: Identify and describe each enterprise
- Activity 1.2: Identify and describe the security policy of each enterprise
- Activity 1.3: Identify and describe security requirements and security mechanism of each enterprise using UMLsec
- Activity 1.4: Describe the set of enterprises contained in the networked enterprise
- Activity 1.5: Describe the common security policy of the networked enterprise
- Activity 1.6: Specify the set common security requirements based on the common security policy using UMLsec

Sub-process 2: Identify relationships between enterprises

- Activity 2.1: Identify and describe relationship between each enterprise in the networked enterprise.
- Activity 2.2: Identify and describe the distribution of responsibility among the enterprises.
- Activity 2.3: Identify stakeholders and specify in which enterprise they belong.

The CORAS framework covers the development of systems in networked enterprises through the activities in context identification. However, they do not state explicit how networked enterprises differs from traditional development settings. In the extended approach the two first sub-process are designed in order to meet the challenges of development in networked settings by describing each enterprise, their contribution to the future system, their security policy and thus their security requirements. Further, the relationships between the enterprises and how this affect the future system

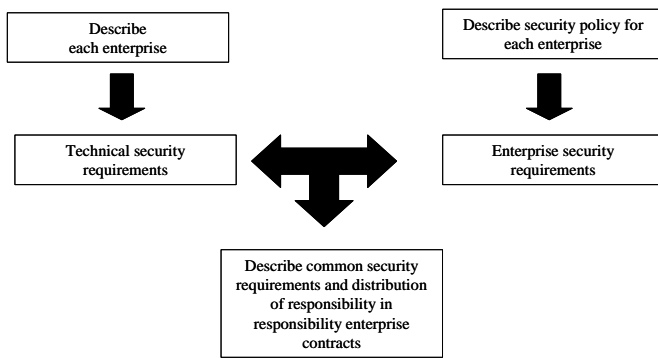


Figure 4. Overview of sub-process 1 and 2

is also specified. After handling each enterprise separately one aggregates the information into common security requirements and a clearly specified distribution of responsibilities among the involved enterprises in responsibility contracts. Figure 4 provides an overview of sub-process 1 and 2 in MBRA for networked enterprises.

5 Secure development of web-based systems

In system development one usually distinguishes between three levels of abstractions: the requirement specification, the design specification and the implementation. The design specification is thus a refinement of the requirement specification, and the implementation is a refinement of the design specification. In the integrated system development and risk management process one has to consider both the intended and the unintended but possible behaviours of the system. In this section we will demonstrate the applicability of the approach using a small e-commerce example. Due to space limitation we will only focus on sub-processes 1, 2 and the activities identify threats to assets (activity 4.1) and identify treatment options (activity 7.1).

The example system is an e-commerce system using the Internet as a backbone in a networked enterprise setting. Involved enterprises service provider and payment service provider. Figure 5 illustrates the payment service and the involved enterprises of the e-commerce system.

Sub process 1 consists of six activities, where activity 1.3 focuses on specifying security requirements and security mechanism using UMLsec. In activity 1.1 we identify and describe each of the enterprises involved and in activity 1.4 we describe the set of enterprises contained in the networked enterprise. There

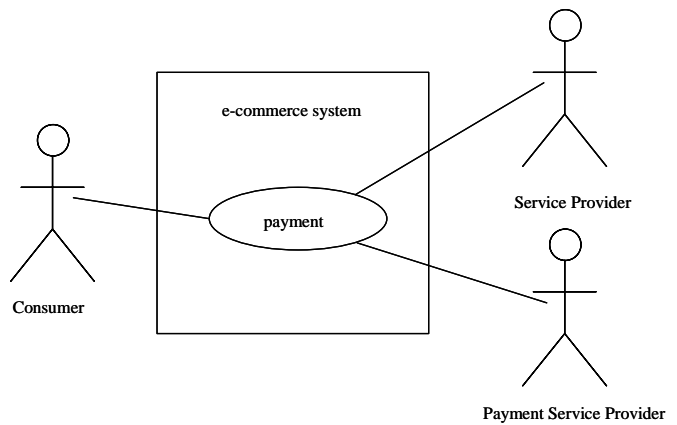


Figure 5. Involved enterprises related to the payment service in the e-commerce system

are two enterprises involved in the e-commerce system (see Figure 5). The service provider is dependent on the payment services delivered by the payment service provider. The e-commerce system procure travel-packets on the Internet. The customer pays for the travel-packets online for which the service provider uses a payment service provider to check, perform and verify the payment.

Activity 1.2 deals with identifying and describing the security policy of each enterprise. Due to space limitations we will only focus on the overall aspects related to the payment service. The service provider has specified that the identity and payment information of a customer should never be revealed to an unauthorised party. The service provider has a contract with the payment service provider stating that the identify of customer and payment information should never be disclosed to an unauthorised party. In the following we will transform the security policies of each of the enterprise to a common set of security requirements in the networked enterprise setting, which denotes activity 1.3.

The security requirements that are relevant here are:

- **Confidentiality:** The following data should remain confidential: the customer id, account number, and expiry date (to prevent misuse of the data for unauthorised transactions), and the amount of money to be paid (for privacy).
- **Integrity:** The integrity of the following data should be protected: amount of money to be paid (to prevent unauthorised increase of the amount to the benefit of the recipient).

- Non-repudiation of a transaction and accountability of the customer and the service provider should be guaranteed to prevent all parties from cheating.

To ensure confidentiality and integrity we make use of the UMLsec stereotype *llcriticalgg* with the tagged values *secrecy* for confidentiality and *integrity* for integrity. To ensure non-repudiation we use the UMLsec stereotype *llfair exchangegg* with the tagged values *start* and *stop*. Figure 6 specify the security requirements for confidentiality and integrity, while Figure 7 specify the non-repudiation requirement. In addition to technical security requirements one needs enterprise procedures and guidelines specifying clearly how the different technical security requirements should be maintained. By specifying procedures and guidelines on enterprise level based on the common security requirements and the responsibility contracts one ensures that the technical security requirements are maintained throughout the whole life cycle of the system.

Sub-process 2 concerns the identification of relationships among the enterprises, along with describing the distribution of responsibilities. Activity 2.1 deals with describing the relationship between the different enterprises, which in this case was done during sub-process 1. The last activity in this sub-process concerns the identification of stakeholders along with specifying in which enterprise they belong. For simplicity reasons this is not done in this paper. In the following we illustrate sub-process 2 by describing the distribution of responsibilities related to each of the security requirements, which denotes activity 2.2:

- The confidentiality and integrity of the customer data and payment information is within the responsibility of the parties involved in communicating this data. The service provider is responsible for the communication issued by the service provider or the consumer, while the payment service provider is responsible for all communication issued from the payment service provider. Secure storing of information is not considered in this example.

In activity 4.1, identification of threats, one uses threat identification input diagrams to identify threats related to the identified assets. Figures 6 provides an example of a threat identification diagram. Output from threat identification is modelled as a threat scenario. For an example of how to model threat scenarios using UML see [HdBLS02]. Identified threats is eavesdrop of account number, expire date and amount of money to be paid and manipulation of amount of money to be paid.

With regards to risk treatment and the evaluation of alternative approaches in activity 7.1 and 7.2, there are several possible solutions to reduce both the frequency and the impacts of the identified threats. Treatment options are considered in each step of the iterative development and assessment process and provide feedback for specification of new or refined security requirements. Risk assessment involves not only risk identification and analysis, but also a decision process regarding acceptable and not acceptable risks. This is captured through the risk acceptance criteria activity in sub-process 3, for which provides rules for risks in need of treatment.

Treatment options identified in this assessment is to encrypt the link between the customer, service provider and the payment service provider. Further, one should also authenticate users of the e-commerce system using either weak authentication (username and password), which are already implemented, or strong authentication. Figures 8 and 9 illustrate possible treatments, which are used as input to system development through the requirement specification, design specification, and implementation phase. More information on this is given in [Jür02, Jür03b].

6 Tool support

In this section we describe some prototypical tool-support for security-critical systems development for checking constraints such as those associated with the stereotypes defined in Section 2, which is currently under development. A first version has been demonstrated at [Jür03a]. The tool works with UML 1.4 models, which can be stored in a XMI 1.2 (XML Metadata Interchange) format by a number of existing UML design tools. To avoid having to process UML models directly on the XMI level, the MDR (MetaData Repository, <http://mdr.netbeans.org>) is used, which allows one to directly operate on the UML concept level (this is, for example, used by the UML CASE tool Poseidon, <http://www.gentleware.com>). The MDR library implements repository for any model described by a modelling language compliant to the MOF (Meta Object Facility). It is hoped that this approach may ease the transition to future UML versions.

Figure 10 illustrates the functionality of the tool. The developer creates a model and stores it in the UML 1.4 / XMI 1.2 file format. The file is imported by the tool into the internal MDR repository. The tool accesses the model through the JMI interfaces generated by the MDR library. The checker parses the model and checks the constraints associated with the stereotype. The results are delivered as a text report for the

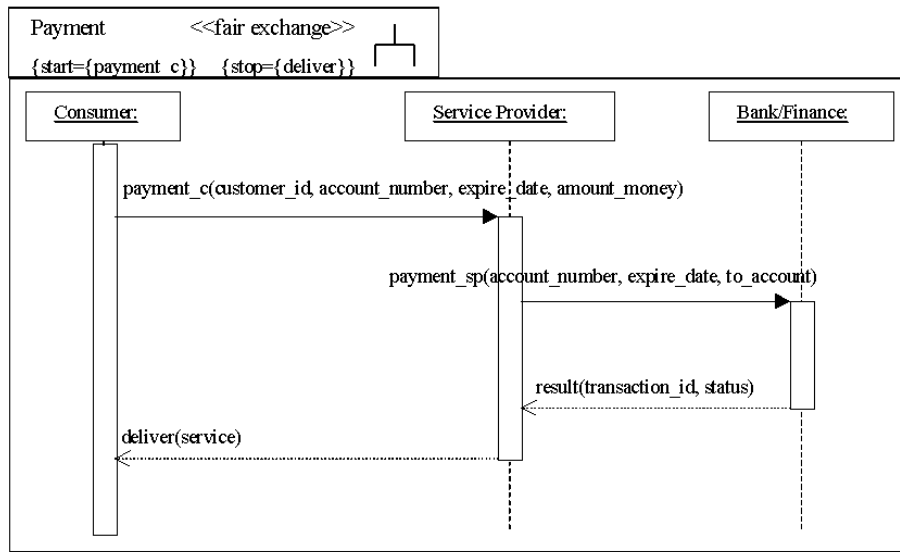


Figure 6. System description (security requirements) and threat identification input diagram using UMLsec

developer describing found problems, and a modified UML model, where the stereotypes whose constraints are violated are highlighted.

7 Related work

There exist a number of specialised risk assessment methodologies for the security domain. Within the domain of healthcare information systems, for example, the British Government’s Central Computer and Telecommunication Agency (CCTA) has developed CRAMM (CCTA Risk Analysis and Management Methodology) [BD92]. CRAMM aims at providing a structured and consistent approach to computer management of all systems. The UK National Health Service considers CRAMM to be the standard for risk analysis within systems supporting healthcare. However, CRAMM is intended for risk analysis of computerised systems in general. However, it is not based on the concept of MBRA and thus do not benefit from using system models as direct input to risk assessment.

Reactive System Design Support (RSDS) [LAC00] and Surety Analysis [WCF99] are methodologies integrating modelling and risk analysis methods. RSDS is an integrated modelling and risk analysis tool-supported methodology developed by King’s College London and B-Core UK, Ltd. Surety Analysis is a method developed in Sandia National Laboratories,

a governmental research organisation in the U.S. and aims at modelling and risk analysis of critical and complex systems. These approaches do not however put particular focus on networked enterprises, thus they do not provide particular support to handle these challenges.

E.B. Fernandez and J.C. Hawkins present in [FH97] an extension of use cases and interaction diagrams to develop distributed system architecture requirements. Among other non-functional requirements they introduce questions for requirements elaboration, like e.g. system communication load, fault tolerance, safety, real-time deadlines and security. However, this work is mainly focused on application examples for use cases in security-critical systems, not on giving a methodology for their development or a concept for their integration with domain models. More generally, there are further approaches to a rigorous development of critical systems based on UML, including [PO01, GFR02] (and other articles in [JCF⁺02]).

8 Conclusion

In this paper, we have presented an integrated system development and risk management process for development in a networked enterprise setting. The proposed approach extends the CORAS framework with two sub-processes and incorporates support for specif-

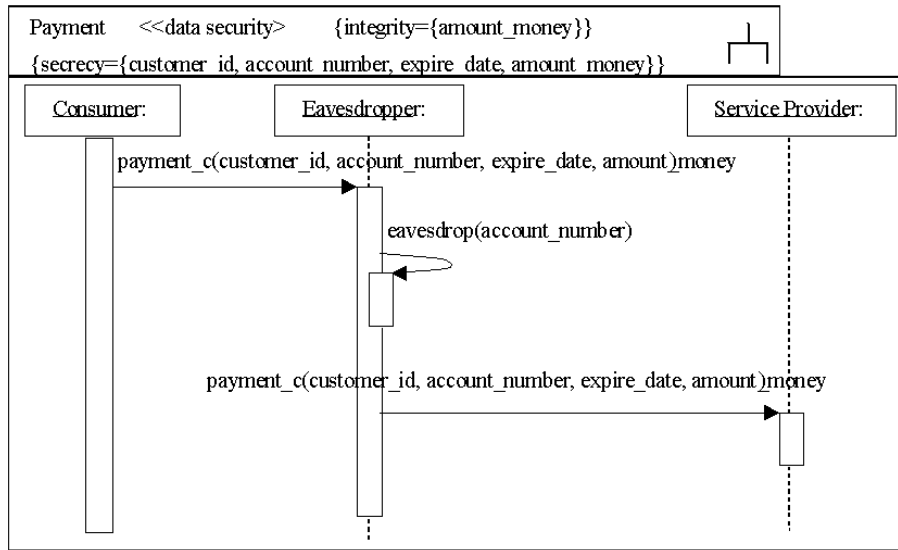


Figure 7. Security requirement for confidentiality and integrity using UMLsec

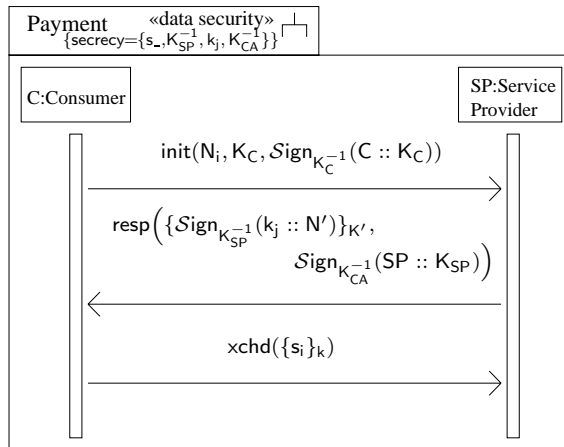


Figure 8. Treatment options encryption of links

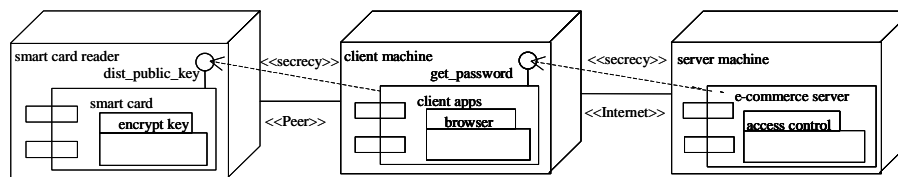


Figure 9. Treatment option strong authentication of users

ing security requirements and mechanical analysis using UMLsec.

The integrated system development and risk management process for networked enterprises consists of seven sub-processes: (1) Identify enterprises, (2) Identify relationship between enterprises, (3) Identify context, (4) Identify risks. (5) Analyse risk, (6) Risk evaluation and (7) Risk treatment. Each of the sub-processes consist of a set of activities, interacting both between activities within one sub-process and between activities in different sub-processes. Furthermore, the process is based on RUP and RM-ODP and is therefore iterative. One would typically specify, analyse, and design parts of the system or the whole system at varying levels of detail in one iteration. After some iterations, the system is specified, analysed and designed, making sure that risks are treated at the most cost-effective time in the process (that is, as early as possible), and that the security level of the system is thoroughly documented and correct according to the setting in which it will operate.

In the proposed approach, models are used for five purposes: (1) Precise specification of security requirements, (2) As a medium to communicate security requirements, (3) To describe the target of assessment, (4) As a medium to communicate risk assessment results and (5) To document risk assessment results.

The focus for the work was to support secure development of web-based systems in a networked enterprise setting. The combination of MBRA, process and precise semantic ensures the correct level of security by describing the relationship between the different enterprises, distributing responsibility among them, performing a risk assessment and by focusing on cost-effective implementation of security mechanisms. These aspects are documented and illustrated using an example that focuses on how MBRA, processes, UMLsec and mechanical analysis interact by performing risk identification, analysis, and evaluation based on a precise description of the system.

Acknowledgments: G. Wimmel, G. Popp and T. Kuhn provided helpful comments on a draft of this paper. The work is further based on the results from the IST-project CORAS and the work done by the 11 partners in this project.

References

- [13300] ISO/IEC TR 13335. Information technology – Guidelines for management of IT Security. <http://www.iso.ch/>, 1996–2000.
- [14700] IEEE 1471. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, 2000.
- [17700] ISO/IEC 17799-1. Information technology – Code of Practice for information security management. <http://www.iso.ch/>, 2000.
- [43699] AS/NZS 4360:1999. Risk management. Standards Australia, Strathfield, 1999.
- [BD92] B. Barber and J. Davey. The use of the ccta risk analysis and management methodology cramm in health information systems. In K.C. Lun, P. Degoulet, T.E. Piemme, and O. Rienhoff, editors, *MEDINFO 92*, pages 1589–1593, Amsterdam, 1992. North Holland Publishing Co.
- [COR02] CORAS, The CORAS Integrated Platform. Poster at the CORAS public workshop during ICT-2002, 2002.
- [DRR⁺02] T. Dimitrakos, B. Ritchie, D. Raptis, J. Oyvind Aagedal, F. den Braber, K. Stølen, and S. Houmb. Integrating model-based security risk management into ebusiness systems development: The coras approach. In J. Monteiro, P. Swatman, and L. Tavares, editors, *Second IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2002)*, volume 233 of *IFIP Conference Proceedings*, pages 159–175. Kluwer, 2002.
- [FH97] E.B. Fernandez and J.C. Hawkins. Determining role rights from use cases. In *Workshop on Role-Based Access Control*, pages 121–125. ACM, 1997.
- [GFR02] G. Georg, R. France, and I. Ray. An aspect-based approach to modeling security concerns. In Jürjens et al. [JCF⁺02].
- [GO84] S.B. Guarro and D. Okrent. The logic flowgraph: A new approach to process failure modeling and diagnosis for disturbance analysis applications. *Nuclear Technology*, page 67, 1984.
- [HdBLS02] S.-H. Houmb, F. den Braber, M. Soldal Lund, and K. Stølen. Towards a UML profile for model-based risk assessment. In Jürjens et al. [JCF⁺02].
- [JCF⁺02] J. Jürjens, V. Cengarle, E. Fernandez, B. Rumpe, and R. Sandner, editors. *Critical Systems Development with UML*, number TUM-I0208 in TUM technical report, 2002. UML'02 satellite workshop proceedings.
- [Jür02] J. Jürjens. UMLsec: Extending UML for secure systems development. In J.-M. Jézéquel, H. Hussmann, and S. Cook, editors, *UML 2002 – The Unified Modeling Language*, volume 2460 of *LNCS*, pages 412–425, Dresden, Sept. 30 – Oct. 4 2002. Springer.
- [Jür03a] J. Jürjens. Developing Security-Critical Systems with UML, 2003. Series of tutorials at international conferences including OMG DOCsec 2002, IFIP SEC 2002, APPLIED INFORMATICS 2003, ETAPS 2003, OMG Workshop On UML for Enterprise Applications 2003, Formal Methods Symposium 2003. Download of material at <http://www4.in.tum.de/~juerjens/csdumltut>.
- [Jür03b] J. Jürjens. *Secure Systems Development with UML*. Springer, 2003. In preparation.
- [KM87] I.S. Kim and M. Modarres. Application of Goal Tree-Success Tree Model as the Knowledge-Base of Operator Advisory System. *Nuclear Engineering & Design J.*, 104:67–81, 1987.
- [Kru99] P. Krutchten. *The Rational Unified Process, An Introduction*. Readings, MA. Addison-Wesley, 1999.
- [LAC00] K. Lano, K. Androutopoulos, and D. Clark. Structuring and Design of Reactive Systems using RSDS and B. In *FASE 2000*, LNCS. Springer-Verlag, 2000.

- [PJWB03] G. Popp, J. Jürjens, G. Wimmel, and R. Breu. Security-critical system development with extended use cases. In *10th Asia-Pacific Software Engineering Conference (APSEC 2003)*, Chiangmai (Thailand), 10-12 December 2003. IEEE Computer Society.
- [PO01] R.F. Paige and J.S. Ostroff. A proposal for a lightweight rigorous UML-based development method for reliable systems. In *Workshop on Practical UML-Based Rigorous Development Methods*, Lecture Notes in Informatics, pages 192–207. German Computer Society (GI), 2001. UML 2001 satellite workshop.
- [Put00] J.R. Putman. *Architecting with RM-ODP*. Prentice-Hall, 2000.
- [RJB99] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.
- [WCF99] G. Wyss, R. Craft, and D. Funkhouser. *The Use of Object-Oriented Analysis Methods in Surety Analysis*. Sandia National Laboratories Report, 1999.

9 Appendix: Terminology

Enterprise A company, firm or association, or other legal entity or part thereof, whether incorporated or not, public or private, that has its own function(s) and administrations [43699].

Networked Enterprise A set of companies, firms or associations, or other legal entities or part thereof, whether incorporated or not, public or private, that has its own function(s) and administration.

Information System A collection of components and stakeholders organised to accomplish a specific function or set of functions.

Target of Evaluation (ToE) The part of an information system that is the subject of a security assessment.

Context The strategic, organisational and risk management context in which the rest of the risk management process will take place [43699].

Security Policy Rules, directives, and practices that govern how assets, including sensitive information, are managed, protected and distributed within an organisation and its IT systems. The definition is derived from IT security policy in ISO/IEC 13335 [13300].

View A representation of a whole system from the perspective of a related set of concerns [14700].

Viewpoint A specification of the conventions for constructing and using a view [14700].

System Stakeholder An individual, team, or organisation (or classes thereof) with interest in, or concerns relative to, a system [14700].

Asset Something to which an organisation directly assigns value and, hence, for which the organisation requires protection [43699].

Asset Value The value of assets in terms of their importance to the business. These values are usually expressed in terms of the potential business impacts or unwanted incidents. This could, in turn, lead to financial losses, loss of revenue, market share, or company image [43699].

Threat A potential cause of an unwanted event, which may result in harm to a system or organisation and its assets [13300].

Vulnerability A weakness of an asset or group of assets which can be exploited by one or more threats [23]. Unwanted Incident - Incident such as loss of confidentiality, integrity and/or availability [43699].