



HOVEDOPPGAVE

Kandidatens navn: Kine Kvernstad Hansen

Fag: Datateknikk

Oppgavens tittel (norsk):

Oppgavens tittel (engelsk): Towards a UML profile for model-based risk assessment of security critical systems

Oppgavens tekst:

The EU IST-project CORAS has developed a framework for model-based risk assessment of security critical systems based on the Australian/New Zealand Standard for risk management, AS/NZS 4360, the security standards ISO/IEC 17799 and ISO/IEC 13355 and the safety standard IEC 61508. As a part of the risk documentation framework CORAS has developed a UML (Unified Modeling Language) profile for documenting risk assessment results. The UML-profile covers part of the activities in the risk management process, but does not provide sufficiently support for detailed specifications of output from risk identification and risk analysis, sub-process 2 and 3 in the CORAS risk management process.

The main aim of this thesis is to specify a UML-profile providing support for detailed specification of output from risk identification and risk analysis (sub process 2 and 3 in the CORAS risk management process) of security critical systems. The profile should offer semantic for UML sequence and activity diagrams understandable for non-technical stakeholders without prior knowledge of UML.

The work will include a study of AS/NZS 4360. In particular, the thesis should refine the risk management process presented by the standard in order to provide particularly support for the security domain. Based on the refined process, a security documentation framework specifying required output from the five sequential sub-processes within AS/NZS 4360 should be specified. This framework will form the foundation for eliciting requirements for the UML profile.

Oppgaven gitt:	27. januar 2003
Besvarelsen leveres innen:	30. juni 2003
Besvarelsen levert:	27. juni 2003
Utført ved:	Institutt for datateknikk og informasjonsvitenskap
Veileder:	Siv Hilde Houmb

Trondheim, 27. juni 2003

Tor Stålhane
Faglærer

Abstract

Security assessment is a multidisciplinary task involving both technical and non-technical stakeholders. One major challenge when performing such assessments is to establish a common understanding of threats, vulnerabilities and security risks among the different groups of stakeholders participating in the assessment. In order to enhance communication, we need notation understandable for both technicians and non-technicians. Our work is based on the assumption that the use of visual models may enhance communication and interaction by offering a common understanding of concepts used. Furthermore, use of models may support reuse of assessment results by offering a uniform and repeatable way of representing information.

This thesis specifies SecurityAssessmentUML, a UML profile for model-based security assessments, as well as a security assessment process and its associated documentation framework. The main objective of SecurityAssessmentUML is to support documentation of all required output from risk identification and risk analysis of security critical systems. In particular the profile supports specification of concrete scenarios demonstrating how attacks may occur, as well as fault tree inspired activity diagrams for analysing the frequency of risks.

Requirements for the profile was elicited based on common security ontology and information flow between activities in the security assessment process. The security assessment process was specified based on the Australian/New Zealand standard for risk management, AS/NZS 4360. Furthermore, the work is inspired by the CORAS project.

SecurityAssessmentUML has not been subject to extensive testing and is therefore in a draft version. However, preliminary evaluation of the profile has identified some areas for improvement. In particular the profile should be extended to support specification of how unwanted incidents affect the value of system assets. Furthermore, the profile should be evaluated by target users in order to check the suitability of the diagrams as a communication medium.

Preface

This report is the result of my master thesis in Software Engineering at the Norwegian University of Science and Technology (NTNU). The work was carried out at the Department of Computer and Information Science (IDI) during spring 2003, under the supervision of PhD. student Siv Hilde Houmb.

I would like to thank my supervisor, Siv Hilde Houmb, for insightful input and valuable feedback throughout the project. I would especially thank her for her inspirational tips and patience, and for helping me back on the track whenever needed. I would also like to thank my subject teacher, Professor Tor Stålhane, for useful guidance. Furthermore, I would like to thank the participants in the judgement study, Kai Hansen for valuable input on final diagrams, Karine Sørby and Ørjan Lillevik for useful discussions on the profile, and Roar Ekren for valuable comments on content and language.

Trondheim, June 27, 2003

Kine Kvernstad Hansen

Contents

- 1 Introduction** **1**
 - 1.1 Motivation and background 1
 - 1.2 Problem description 2
 - 1.3 Related work 2
 - 1.4 Structure of the report 2

- 2 Security critical systems** **4**
 - 2.1 Security and Context 4
 - 2.2 Security Attributes 5
 - 2.3 Security Attacks 5

- 3 The Unified Modeling Language** **7**
 - 3.1 UML diagrams 7
 - 3.2 UML extension mechanisms 8
 - 3.3 UML for the security domain 9

- 4 Risk Management and Assessment** **11**
 - 4.1 Terminology 11
 - 4.2 Methods for risk identification and analysis 12
 - 4.2.1 FTA 12
 - 4.2.2 Markov analysis 12
 - 4.3 Australian Standard for Risk Management 14
 - 4.3.1 Establish the context 15
 - 4.3.2 Risk identification 15
 - 4.3.3 Risk analysis 15
 - 4.3.4 Risk evaluation 16
 - 4.3.5 Risk treatment 16
 - 4.3.6 Monitoring and Review 16
 - 4.3.7 Communication and Consultation 16
 - 4.3.8 Discussion 17

- 5 The CORAS Approach** **19**
 - 5.1 Introduction to the CORAS approach 19
 - 5.2 CORAS risk management process 20
 - 5.3 CORAS risk documentation framework 21
 - 5.4 The CORAS UML-Profile for Model-based Risk Assessment 21
 - 5.5 Threat diagrams 22
 - 5.6 State Analysis diagrams 23
 - 5.7 Discussion 24

6	Security Assessment	25
6.1	Terminology	25
6.2	Ontology	27
6.3	Process for security assessment and security management	30
6.3.1	Security management process	30
6.3.2	Information flow between activities	32
6.4	Security documentation framework	33
6.5	Requirements for the UML profile for security assessment	36
6.5.1	Sub-process 2: Identify risk	37
6.5.2	Sub-process 3: Risk analysis	39
7	UML Profile for Security Assessment	44
7.1	Risk Identification	44
7.2	Risk Analysis	46
7.2.1	Determine existing controls	46
7.2.2	Determine consequence and likelihood of unwanted incidents	47
7.3	Discussion	50
8	Evaluation process	52
8.1	Evaluation criteria for the UML profile	52
8.2	Evaluation strategy	53
8.3	Accomplishment of expert judgements	55
9	Evaluation	56
9.1	Mapping schema between ontology and modelling constructs	56
9.2	Trial	56
9.2.1	Description of analysis object	58
9.2.2	Risk identification	58
9.2.3	Risk analysis	60
9.3	Evaluation criteria	64
10	Discussion	67
10.1	Completeness of profile	67
10.2	Comparison to existing methods	67
10.3	Trade-off between intuitive description and standard syntax	68
10.4	Validity of evaluation	68
10.5	Suitability of UML for non-technicians	69
10.6	Adaption to the safety domain	69
11	Conlusion and further work	70
11.1	Conclusion	70
11.2	Further work	70
A	Glossary	72
B	Research Agenda	76
C	Evaluation schema used in judgement study	77
D	Results from the judgement study	83

E FMEA used for evaluation of SecurityAssessmentUML	85
Bibliography	86

List of Figures

1.1	Report structure	3
2.1	Relationship between IT-system, information system, organisation, society, ToE and context	5
2.2	General categories of security threats [52]	6
3.1	Alternative stereotype representations [51].	9
3.2	Tagged value. Modified from [21]	9
3.3	Constraint [9]	9
4.1	Fault tree symbols [35].	13
4.2	State transition diagram	13
4.3	AS/NZS 4360: Risk mangement process [4]	14
4.4	Information flow between activities in AS/NZS	17
5.1	The CORAS framework [54]	19
5.2	Model-based risk assessment [21].	20
5.3	The CORAS risk documentation framework [21].	22
5.4	Threat diagram. Generalized from [21].	23
5.5	State Analysis diagram [51]	23
6.1	The relationship between assets and their environment	28
6.2	Undesirable system behaviour	28
6.3	Ontology for risk concepts	29
6.4	Ontology for security assessment	30
6.5	Relationship between threats, vulnerabilities, safeguards and unwanted incidents	31
6.6	Security management process	32
6.7	Informationflow in the security management process	33
6.8	Security documentation framework	34
6.9	Ontology for risk identification and risk analysis	36
6.10	Example system	37
6.11	Sequence diagram showing events using UML1.4	38
6.12	Activity diagram showing events using UML1.4	39
6.13	Activity diagram showing events using UML1.4	40
6.14	FTA notation for example scenario	41
6.15	Fault tree of example scenario represented as a UML activity diagram	42
7.1	Mapping of concepts to UML sequence diagrams for risk identification	45
7.2	Example attack documented using extensions to sequence diagrams	45
7.3	Categories of threats illustrated using extensions to sequence diagrams	46

7.4	Mapping of concepts to UML activity diagrams for risk identification	46
7.5	Example attack documented using extension to activity diagrams	47
7.6	Mapping of the concept safeguard to UML activity diagram semantic	47
7.7	Example attack documented using extension to activity diagrams	48
7.8	Mapping of concepts to UML activity diagrams for use of fault tree notation . .	48
7.9	Example fault tree specified using extensions to activity diagrams	49
7.10	Sequence diagram tagged with output from risk analysis	50
7.11	Activity diagram tagged with output from risk analysis	51
8.1	Research strategies [38]	54
9.1	Mapping between concepts in ontology and UML extensions defined by SecurityAssessmentUML	57
9.2	Normal system behaviour	58
9.3	Manipulation-attack specified using extensions to activity diagrams	59
9.4	Manipulation-attack specified using extensions to sequence diagrams	60
9.5	SecurityAssessmentUML notation for fault tree “Fraud against consumer”. Adapted from [19]	61
9.6	SecurityAssessmentUML notation for generic fault tree 1,6. Adapted from [19] .	62
9.7	SecurityAssessmentUML notation for generic fault tree 1,0,1. Adapted from [19]	63

List of Tables

4.1	Transition probability matrix	13
5.1	The activities of the CORAS risk management process [21]	21
6.1	Description of concerns	35
6.2	Required modelling support in UML sequence diagrams for documenting output from risk identification	38
6.3	Required modeling support in UML activity diagrams for documenting output from risk identification.	39
6.4	Required modelling support in UML activity diagrams for documenting safeguards	40
6.5	Required modelling support for UML activity diagrams for specifying fault trees	42
6.6	Required modelling support in UML activity diagrams for specifying consequence value	42
6.7	Required modelling support in UML activity diagrams for specifying estimated risk level	43
7.1	Mapping of concepts to UML activity diagrams for risk analysis	49
D.1	Results of judgement study on extension to sequence diagram for documenting results from risk identification	83
D.2	Results of judgement study on extension to activity diagram for documenting results from risk identification	84
D.3	Results of judgement study on extension to activity diagram for documenting results from risk analysis	84
D.4	Results on general questions about the profile	84
E.1	FMEA used for evaluation	85

Chapter 1

Introduction

1.1 Motivation and background

Modern society heavily relies on networked information systems. The risks associated with these systems may be fatal, threatening the economical and physical well-being of people and organisations. Unavailability of a telemedicine platform may, for instance, result in loss of life, while an organisation conducting business electronically can be put out of business as a result of a successful denial-of-service attack.

Security is often compromised by exploiting weaknesses in the way security mechanisms are implemented, and not by breaking the mechanisms themselves [32]. A common problem is that protection decisions are incomplete or ineffective because they are based on prior experience with vulnerabilities and current threats [3]. As a means for reducing this problem, risk assessments can be conducted. The main objective of such assessments is to support the decision making processes by offering greater insight into risks and the impact of these risks [4].

Risk assessments have been widely used in the safety domain, but currently there is no largely used standard targeting security in particular. However, the EU IST-project CORAS [13] has developed a framework for model-based risk assessment of security critical systems based on the Australian/New Zealand Standard for risk management, AS/NZS 4360 [4], the security standards ISO/IEC 17799 and ISO/IEC 13355 and the safety standard IEC 61508. As a part of the framework CORAS has developed a UML (Unified Modeling Language) profile for documenting risk assessment results. The UML-profile covers part of the activities in the risk management process, but does not provide sufficient support for detailed specifications of output from risk identification and risk analysis, sub-process 2 and 3 in the CORAS risk management process.

The purpose of model-based risk assessment is to solve two main challenges related to risk assessment; efficient communication and reuse of assessment documentation. Risk assessment is a multidisciplinary task involving experts in all relevant areas, including both technical and non-technical people. In order to enhance communication and interaction between the participants an unambiguous understanding of concepts used is necessary. Work addressing model-based risk assessment is based on the assumption that use of visual models is an appropriate means for achieving this. Use of models is also believed to increase reuse of assessment documentation as they may contribute to the repeatability and objectivity of the way results are described.

1.2 Problem description

The main aim of this thesis is to specify a UML-profile providing support for detailed specification of output from risk identification and risk analysis (sub process 2 and 3 in the CORAS risk management process) of security critical systems. The profile should offer semantic for UML sequence and activity diagrams understandable for non-technical stakeholders without prior knowledge of UML.

The work will include a study of AS/NZS 4360. In particular, the thesis should refine the risk management process presented by AS/NZS 4360 in order to provide particularly support for the security domain. Based on the refined process, a security documentation framework specifying required output from the five sequential sub-processes within AS/NZS 4360 should be specified. This framework will form the foundation for eliciting requirements for the UML profile.

Description of the work process is found in Appendix B.

1.3 Related work

In practice, the traditional strategy for security assurance has been “penetration and patch” [31]: if a penetration of a system is noticed and the exploited weakness can be identified, the vulnerability is removed. The strategy may be supported by use of “tiger teams” [18], [16]. Furthermore, focus within security has traditionally been on technical solutions rather than on providing formal arguments for the need of technical solutions.

Recently, work has been performed in order to develop a structured and systematic approach for identifying and managing risks to security critical systems. The OCTAVE framework [2], developed by the NSS Program at SEI, provides guidelines enabling organisations to develop appropriate protection strategies based on risks to critical information assets. OCTAVE does not provide support for model-based risk assessment, however. Model-based risk assessment is supported by the methods CRAMM [7], ATAM [12], RSDS [34] and CORAS [13]. The particular angle of the CORAS approach is its emphasis on security and risk assessment tightly integrated with UML and RM-ODP.

The idea of extending UML with security features is not new. Research has been done to encapsulate information about security requirements into UML. UMLsec [30, 32] is an extension that can be used to specify standard requirements for security critical systems. The profile uses a formalization of the security requirement secrecy. Another example is secureUML [36], which defines a vocabulary to express different aspects of access control, like roles, role permissions and user-role assignments. By applying these profiles security aspects can be integrated into the traditional system development.

1.4 Structure of the report

Figure 1.1 shows the structure of the report. As indicated in the figure, Chapter 2, 3, 4, and 5 present the theoretical background necessary for understanding the rest of the report.

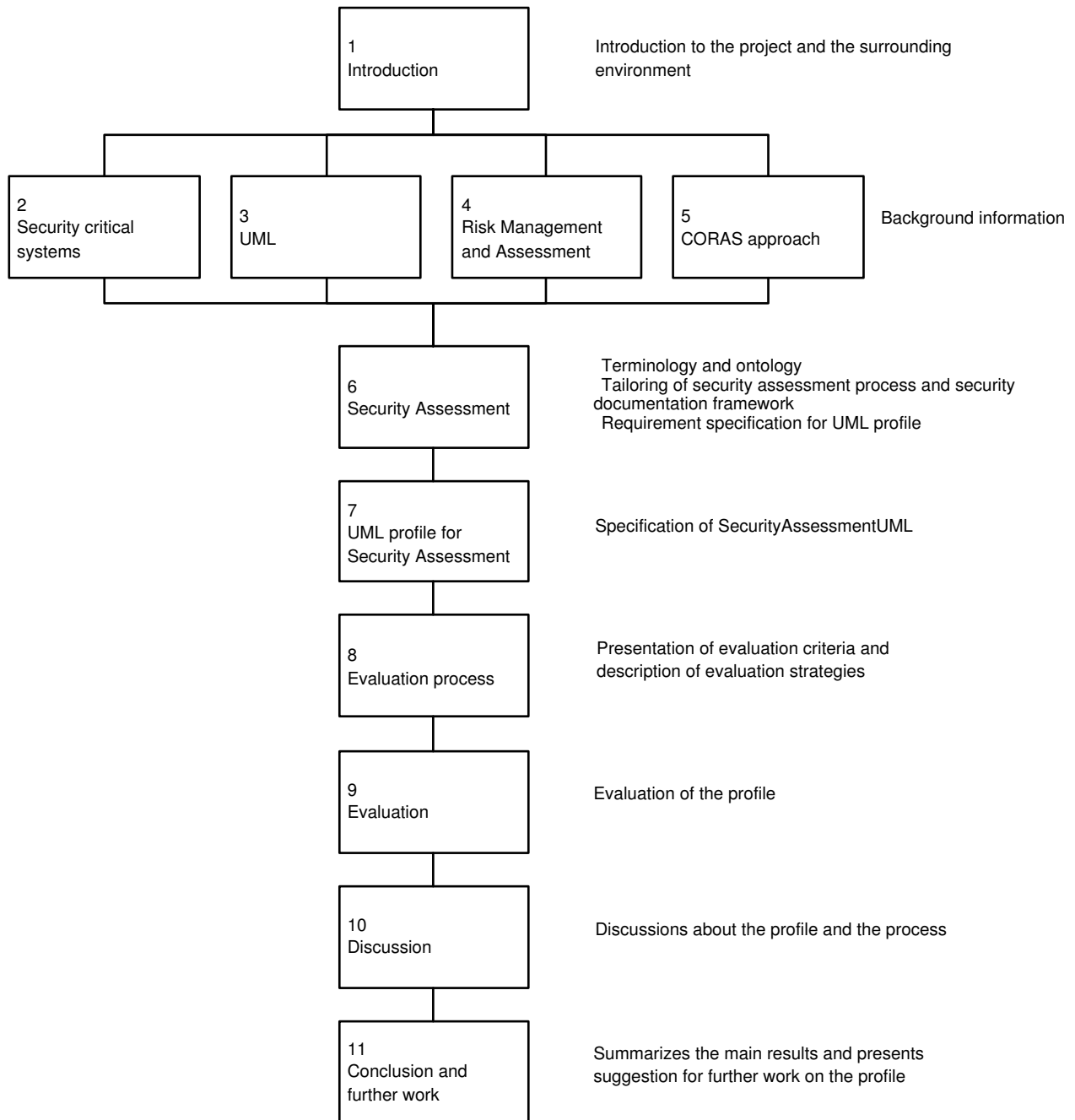


Figure 1.1: Report structure

Chapter 2

Security critical systems

Security is about preserving a set of security attributes in order to protect the assets of the system from potential threats [14]. IT security relates to an IT system, while information security relates to an information system. Normally information security is characterized as the preservation of confidentiality, integrity and availability [27]. However, up to four additional attributes can be added to the list used to characterize a secure system. For instance, ISO13335 [26] defines IT security as defining, achieving, and maintaining the seven security attributes confidentiality, integrity, availability, non-repudiation, accountability, authenticity and reliability [26]. Examples of security critical systems include systems for e-commerce, as well as medical and legal databases. The information held by these systems constitutes a critical resource for the organisation to succeed in its mission, and individuals expect that personal information provided to the system remain private.

This chapter provides a short introduction to security critical systems. First, the relationship between IT systems and information systems are clarified, including a notation on their relationships to their surroundings. Section 2.2 provides definitions of security attributes, and Section 2.3 provides a brief introduction to possible attacks to security critical systems.

2.1 Security and Context

In this report the term security critical system is used to denote the IT system itself and the humans interacting with the technology, as well as all relevant aspects of the organisation and the society surrounding the system. The relationship between these aspects is illustrated in Figure 2.1. *IT systems*, which constitute the computerized part of an information system, consist of computers connected to a network. There are three major assets of such systems, data, hardware and software [42], forming potential targets for attacks affecting the system itself, its data or its stakeholders. *Information systems* consider stakeholders and users in addition to the technological parts of the system. IT and information systems are influenced by the policies of the surrounding *organization* and by laws and regulations of the *society* controlling the organization. An organisation will typically contain several information systems, and an information system may consist of several IT systems.

The target of evaluation (ToE) in a security assessment is defined as an IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation [14]. Most often the ToE covers a subset of the IT components and stakeholders present in one

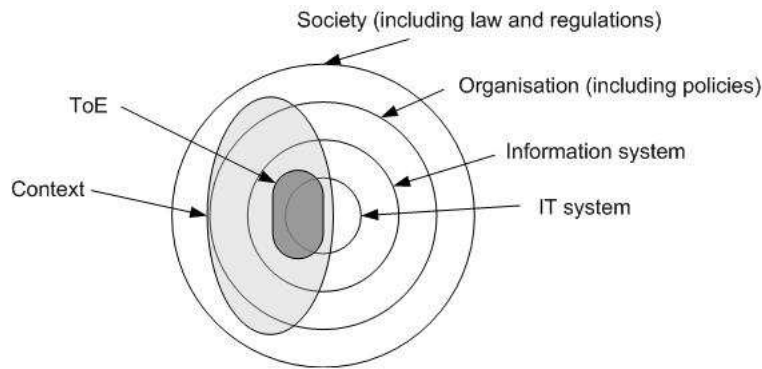


Figure 2.1: Relationship between IT-system, information system, organisation, society, ToE and context [22]

or more information systems. Thus, when assessing security, the IT system must be viewed in a wider context, including organisational policies, stakeholders interests and environmental laws and regulations. This will be further elaborated in Chapter 6, Security Assessment.

2.2 Security Attributes

As mentioned above, a secure IT system provides seven attributes: *confidentiality*, *integrity*, *availability*, *accountability*, *non-repudiation*, *authenticity* and *reliability* [26]. Although only confidentiality, integrity and availability are emphasized in this report, all attributes are defined in this section. *Confidentiality* (or *secrecy*) means that information is made available or disclosed only to authorized individuals, entities, or processes. [26]. *Integrity* means that information is not destroyed or altered in an unauthorized manner and that the system performs its intended function in an unimpaired manner, free from deliberate or accidental unauthorized manipulation of the system [26]. *Availability* means that system services are accessible and usable upon demand by an authorised entity [26]. The ability to prove that an action or event has taken place, in order to prevent later repudiation of this event or action is called *non-repudiation* [26]. *Accountability* means that actions of an entity may be traced uniquely to the entity [26]. *Authenticity* ensures that the identity of a subject or resource is the one claimed [26]. *Reliability* is the property that intended behaviour and results are consistent [26]. IT security includes all aspects related to defining, achieving and maintaining these properties of IT systems [26].

2.3 Security Attacks

The term security attack is used to denote an action that violates one or more of the security attributes of an information system by exploiting a vulnerability of the system. An attack consists of a misuser and a misuse and can be posed either intentionally or unintentionally. Furthermore, the attack may be initiated by either external or internal users of the system being either authorised or unauthorised. In fact, research indicates that about 80 per cent of all security breaches are caused by a company's own staff [17].

Security attacks can be categorised as either active or passive [29]. In an active attack the intruder engages in tampering with the information being exchanged, either through modification of a data stream or through creation of a false data stream. Active attacks relate to attacks on integrity and availability to assets. In a passive attack the intruder merely observes the information passing through the channel without interfering with its flow or content. Passive attacks relate to attacks on confidentiality.

Furthermore, security attacks can be categorised according to the way the attacker intervenes with the information provided [29]. Figure 2.2 provides an illustration of four general categories of threats to the security of a network: interception, fabrication, modification and interruption. The threats are viewed in light of normal operation of a system, where information is successfully passed between the source and the destination without any form of intervention from unauthorised parties (Fig. 2.2a).

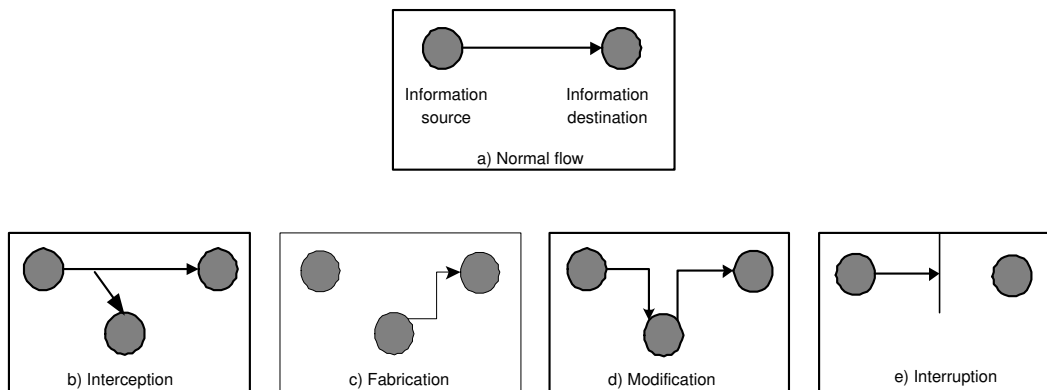


Figure 2.2: General categories of security threats [52]

Interception [29] (Figure 2.2b) is a passive attack where an unauthorised party gains access to an asset by intercepting the information provided. Interception is normally divided into two subtypes, *release of message content* [52] and *traffic analysis* [52]. Traffic analysis is a variant of release of message content, where the intruder must analyse the message using frequency and length information in order to reveal the content of the message.

The three remaining figures represent different forms of active attacks. *Fabrication* [29] (Fig. 2.2c) is an attack where counterfeit objects are inserted into the system by an unauthorised party. Two common variants of fabrication are *replay* [52] and *masquerade* [52]. Replay is an attack where a data unit is retransmitted to produce an unauthorized effect. Masquerade is an attack where one entity pretends to be a different entity.

Modification [29] (Figure 2.2d) involves alteration, delaying or reordering of a message. Examples include changing values of items, altering a program so that it performs incorrectly, and modifying an incoming message. This attack relates to attacks on integrity.

Interruption [52] (Fig. 2.2e) is an attack, which prevents or inhibits the normal use or management of communication facilities. Examples include *denial of service*, *viruses* [43] and *worms* [43]. Interruption relates to attacks on availability of the system.

Chapter 3

The Unified Modeling Language

The Unified Modeling Language (UML) is a general-purpose graphical modelling language for specifying, visualizing, constructing and documenting the artefacts of software-intensive systems [9]. Standardized by the Object Management Group (OMG) in November 1997, it is being supported by numerous tool vendors and is widely used for modelling complex systems. The language is the result of adaption of three OO-modelling languages; Booch's Object-Oriented Analysis and Design [8], Rumbaugh's OMT [48] and Jacobsen's OOSE [28]. The UML provides a set of graphical and textual modelling techniques that aim to provide a common understandable language for developers and customers furthering communication and interaction. Moreover, extension mechanisms, which allow tailoring of the UML for specific application domains, are included. This chapter provides a short introduction to UML diagrams, as well as a description of the extension mechanisms provided. For more information on UML the reader is referred to one of the many textbooks on the subject, for instance [9] or [49].

3.1 UML diagrams

A diagram in UML is a graphical presentation of a set of elements, most often rendered as a connected graph of arcs (relationships) and vertices (other model elements) [49]. Diagrams are used to visualize a system from different perspectives. Because no complex system can be understood in its entirety from only one perspective, the UML defines nine types of diagrams allowing the developer to focus on different aspects of the system independently:

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Statechart diagram
- Activity diagram
- Component diagram

- Deployment diagram

Four of the diagrams are used to represent static system aspects. These are *class diagrams*, *object diagrams*, *component diagrams* and *deployment diagrams*. The remaining five diagrams, *use case*, *sequence*, *collaboration*, *statechart* and *activity*, are used to represent dynamic behaviour of the system under consideration. *Class diagrams* picture the static structure of systems in terms of classes and relationships between them. For each class attributes and methods are defined. In *object diagrams* the classes are replaced with objects. *Component diagrams* are used to illustrate the relationship among software components in the system. These diagrams can be combined with *deployment diagrams*, which represent the pieces of hardware and the relations between them. *Use case diagrams* represent the functions of the system from the users' point of view. Each use case, drawn as ovals, represents a function. The use cases are linked to one or more actors, represented as stick figures. An actor is something external to the system interacting with the system. *Sequence diagrams* provide interactions among objects arranged in a time sequence. In particular, they show the objects participating in the interaction and the sequence of messages exchanged. Interactions among objects are also depicted in *collaboration diagrams*. However, in collaboration diagrams interactions are not ordered according to time. *Statechart diagrams* show state machines, emphasizing on the flow of control from state to state. These diagrams consist of states and transitions between states. *Activity diagrams* are special statechart diagrams in which all or most of the states are activity states and all or most of the transitions are triggered by completion of activities in the source state.

3.2 UML extension mechanisms

The UML defines three extension mechanisms allowing modellers to tailor UML to specific application domains without having to modify the underlying modelling language. The three mechanisms are stereotypes, tagged values and constraints. A UML extension collects stereotypes, tagged values and constraints into a profile. The resulting UML profile adapts the language to the need of the specific domain, while still sharing the concepts generic and common to all domains.

A *stereotype* is an extension to the vocabulary of the UML, allowing the modeller to add new building blocks derived from existing ones. Stereotypes are normally shown as text strings surrounded by brackets placed in or near the symbol for the base model element. However, in order to increase the readability of the diagrams, a representation depicting stereotypes as intuitively understandable icons can be used. Figure 3.1 presents alternative representations for the stereotype Asset, derived from the meta-class “Class”.

A *tagged value* is used to store information about an individual model element. Tags can be defined for existing elements of UML or for individual stereotypes. Its value applies to the element itself and not its instances, and does thus not correspond to a class attribute. A tagged value is represented as a string enclosed by braces and placed below the name of the element it belongs to. The string includes a name of some property the modeller wants to record (the tag), a separator (an equal sign), and the value of that property for the given element. An example is given in Figure 3.2, which specifies the consequence value for an unwanted incident.

A *constraint* is an extension of the semantics of a UML element, providing the possibility to add new rules or modifying existing ones. Constraints specify conditions that must be true for the model to be well-formed. Constraints are shown as strings enclosed in braces and placed near

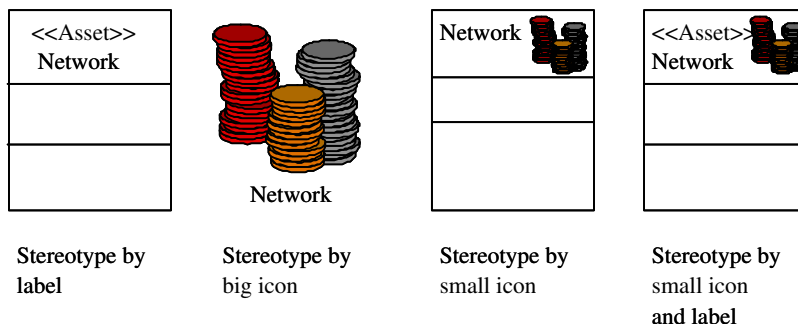


Figure 3.1: Alternative stereotype representations [51].



Figure 3.2: Tagged value. Modified from [21]

the associated element or attached to a dependency relationship. An example is given in Figure 3.3.

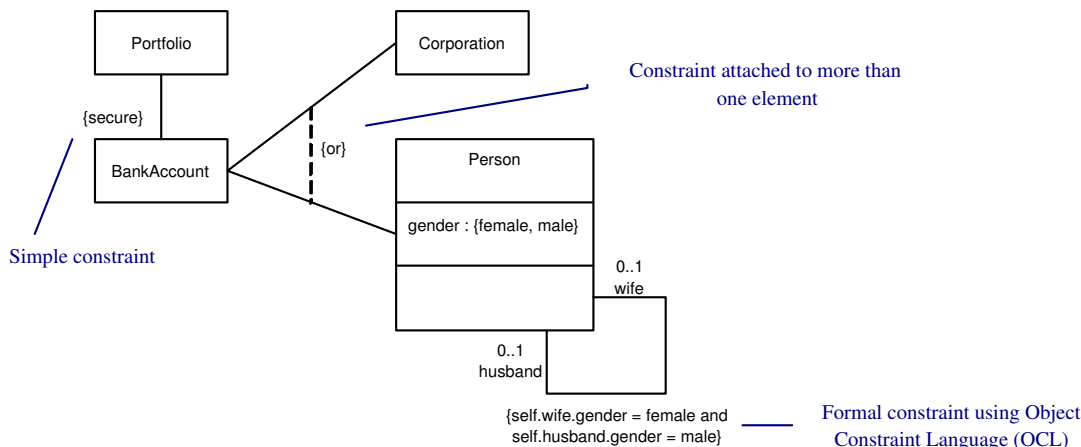


Figure 3.3: Constraint [9]

3.3 UML for the security domain

In general, the UML is a tool for modelling desirable properties of a system. When developing security critical systems, however, the design process needs to take both desirable and undesirable behaviour into account. Thus, in addition to focus on desirable system behaviour, evil actors, as well as vulnerabilities and possible threats to the system, must be identified. Currently, UML does not support modelling of attackers, threats, vulnerabilities and other aspects

related to undesired system behaviour. Extensions can be made, however, using the mechanisms presented in the previous section, to tailor the language to development of systems with a special emphasis on security.

Some research has been done on incorporating security aspects such as the ones mentioned above into UML1.4. CORAS has developed a UML profile for model-based risk assessment aiming at documenting the results of traditional risk assessment techniques in the same way as for the normal system requirements. The profile, which encapsulate the concept of unwanted behaviour and attacks into UML1.4, is presented in chapter 5.4. Jurjens [30, 32] has developed an extension that can be used to specify standard requirements for security critical systems. The profile, called UMLsec, use a formalization of the security requirement secrecy. Another example is SecureUML [36], which defines a vocabulary to express different aspects of access control, like roles, role permissions and user-role assignments. By using one of these profiles or a combination of them, security aspects can be integrated into system development.

Chapter 4

Risk Management and Assessment

This chapter provides a short introduction to the process of managing risk. The focus of the presentation is on the Australian/New Zealand Standard for risk management [4], presented in section 4.3. The standard is used as a basis for the risk management process defined by CORAS (presented in Chapter 5.2) and for the security documentation framework presented in Chapter 6. Terminology connected to risk management is included in Section 4.1, and Section 4.2 provides a brief introduction to widely used methods for performing risk identification and risk analysis.

4.1 Terminology

All definitions in this section is taken from the Australian/New Zealand Standard for Risk Management, AS/NZS 4360:1999 [4].

Risk - The chance of something happening that will have an impact upon objectives. It is measured in terms of consequences and likelihood [4]

Risk analysis - A systematic use of available information to determine how often specified events may occur and the magnitude of their consequences [4]

Risk assessment - The overall process of risk analysis and risk evaluation [4].

Risk evaluation - The process used to determine risk management priorities by comparing the level of risk against predetermined standards, target risk levels or other criteria [4].

Risk identification - The process of determining what can happen, why and how [4].

Risk management - The culture, processes and structures that are directed towards the effective management of potential opportunities and adverse effects [4].

Risk management process - The systematic application of management policies, procedures and practices to the tasks of establishing the context, identifying, analysing, evaluating, treating, monitoring and communicating risk [4].

Risk treatment - Selection and implementation of appropriate options for dealing with risk [4].

4.2 Methods for risk identification and analysis

Methods for identification and analysis of risk have been applied in the safety domain for nearly half a decade. Some of these methods have been adopted for use in the security domain. The rationale for this is that both domains deal with risks and protection against loss; safety deals with threats to life and property, security with threats to privacy and secrecy [35]. Among the most widely used methods are Hazard and operability Analysis (HAZOP), Failure Modes and Criticality Analysis (FMEA) and Fault Tree Analysis (FTA). Analysis may also be performed using Markov modelling. FTA and Markov are briefly described below. For information about Hazop, the reader is referred to [39, 46, 56]. For information about FMEA, the reader is referred to [10].

4.2.1 FTA

Fault Tree Analysis (FTA) is primarily a means for analysing causes of threats, not identifying threats [35]. An undesired system state is specified and the method works backwards to determine its possible causes. The result of an FTA analysis is a set of graphical displays using Boolean logic to depict the logical interrelationships of individual faults that can constitute an unwanted incident. Each level in the tree lists the more basic events necessary and sufficient to cause the event in the level above. Figure 4.1 presents the symbols used when constructing fault trees. For more complex systems, use of additional gates may be necessary [35].

FTA was originally developed in 1962 by Bell Telephone Laboratories. For more information on FTA, the reader is referred to [35, 24]. For information about how to perform qualitative and quantitative analysis on fault trees, the reader is referred to [45].

4.2.2 Markov analysis

Markov analysis provides a means of analysing the reliability and availability of systems. A Markov model consists of a state transition diagram and a probability matrix. The probability matrix is used to document the transition probabilities between the states in the diagram. Consider the simple three-state system presented in Figure 4.2. The diagram corresponds to the probability matrix provided in Table 4.1, where P_{xy} is the probability (i.e. the frequency) of transition between state x and state y .

Markov models can be used to compute MTTF (Mean Time To Failure), MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair). One major drawback of Markov is that Markov diagrams are difficult to construct for large systems [6]. However, the problem can be avoided by independent analysis of smaller parts of the system. For more information about Markov models, the reader is referred to [6] or [47].








	Symbol	Meaning
Logical gates	AND gate 	Output occurs only if all inputs occur.
	OR gate 	Output occurs if at least one of the inputs occur.
States		An event that results from a combination of events through a logic gate.
	Basic event 	A basic fault event that requires no further development. These are leaf nodes in the tree.
	Undeveloped event 	A fault event that is not fully traced to its source.
Transfer symbols	Transfer down 	Transfer symbol used for further development in a cause-chain.
	Transfer up 	Used when the same branch is involved in several paths and when the fault tree spans more than one page.

Figure 4.1: Fault tree symbols [35].

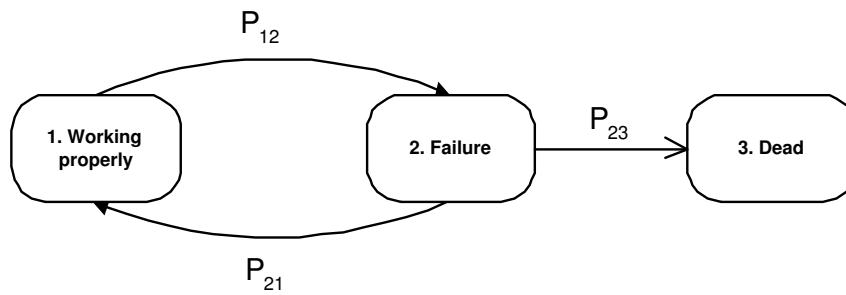


Figure 4.2: State transition diagram

Table 4.1: Transition probability matrix for system depicted in Figure 4.2.

From \ To	1	2	3
1	0	P_{12}	0
2	P_{21}	0	P_{23}
3	0	0	0

4.3 Australian Standard for Risk Management

The Australian/New Zealand Standard for Risk Management, AS/NZS 4360 [4], provides a generic framework for the process of managing risk. The standard divides the elements of the risk management process into seven sub-processes, of which five are sequential and two are in parallel with the other five. An overview of the process is given in Figure 4.3. During the first

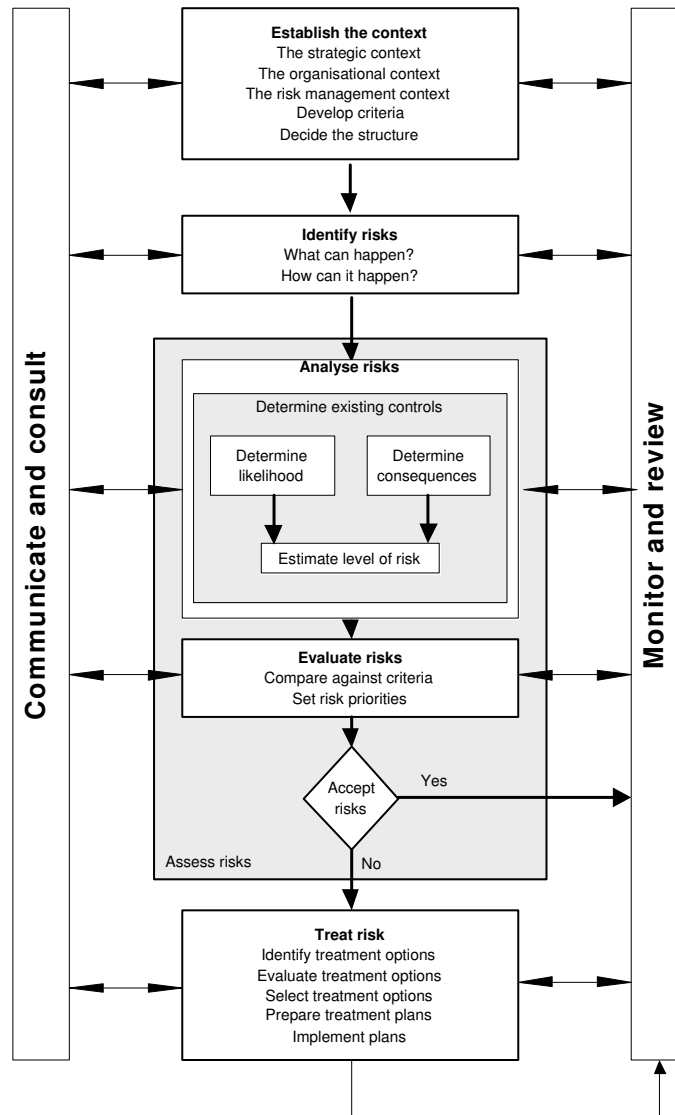


Figure 4.3: AS/NZS 4360: Risk management process [4]

phase the context directing the rest of the process is established and the criteria for which the risk will be evaluated against is defined. During the second phase, possible risks are identified, followed by the risk analysis phase and the risk evaluation phase. The fifth phase decides how to treat risks not accepted according to the evaluation criteria. The two parallel sub-processes focus on communication and consultation with stakeholders as well as monitoring and reviewing of each of the sequential sub-processes, giving rise to an iterative risk management process.

Section 4.3.1 to 4.3.7 briefly describes the sub-processes in the risk management process. Each

of the activities in a sub-process produces some sort of information that must be documented and passed on to the succeeding activities. An overview of important inputs and outputs to each of the activities in the five sequential sub-processes are presented in Figure 4.4. Only input and output explicitly mentioned by the standard are included in the figure. However, the figure indicates that in general output from one sub-process is used as input for the succeeding sub-process. For more information on the standard, the reader is referred to [4].

4.3.1 Establish the context

The first step in the risk management process focuses on the context in which the organisation operates. During this phase, the boundaries in which risks must be managed are established. The phase is divided into five activities:

Establish the strategic context focuses on the relationship between the organisation and its environment in order to identify elements which might support or impair the organizations ability to manage the risks it faces.

Establish the organizational context aims at understanding the organization and its capabilities, including its goals, objectives and strategies.

Establish the risk management context identifies the goals, objectives, strategies, scope and parameters of the risk management activity.

Develop risk evaluation criteria decides the criteria for which risk will be evaluated against. The criteria may be affected by legal requirements, the perceptions of external and internal stakeholders, or by organizational policies.

Define the structure divides the assessment process or project into a set of elements providing a logical framework for identification and analysis of risks [4].

4.3.2 Risk identification

The second step in the risk management process seeks to identify the risks to be managed. Systematic methods for risk identification like the ones described in section 4.2 is often applied at this stage. Identification of risks is critical as potential risks not identified at this stage is excluded from further analysis [4]. The process is divided into two activities:

What can happen identifies a list of unwanted incidents.

How and why it can happen identifies possible causes initiating the risks and scenarios demonstrating how the risk can occur.

4.3.3 Risk analysis

Based on information from sub-process 1 and 2, the impacts of all identified risks are assessed. The objective of this process is to separate the minor acceptable risks from risks requiring further treatment. Risk is analysed by combining estimates of consequence and likelihood in the context of existing control measures. Based upon available data, the analysis can be either qualitative or quantitative. Qualitative analysis [4] uses descriptive scales to describe the magnitude of potential consequences and the likelihood that those consequences will occur, while quantitative analysis [4] uses numerical values for both measures. The process is divided into the following activities:

Determine existing controls identifies the existing management, technical systems and procedures to control risk and assess their strengths and weaknesses.

Determine consequence and likelihood estimates the magnitude of consequences of an event, and the likelihood of the event and its associated consequences [4]. The assessment is performed in the context of existing controls.

Estimate level of risk estimates a level of risk based on estimated values for consequence and likelihood.

4.3.4 Risk evaluation

During this sub-process risks are evaluated by comparing the level of risk with the risk evaluation criteria established in sub-process 1. Based on the evaluation it is decided whether a risk is acceptable or not. Risks requiring further treatment are assigned a priority. Accepted risks should be monitored to ensure that they remain acceptable.

4.3.5 Risk treatment

The last sequential sub-process concerns treatment of risks classified as unacceptable.

Identify treatment options identifies options for treatment of risk. Four treatment options are defined: 1) Avoid, 2) Transfer in full or in part, 3) Reduce likelihood, and 4) Reduce consequence [4].

Consider feasibility costs and benefits compares the benefits of implementing a treatment option with the cost of that particular implementation.

Recommend treatment strategies evaluates the identified treatment options based on information about costs and benefits and by considering the risk evaluation criteria. The general rule is to choose the option making the risk as low as reasonably practicable.

Select treatment options selects treatment options to be implemented. In some cases, it may be necessary to use a combination of options.

Prepare treatment plan makes plans for how the selected treatment options will be implemented.

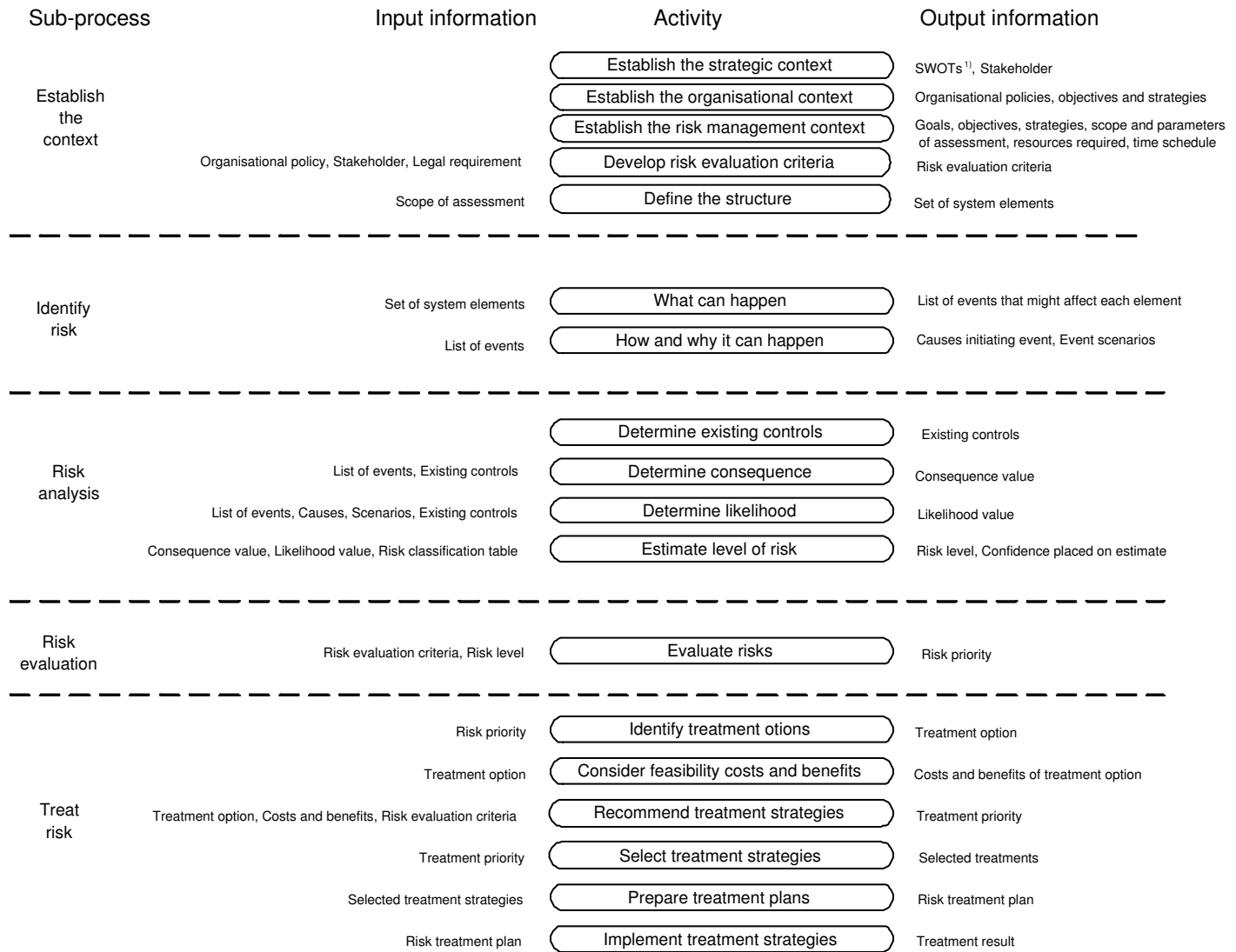
Implement treatment plan specifies how the implementation should be performed.

4.3.6 Monitoring and Review

This sub-process monitors and reviews the performance of the risk management process and changes which might affect it [4]. Strategies used to manage risks need to be constantly monitored and evaluated, while the risk management plan should be periodically reviewed to ensure that it remains relevant.

4.3.7 Communication and Consultation

This sub-process deals with communication and consultation with internal and external stakeholders as appropriate at each stage of the risk management process and concerning the process as a whole [4]. Continuous improvement of risk management requires communication and consultation to ensure that the risk management plan remains relevant for those involved.



¹ Strengths, Weaknesses, Opportunities and Threats

Figure 4.4: Information flow between activities in AS/NZS

4.3.8 Discussion

The risk management process defined by AS/NZS 4360 is refined in Chapter 6 in order to tailor the standard for use in the security domain. This is necessary as AS/NZS 4360 is generic and independent of any specific industry or economic sector [4]. For instance, we want to use security specific terms as threats, vulnerabilities and unwanted incidents. The refined process is used to define a security documentation framework, which forms the foundation for the requirement specification for SecurityAssessmentUML. Hence, it is of crucial importance that the security documentation framework specifies all necessary output from security assessments. Furthermore, we want all inputs to an activity to be specified as output from a previous activity in the assessment process. This requires a more detailed specification of information flow than

provided by AS/NZS 4360. Considering the illustration in Figure 4.4, it is clear that not all output information is explicitly mentioned. For instance, the standard does not specify that a detailed description of the target system, which is of crucial importance when identifying risks, is produced by any of the activities. Furthermore, legal requirements is mentioned as a necessary input for developing risk evaluation criteria, but none of the activities produce this information. For other activities, e.g. "Determine existing controls" input information is not defined. Instead the standard specifies possible tools for supporting the activity.

Figure 4.4 does not consider the information flow between the two parallel sub-processes in the risk management process. The reason for this is that the rest of this report solely focus on the five sequential sub-processes. Tailoring of the two parallel sub-processes to the security domain should be subject for further work. The output "Confidence on risk estimate" will not be further treated in this report as AS/NZS 4360 does not specify how this information shall be used.

Chapter 5

The CORAS Approach

5.1 Introduction to the CORAS approach

The overall objective of the CORAS project [13] is to provide an integrated methodology for precise, unambiguous, and efficient risk assessment of security critical systems [44]. The main focus of the framework is the concept of model-based risk assessment [54]. As illustrated in Figure 5.1, the framework is founded on four pillars: (1) A risk documentation framework based on RM-ODP, (2) A risk management process based on AS/NZS 4360, (3) An integrated risk management and development process based on UP, and (4) A platform for tool-inclusion based on XML.

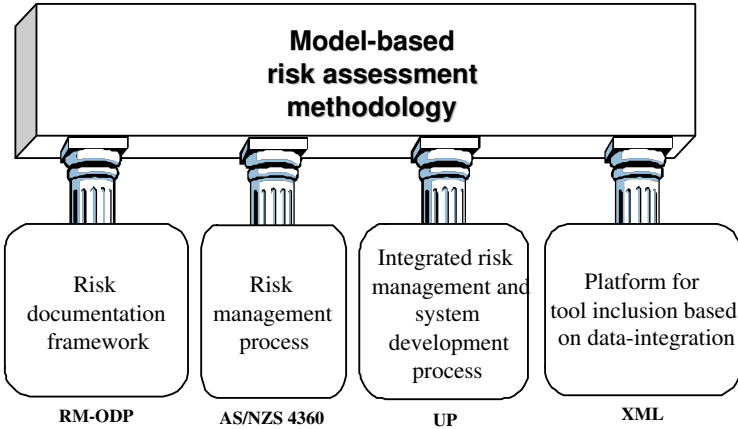


Figure 5.1: The CORAS framework [54]

The framework is model-based in the sense that it gives detailed recommendation for modelling using UML in conjunction with risk assessment. As illustrated in Figure 5.2, UML is used for three purposes [21]: (1) To improve precision of descriptions of the target system, (2) As a media for communication between stakeholders involved in a risk assessment, and (3) To document risk assessment results and the assumptions on which these results depend.

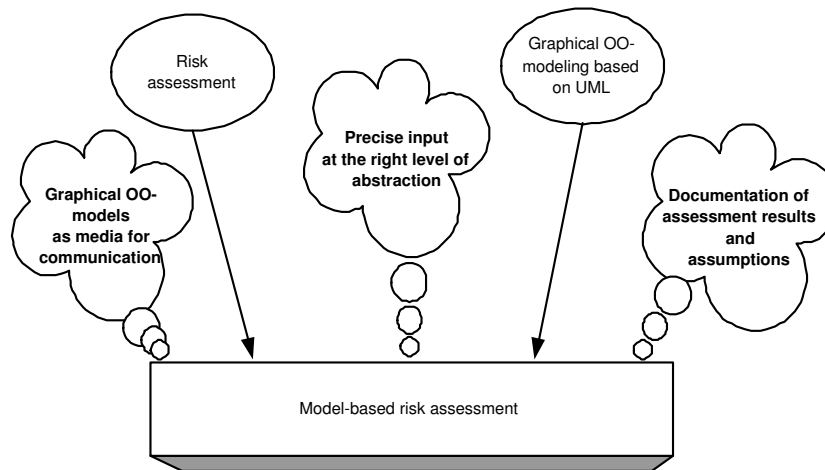


Figure 5.2: Model-based risk assessment [21].

Furthermore, the risk assessment methodology integrates aspects from partly complementary risk assessment methods: Hazard and Operability analysis (Hazop) [40], Fault Tree Analysis [24], Failure Modes, Effect and Criticality Analysis (FMECA) [10], Markov analysis [6], as well as CRAMM [7].

The CORAS methodology is asset-driven meaning that identification of assets is crucial in order to be able to perform risk assessment [15]. If no assets are identified, there is no need to perform a risk assessment.

The following subsections provide a brief introduction to two of the pillars of the CORAS methodology. Section 5.2 provides a brief introduction to the risk management process, and section 5.3 provides a brief introduction to the documentation framework. More information on these and other aspects of the CORAS approach can be found in [15, 21, 54, 44].

5.2 CORAS risk management process

The CORAS risk management process is based on the Australian/New Zealand standard AS/NZS 4360:1999 “Risk Management” [4], which was presented in Chapter 4.3 and the standard ISO/IEC 17799-1:1999 “Code of Practice for Information Security Management” [27]. Furthermore, the underlying terminology is supported by the two standards (1) ISO/IEC TR 13335-1: “Guidelines for the Management of IT Security” [26], and (2) IEC 61508: “Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems” [23].

As AS/NZS 4360, CORAS risk management process consists of seven sub-processes (see Figure 4.3). In order to adapt the standard for use in the security domain, CORAS has decomposed the five sequential sub-processes into activities as specified in Table 5.1. For each of these sub-processes, the CORAS methodology provides guidelines with respect to which models should be constructed, and how they should be expressed [11]. The major change from AS/NZS 4360 is that CORAS risk management process is asset-driven meaning that assets identified in Activity 1.2 guide the whole risk assessment process from Activity 1.3 onwards [37].

Table 5.1: The activities of the CORAS risk management process [21]

<p>Sub-process 1: Identify Context</p> <p>Activity 1.1: Identify areas of relevance</p> <p>Activity 1.2: Identify and value assets</p> <p>Activity 1.3: Identify policies and evaluation criteria</p> <p>Activity 1.4: Approval</p>	<p>Sub-process 4: Risk Evaluation</p> <p>Activity 4.1: Determine level of risk</p> <p>Activity 4.2: Prioritise risks</p> <p>Activity 4.3: Categorise risks</p> <p>Activity 4.4: Determine interrelationships among risk themes</p> <p>Activity 4.5: Prioritise the resulting risk themes and risks</p>
<p>Sub-process 2: Identify Risks</p> <p>Activity 2.1: Identify threats to assets</p> <p>Activity 2.2: Identify vulnerabilities of assets</p> <p>Activity 2.3: Document unwanted incidents</p>	<p>Sub-process 5: Risk Treatment</p> <p>Activity 5.1: Identify treatment options</p> <p>Activity 5.2: Assess alternative treatment approaches</p>
<p>Sub-process 3: Analyse Risks</p> <p>Activity 3.1: Consequence evaluation</p> <p>Activity 3.2: Frequency evaluation</p>	

5.3 CORAS risk documentation framework

The CORAS system documentation framework is a specialization of the ISO/IEC 1074 standard “Basic Reference Model for Open Distributed Processing” (RM-ODP) [25], and can be understood as a reference framework for model-based risk assessment. The parts of RM-ODP that are directly relevant for risk assessment of security critical systems are refined [15]:

- The RM-ODP terminology is extended with concepts for risk assessment and security.
- The RM-ODP viewpoint structure is divided into 22 cross-viewpoint concerns targeting security in general and model-based risk assessment in particular. As illustrated in Figure 5.3 each concern is assigned a specific activity in the risk management process and links together related information within the five viewpoints. Furthermore, each concern is decomposed into models, providing the content of the concern with respect to a particular viewpoint. For each model guidelines for its development exist.

Moreover, the CORAS risk documentation framework provides libraries of reuse of reusable elements [53].

5.4 The CORAS UML-Profile for Model-based Risk Assessment

The CORAS UML Profile for model-based risk assessment is defined as a specialisation of UML1.4 [1] for risk assessment [21]. The profile, which defines UML stereotypes and rules for specialized UML diagrams, is developed to support the model-based risk assessment process of the CORAS project. The profile consists of six packages [21]:

- **Actors Package:** defines actor stereotypes
- **SWOT Model Package:** defines SWOT diagrams
- **Asset Model Package:** defines Asset diagrams
- **Threat Model Package:** defines Threat diagrams

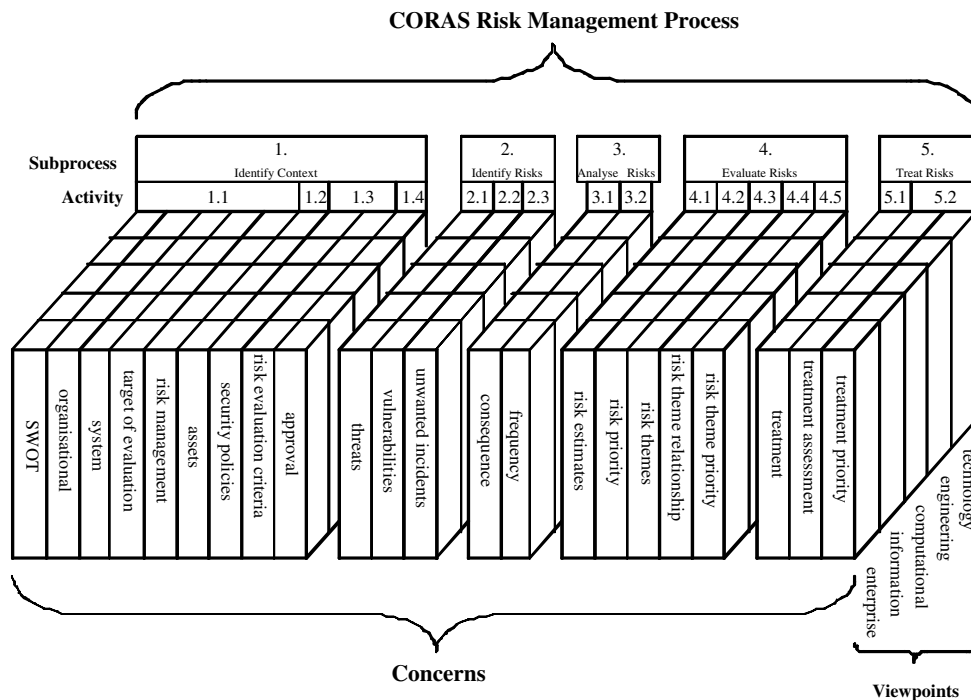


Figure 5.3: The CORAS risk documentation framework [21].

- **State Analysis Model Package:** defines State Analysis diagrams
- **Treatment Model Package:** defines Treatment diagrams, which are extensions of Threat diagrams

In the following the two diagrams supporting risk identification and risk analysis of security critical systems are described. For information about SWOT, Asset or Treatment diagrams, the reader is referred to [21].

5.5 Threat diagrams

Threat diagrams are specialized use case diagrams used for documenting threats to and vulnerabilities of assets [21]. An example threat diagram is shown in Figure 5.4. A threat is related to the asset it is a threat against and, if the threat involves actions carried out by humans, to possible user or mis-user roles. Textual descriptions, sequence diagrams or activity diagrams can be used to further specify the threats,

In Threat diagrams, the asset notation includes information about vulnerabilities of the asset, i.e. undesired attributes or operations of the asset. As indicated in the figure, vulnerabilities are depicted as coloured attributes or operations.

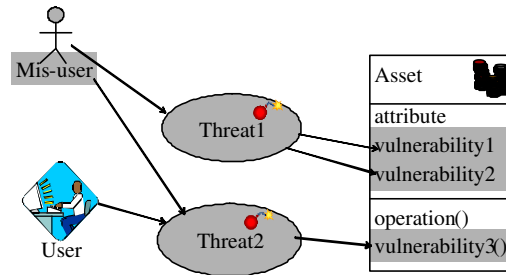


Figure 5.4: Threat diagram. Generalized from [21].

5.6 State Analysis diagrams

State Analysis diagrams are extended UML State diagrams used to support consequence and frequency evaluation [21]. State analysis diagrams specify undesired, as well as desired, behaviour of the system. Potential misbehaviour is modelled as undesired transitions and states. An example diagram is shown in Figure 5.5. As illustrated in the figure, undesired transitions are triggered by unwanted incidents, which are modelled as a specialization of the UML concept event. Documentation of consequence and frequency of the risk is made by attaching severity and likelihood values to bad states and unwanted incidents, respectively.

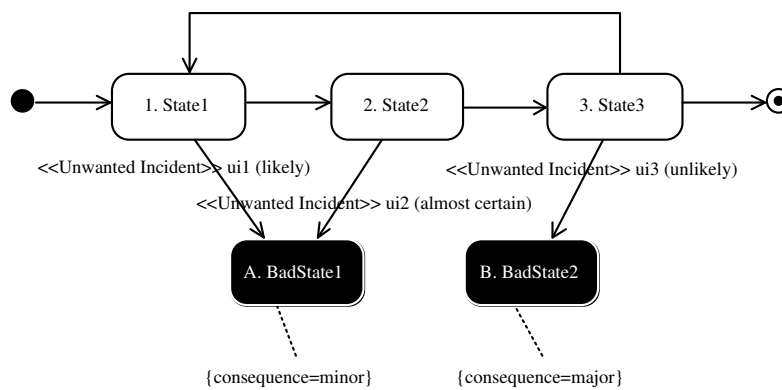


Figure 5.5: State Analysis diagram [51]

State Analysis diagrams may be used to generate a state probability matrix for further analysis using Markov analysis and Monte Carlo simulation [47].

5.7 Discussion

This section discusses the CORAS UML profile as a means for specifying output from sub-process 2 and 3 of the CORAS risk documentation framework, risk identification and risk analysis. As indicated in Figure 5.3, five concerns must be documented; 1) threats, 2) vulnerabilities, 3) unwanted incidents, 4) consequence evaluation, and 5) frequency evaluation.

All five concerns are to some extent covered by the profile. As depicted in Figure 5.4, vulnerabilities and threats are specified in Threat diagrams. In terms of vulnerabilities, threat diagrams express where in the system the vulnerability is, i.e. which asset that contains the vulnerability. In terms of threats the diagram specifies which vulnerabilities a threat may exploit, as well as a list of actors that may initiate the attack. Specification of the three remaining concerns, unwanted incidents, consequence evaluation and frequency evaluation, are supported by State analysis diagrams (see Section 5.5).

Let us first discuss concerns belonging to sub-process 2; threats, vulnerabilities and unwanted incidents. In addition to what is already covered by CORAS UML profile, we want to be able to express concrete scenarios demonstrating how an attack actually occurs. This means that we want to specify threat-vulnerability-pairs and unwanted incidents in one single diagram, clearly depicting the sequence of actions involved in a security attack. This information is not possible to depict in use cases, as these diagrams only describe functionality.

Let us now consider the two concerns consequence and frequency. In terms of risk analysis, CORAS UML profile supports use of Markov diagrams. These diagrams may be hard to specify as they require a complete list of normal and bad states, as well as possible transitions between these states. This is particularly problematic for complex systems, which may have a large number of states [35].

Chapter 6

Security Assessment

Security assessment is used as a basis for deciding how to handle security threats in security critical systems. This chapter presents a refinement of the risk assessment process defined by AS/NZS 4360, inspired by the CORAS methodology. In particular, the assessment is asset-oriented, meaning that assets identified guide the rest of the assessment. This also means that if no assets are identified, there is no need to continue the assessment.

In order to establish a common means for communication between stakeholders we need clearly defined terms as well as a common understanding of how these terms relate to each other. Definitions of terms used within security assessment is presented in Section 6.1 and the ontology illustrating the relationship between the terms is presented in Section 6.2. The refined security assessment process and security documentation framework is presented in Section 6.3 and 6.4, respectively; and requirements for SecurityAssessmentUML are specified in Section 6.5.

6.1 Terminology

Asset - Something to which an organisation directly assigns value and, hence, for which the organisation requires protection [5]. As illustrated in Figure 6.1 an asset is uniquely identified by the triple entity, value and stakeholder.

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 loss, loss of revenue, market share, or company image [5].

Consequence - The outcome of an event expressed qualitatively or quantitatively, being a loss, injury, disadvantage or gain. There may be a range of possible outcomes associated with an event [4].

Context - The strategic, organisational and risk management context in which the rest of the risk management process will take place [4].

Entity - An entity that becomes an asset when assigned value by a stakeholder [37].

Frequency - A measure of the rate of occurrence of an event expressed as the number of occurrences of an event in a given time [4].

Likelihood - Used as a qualitative description of probability or frequency [4]

Misuse - A function that the system should not allow. Some kinds of misuse are most likely to be performed by intent whereas other may happen accidentally. Some require insiders or people with enormous skill, others not [50].

New definition: **Misuse** - An action that violates the security of a system. Some kinds of misuse are most likely to be performed by intent whereas other may happen accidentally. Some require insiders, other types of misuse may be performed from external locations.

Mis-actor - An actor who initiates functions that the system should not allow [50].

New definition: **Misuser** - An actor who initiates system misuse. The misuser may be either internal or external to the organisation.

Organization - A company, firm, enterprise, or other legal entity or part thereof, whether incorporated or not, public or private, that has its own function(s) and administration [4].

Probability - The likelihood of a specific event or outcome, measured by the ratio of specific events or outcomes to the total number of possible events or outcomes. Probability is expressed as a number between 0 and 1, with 0 indicating an impossible event or outcome and 1 indicating that an event or outcome is certain [4].

Risk - The potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organisation [26]. It is measured in terms of consequence and likelihood [4].

Risk evaluation criteria - A criteria against which risk is to be evaluated [4].

Risk level - Classification of risk associated with a particular unwanted incident. The classification is calculated based on estimated values for consequence and likelihood/frequency [55].

Risk priority - Value indicating the necessity for further treatment of the risk [4].

Risk treatment - Selection and implementation of appropriate options for dealing with risk [4].

Safeguard - A practice, procedure or mechanism that reduces risk [26].

Security attack - Any action that comprises the security of information owned by an organisation [52]

IT Security policy - Rules, directives and practices that govern how assets, including sensitive information, are managed, protected and distributed within an organization and its IT systems [26].

New definition: **Security policy** - Rules, directives and practices that govern how assets, including sensitive information, are managed, protected and distributed within an organization and its information systems.

Stakeholders - Those people or organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity [4].

Target of Evaluation (ToE) - An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation [14]).

Threat - A potential cause of an unwanted incident which may result in harm to a system or organisation [26].

Unwanted incident - An undesired event that may reduce the value of an asset [37].

Vulnerability - A weakness of an asset or group of assets which can be exploited by one or more threats [26].

6.2 Ontology

The purpose of this section is to illustrate the relationships between the terms used within a security assessment, the ontology of a security assessment. A clear and unambiguous understanding of the relationships between the concepts used is a necessity for efficient communication and interaction among the stakeholders participating in the assessment.

Security assessment is focused on assets, which represent valuable entities in the target of evaluation. As illustrated in Figure 6.1, each asset is uniquely identified by the entity it represents, the stakeholder that has an interest in the entity and the value perceived of the entity by that specific stakeholder. Thus, one entity can be represented as several assets, one for each stakeholder who has an interest in the entity. The rationale for this representation is to allow documentation of different perceptions of values of entities. This is important because most often stakeholders having an interest in an asset will also pay for the countermeasures needed. Hence, at this stage it is crucial that relations between perceived asset values and stakeholders are documented.

The target of evaluation is influenced by its context. The relationship between the target of evaluation, context, information system, organisation and society was presented in Figure 2.1. Stakeholders perception of asset value is influenced by the security policy of the organisation the stakeholder belongs to. The security policies should be reflected by all applications used by the organisation.

Figure 6.1 presented the relationships between assets and their environment. In figure 6.2 we look at undesirable behaviour. In security critical systems undesirable behaviour can be divided into three categories, threats, vulnerabilities and unwanted incidents. Threats initiate potential

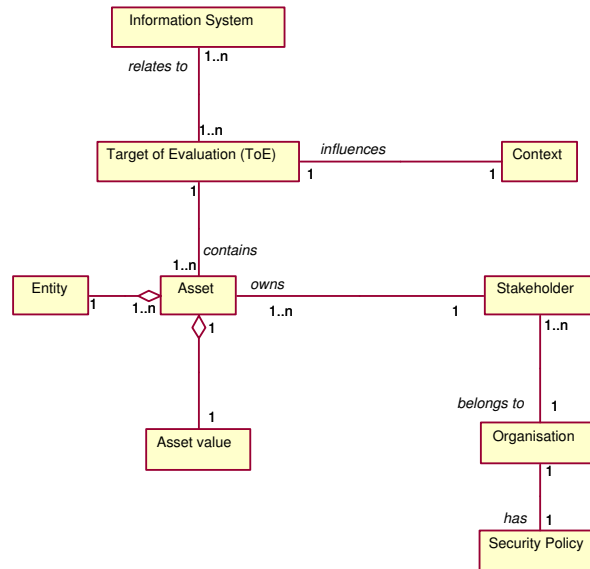


Figure 6.1: The relationship between assets and their environment

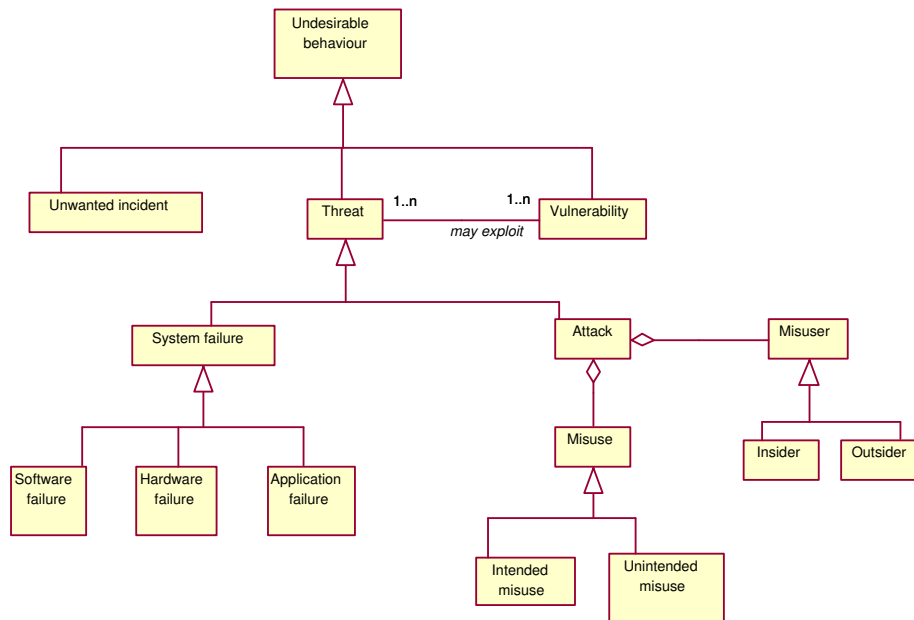


Figure 6.2: Undesirable system behaviour

undesirable behaviour. A vulnerability is a weakness in the information system that might be exploited by threats. Unwanted incidents may only occur as a result of a prior exploitation of vulnerability. Threats may be categorised as either system failure or attack. System failure is failure not influenced by human behaviour. System failure can be categorised into hardware

failure, software failure and application failure. An attack consists of misuse and a misuser. Misuse are unauthorised use of the system, either intended or unintended. Misuse can be initiated by either external or internal users. In the rest of this report we solely focus on threats categorised as attacks. Thus, system failure is not further emphasized.

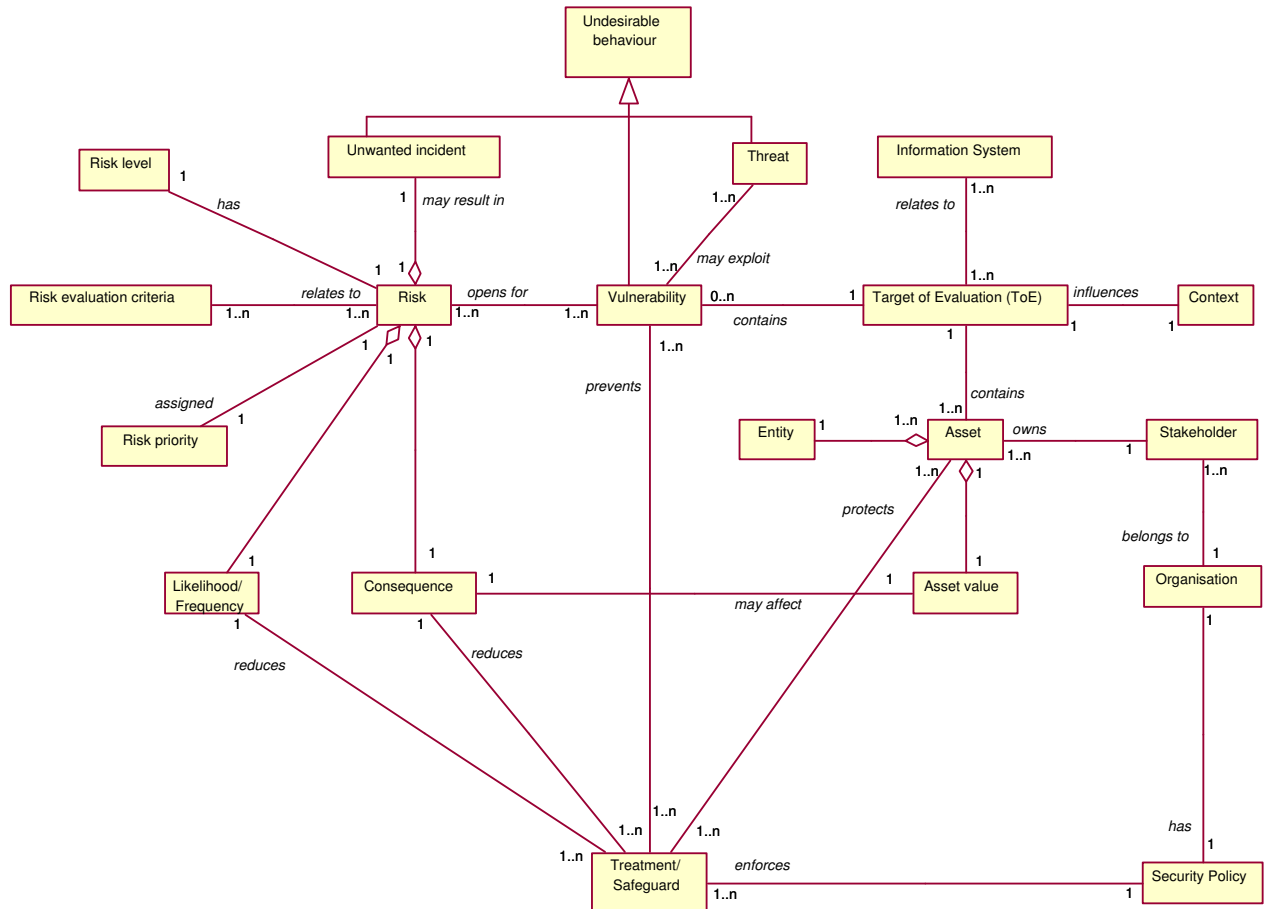


Figure 6.3: Ontology for risk concepts

Figure 6.3 illustrates the relationships between concepts related to risk. The presence of one or more vulnerabilities opens for risks. Risk is the combination of one unwanted incident and its associated values for consequence and frequency/likelihood. The consequence of a risk may affect the value of one or more assets. Based on the values for consequence and frequency/likelihood, the risk is assigned a risk level. The risk level is compared to the risk evaluation criteria, determined by system stakeholders. The criteria characterize acceptable reduction of asset value. Risk priority indicates the necessity for further treatment of risk. Risk treatment may reduce the consequence of the risk, the frequency/likelihood of the risk or both. A safeguard is a treatment that is implemented. Hence, safeguards work preventatively.

Figure 6.4 provides an overview of the relationship between the figures presented above. The relationship between threats, vulnerabilities, unwanted incidents and safeguards is further emphasized in Figure 6.5. As depicted in the diagram, unwanted incidents may only occur as a

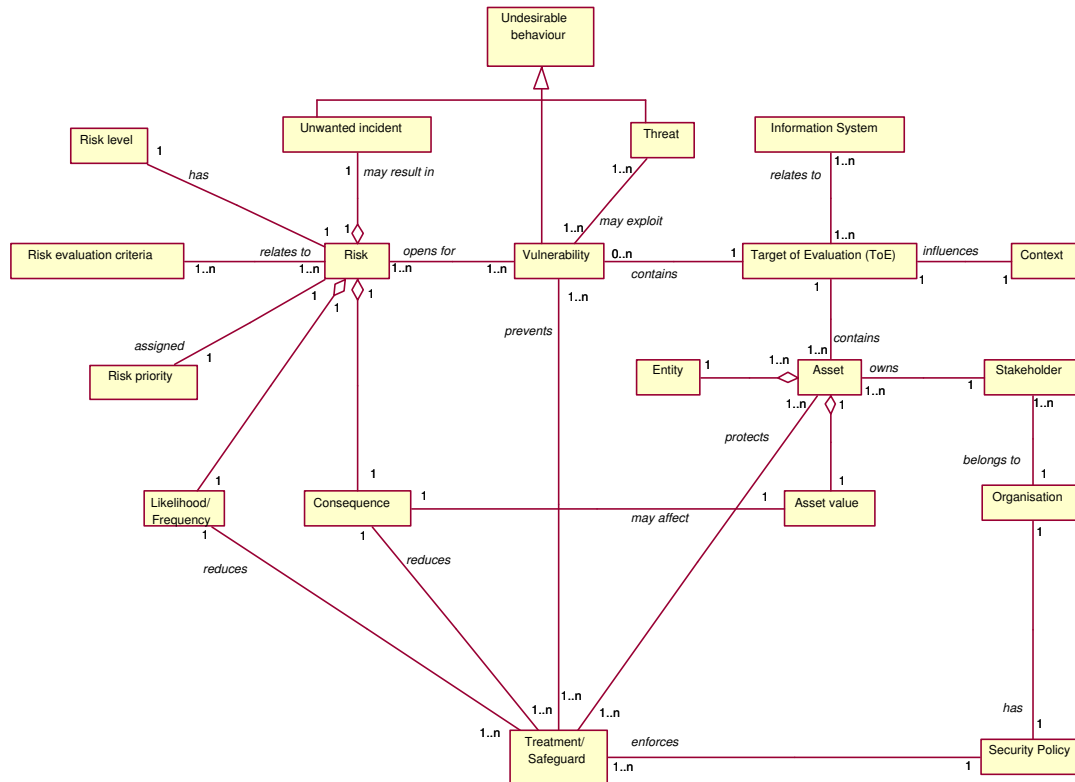


Figure 6.4: Ontology for security assessment

result of a prior exploitation of vulnerability by one or more threats. Furthermore, the diagram illustrates that exploitation of vulnerabilities in information systems may be prevented by implementing one or more safeguards. As illustrated in the figure, three different scenarios are possible when a threat is initiated: 1) A threat is initiated but nothing happens since no vulnerability exists, 2) An initiated threat exploits a vulnerability in the system leading to an unwanted incident, and 3) An initiated threat is prevented from exploiting a vulnerability by a safeguard.

6.3 Process for security assessment and security management

This section presents a refinement of the risk management process defined by AS/NZS 4360 in order to particularly support security issues. Section 6.3.1 presents the refined process while Section 6.3.2 presents the information flow between the activities in the refined process. The refinement is inspired by the CORAS risk management process.

6.3.1 Security management process

The refined process is shown in Figure 6.6, which depicts sub-processes with belonging activities. The process is refined according to practice in the security domain.

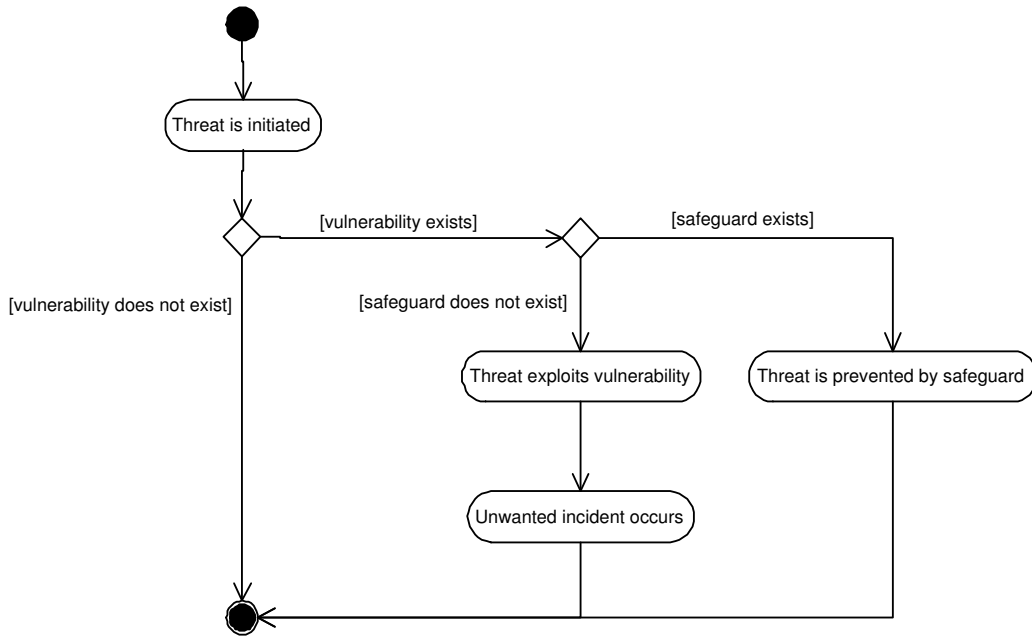


Figure 6.5: Relationship between threats, vulnerabilities, safeguards and unwanted incidents [22]

The main change from the process presented in Figure 4.3 is that the refined process incorporates the concepts of assets and vulnerabilities. This is indicated through the introduction of the activity *Identify and value assets*, and through the activities contained within the sub-process *Identify risks*. The activity of identifying and valuing assets is crucial when performing a security assessment, as the process is asset-driven. Thus, assets not identified at this stage will not be considered later in the assessment. Equivalently, there is no need to continue the assessment if no assets is identified.

As defined in this thesis, the two activities addressing identification of vulnerabilities and identification of threats are performed in sequence. Hence, threat identification is performed with identified vulnerabilities in mind. Using this approach there is a chance that some threat-vulnerability-scenarios are missed. Thus, in some circumstances the approach should be supplemented with a traditional threat-identification. Alternatively, the approach defined by CORAS, viewing the two activities as complementary, could be followed.

Another change from the process defined by AS/NZS4360 is the ordering of activities performed during context-identification. The Australian standard for risk management considers the strategic and organisational context on a general basis before taking the specific target of evaluation into account. In Figure 6.6 this order is changed, to allow an assessment process restricting the context-analysis to the parts of the organisation or environment that directly affects the system under consideration. The former approach is advantageous in static organisations working within static environments as the information can be reused. For an organisation that frequently undergoes change, however, such an approach will be very costly and time consuming as there is less chance for the information to be reused. Thus, in order to choose a profitable approach, organisations should have this in mind when planning a security assessment.

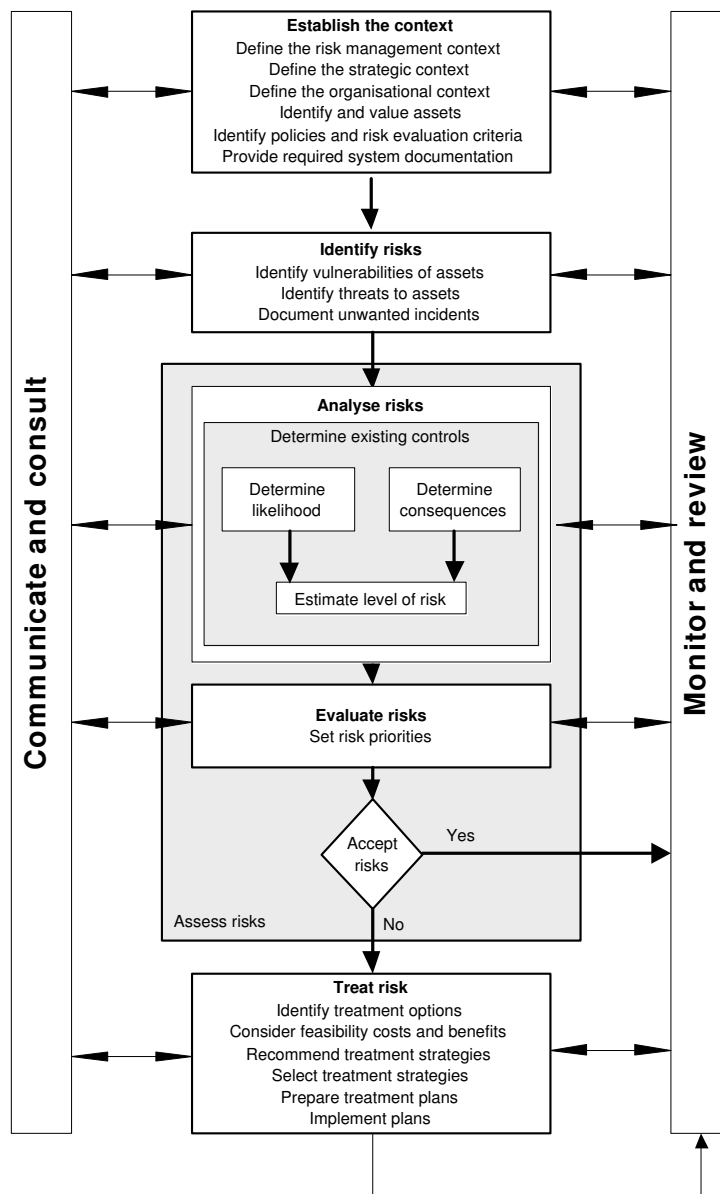


Figure 6.6: Security management process

6.3.2 Information flow between activities

The previous section presented a refined version of the risk management process in order to particularly support a security assessment. Having refined the management process, it is necessary to update the information flow between the activities within this process. In the following only the five sequential sub-processes representing security assessment will be considered. An overview of the information flow in the refined process is presented in Figure 6.7. The figure is based on the information flow presented in Figure 4.4, which is built on AS/NZS 4360, but refined according to the security assessment process. Compared to AS/NZS 4360 more detailed specifications of input and output are provided. In particular, all information required as input to one activity are specified as output from a previous activity.

In the next section, each of the concerns present in the figure is assigned a specific activity forming the basis for the final security documentation framework.

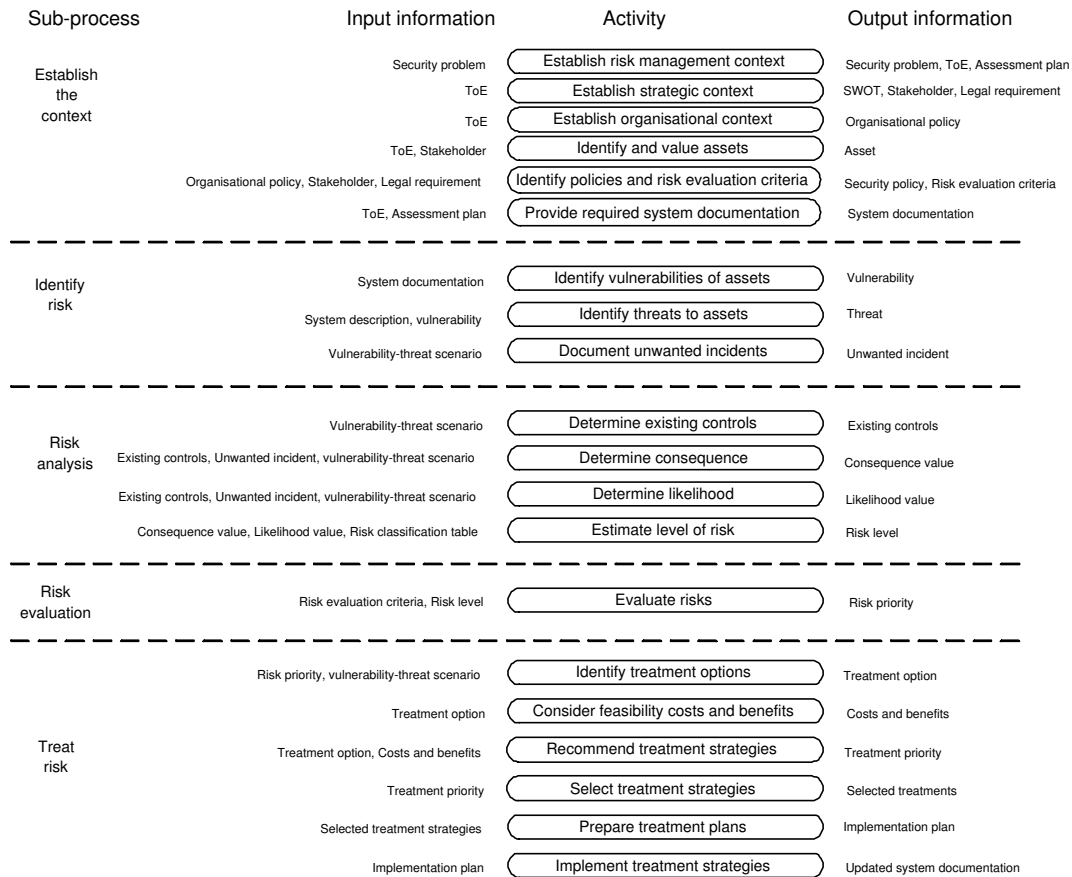


Figure 6.7: Informationflow in the security management process

6.4 Security documentation framework

This section presents the security documentation framework used as a basis for specifying requirements for SecurityAssessmentUML. The documentation framework is divided into 25 concerns, representing the main aspects that must be documented throughout a security assessment. The concerns are structured according to the security management process presented in Figure 6.6 and is assigned a specific activity contained within a sub-process. Concerns which are used as input for more than one activity, are assigned to the first activity under which the concern is mentioned. The final structure is provided in Figure 6.8, and each concern is briefly described in Table 6.1.

Sub-process	Activity	Concern
Establish the context	Establish risk management context	Security problem
		Target of Evaluation (ToE)
		Assessment plan
	Establish strategic context	SWOT
		Stakeholder
	Establish organisational context	Legal requirements
		Organisational policies
Identify and value assets	Assets	
Identify policies and risk evaluation criteria	Security policy	
	Risk evaluation criteria	
Provide required system documentation	System documentation	
Identify risk	Identify risks	Vulnerabilities
		Threats
		Unwanted incidents
Risk analysis	Determine existing controls	Existing controls
	Determine consequence	Consequence
	Determine likelihood	Likelihood
	Estimate level of risk	Risk level
Risk evaluation	Evaluate risks	Risk priority
Treat risk	Identify treatment options	Treatment option
	Consider feasibility costs and benefits	Costs and benefits
	Evaluate treatment options	Treatment priority
	Select treatment options	Selected treatments
	Prepare treatment plans	Implementation plan
	Implement treatment plans	Updated system documentation

Figure 6.8: Security documentation framework

Table 6.1: Description of concerns

Concern	Description
Security problem	The security problem represents the reason for performing a security assessment.
Target of Evaluation	The ToE is a subset of the components in an information system (see Figure 2.1).
Assessment plan	Descriptions of goals and objectives for the study, time schedule and a list of required studies.
SWOT	A list of SWOTs and information about how each SWOT-element might support or impair the organisations ability to manage the risks it faces.
Stakeholder	A list of stakeholders related to the ToE.
Legal requirement	A list of legal requirements that might have an influence on the ToE.
Organisational policy	A list of policies that might have an influence on the ToE.
Asset	A list of assets present in the ToE, including information about their values and stakeholders.
Security policy	A list of security policies that might have an influence on the ToE.
Risk evaluation criteria	A value against which risk is to be evaluated. One criteria is defined for each asset.
System documentation	The system documentation needs to cover information necessary for performing a security assessment of the specified ToE. The documentation should consist of UML diagrams describing the ToE, as well as a link to elements in the context that might affect the ToE.
Vulnerability	A list of weaknesses in the information system that might be exploited by one or more threats. The documentation should include information about where the vulnerability is.
Threat	A list of events that might exploit vulnerabilities and thus lead to unwanted incidents. Threats should be documented as threat scenarios.
Unwanted incident	Unwanted incidents may only occur as a result of a prior exploitation of a vulnerability by a threat. Thus, unwanted incidents should be linked to vulnerability-threat pairs.
Existing control	A list of controls introduced to the system in order to reduce risk. The documentation needs to cover information about which asset the existing control protects.
Consequence	A qualitatively or quantitatively expression of the outcome of an unwanted incident. In addition to the consequence value the documentation should include information about impact on asset value.
Frequency/likelihood	Quantitative measure or qualitative description of the rate of occurrence of an unwanted incident.
Risk level	A classification value of risk based on estimated values for frequency/likelihood and consequence.
Risk priority	A value indicating the necessity for further treatment of the risk.
Treatment option	A description of alternative treatment approaches.
Costs and benefits	A description of the results of cost-benefit analysis of each treatment option. The analysis is concerned about the effect on assets involved.
Treatment priority	Priority ordering of treatments established based on the result of the cost-benefit-analysis.
Selected treatments	A list of selected treatments, including description on how each treatment affects the system and the risks it is intended to treat.
Implementation plan	A list of responsibilities, schedules, expected outcomes of treatments, budgeting information, performance measures and the review process to be set in place.
Updated system documentation	Updated system documentation relates to the proposed treatments.

6.5 Requirements for the UML profile for security assessment

As mentioned in the introduction, SecurityAssessmentUML should cover all required output from sub-process 2 and 3 in the risk management process. In particular, the profile shall support specification of concrete scenarios demonstrating how attacks may occur, as well as specification of consequence, frequency and risk level of unwanted incidents. Hence, as a minimum the profile must be able to support the following information: 1) Vulnerabilities, 2) Threats, 3) Unwanted incidents, 4) Existing controls, 5) Consequence value, 6) Frequency value, and 7) Risk level. Furthermore, the profile needs to support documentation of how these concepts relate to each other. To illustrate relationships between the concepts, the complete ontology for risk identification and risk analysis is depicted in Figure 6.9. The figure is a segment of the complete ontology for security assessment as presented in Figure 6.4. Although belonging to sub-process 1, the concept asset is included as vulnerabilities relates to assets and because safeguards protect assets. The concept asset value is included because this value is affected by the consequence of an unwanted incident.

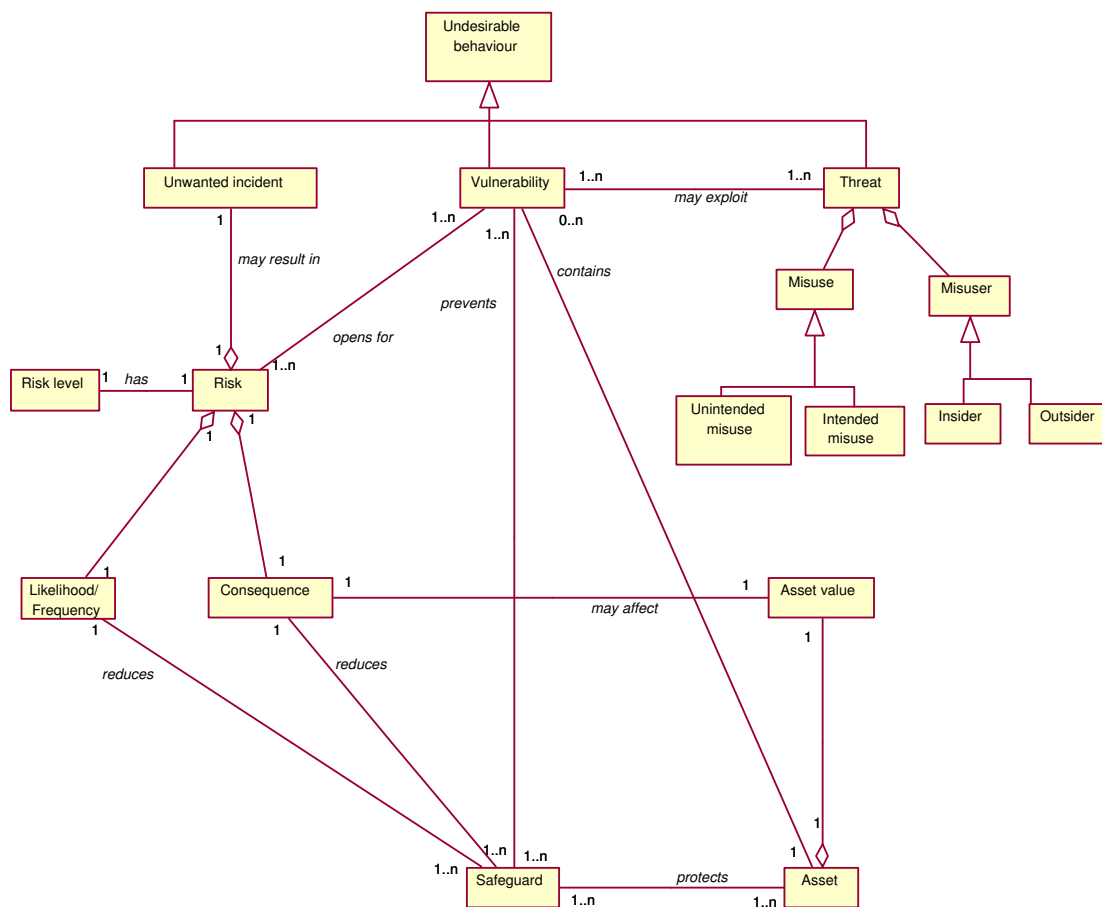


Figure 6.9: Ontology for risk identification and risk analysis

We also want the profile to support specification of how frequency is calculated. To achieve this,

we want modelling support for basic fault tree notation. Requirements related to sub-process 2 is specified in Section 6.5.1 and requirements related to sub-process 3 is specified in Section 6.5.2. First, however, we provide an overview of output that should be supported by a UML profile supporting the three remaining sub-processes. Development of a UML profile covering all these aspects should be subject for further work. In terms of sub-process 1, 11 concerns must be covered; 1) Definition of the security problem, 2) Definition of the target of evaluation, 3) Assessment plan, 4) Description of SWOTs, 5) Stakeholders, 6) Legal requirements, 7) Organisational policies that might affect the ToE, 8) Assets and their values, 9) Security policies, 10) Risk evaluation criteria, and 11) System description. Documentation of SWOTs, stakeholders and assets are supported by CORAS UML profile, while system descriptions can be provided using UML1.4. A UML profile supporting sub-process 4 must be able to express the concern risk priority. In terms of sub-process 5, six concerns must be expressed; 1) Treatment option, 2) Costs and benefits associated with a treatment option, 3) Recommended treatment strategies, 4) Selected treatment strategies, 5) Implementation plan, and 6) Updated system documentation. Treatment options can be specified by Treatment diagrams as defined by CORAS UML profile.

6.5.1 Sub-process 2: Identify risk

According to the security assessment framework presented in Figure 6.8, a UML profile supporting documentation of all aspects related to risk identification must cover the three concerns *vulnerability*, *threat* and *unwanted incident*. None of these three concerns are explicitly supported by UML1.4. Through threat and state analysis diagrams, CORAS UML profile covers all concepts, but the profile does not support documentation of concrete scenarios demonstrating the sequence of events or actions involved from a threat is initiated to an unwanted incident occurs. This is what we want to support in SecurityAssessmentUML. In order to demonstrate the lack of concrete and uniform support in UML1.4, we will model an example scenario using UML sequence and UML activity diagrams as defined by UML1.4.

Let us consider the example system illustrated in Figure 6.10. The figure depicts the connection between the local email server in an organisation and the Internet. As illustrated in the figure, the Internet is connected to the mail-gateway through an ordinary router. The mail-gateway runs the SMTP service, receives all incoming emails, and forwards the emails to the mail-server. The router works as a passiv interface between the Internet and the local area network (LAN). In most cases a firewall would have been placed between the Internet and the mail-gateway in order to protect the internal network from the outside world (the Internet). However, in the system described here, the firewall is omitted for simplicity reasons. Now, assume that an

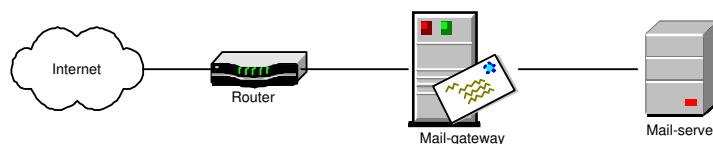


Figure 6.10: Example system

outsider sends an email containing a virus to an address in the LAN. Since no check on the emails is performed, the virus is forwarded to the email server of the organisation and the server is infected.

Figure 6.11 shows the scenario using existing UML sequence diagrams, and Figure 6.12 models

the scenario using UML activity diagrams. None of the diagrams emphasize the difference

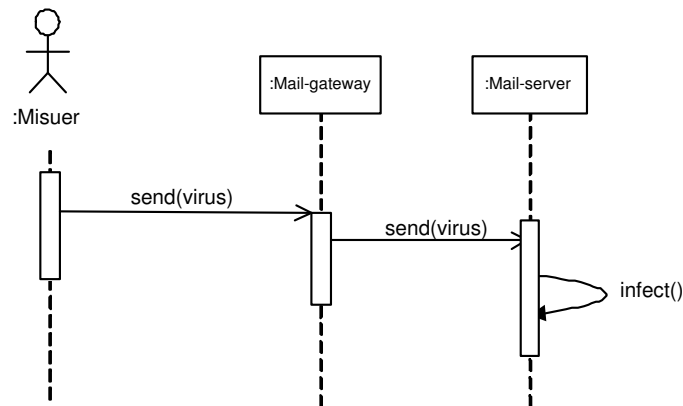


Figure 6.11: Sequence diagram showing events using UML1.4

between normal and undesirable behaviour. It is possible to include additional information emphasizing undesirable behaviour by using notes [9] or the tagged value documentation [9], two standard elements of UML1.4. However, we aim for a tailored and uniform way of expressing this information.

Let us first consider the sequence diagram. The figure shows that a message containing a virus is sent from an attacker to the mail-server via a mail-gateway. The diagram does not differentiate vulnerabilities, threats or unwanted incidents from normal aspects of the system. Furthermore, misusers are not distinguished from normal users. Thus, to improve the representation of the sequence diagram we want to indicate that a *misuser* initiates a *threat* by sending the message “virus”, which exploits a *vulnerability* of the mail-gateway (no anti virus program included), leading to the *unwanted incident* “mail-server infected”. In some cases it may also be advantageous to indicate whether the attack is intended or unintended and whether the misuser is external or internal to the organisation. Table 6.2 sums up existing and not supported modelling requirements in UML sequence diagrams.

Table 6.2: Required modelling support in UML sequence diagrams for documenting output from risk identification

Concept	Mapping to existing UML	Not supported
Vulnerability	Object	Particular expression power to specify vulnerabilities. Furthermore, it should be possible to express which entity the vulnerability is contained within.
Unwanted incident	Message	Particular expression power to specify unwanted incidents
Misuser	Actor instance	Particular expression power to specify misuser. Furthermore, it should be possible to express whether the misuser is internal or external to the organisation and whether the misuse is intended or unintended.
Threat	Message	Particular expression power for threats.

Now, let us consider the activity diagram presented in Figure 6.12. The figure specifies the activities involved when a misuser sends a message containing a virus to the organisation. The diagram indicates that if no antivirus program is installed, the message will be forwarded to

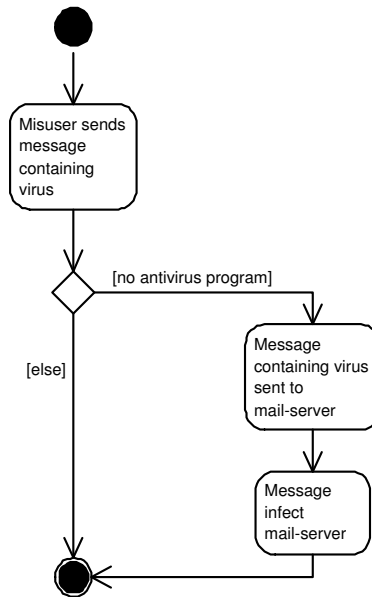


Figure 6.12: Activity diagram showing events using UML1.4

and infect the mail-server. However, the diagram does not differentiate the representation of vulnerabilities, threats or unwanted incidents from the representation of normal aspects of the system. Hence, we want specific support for specifying that an action possess a *threat* to the system or represents an *unwanted incident*, and that the guard associated with a conditional branch checks for a *vulnerability* in the system. Table 6.3 sums up existing and not supported modelling requirements in UML activity diagrams.

Table 6.3: Required modeling support in UML activity diagrams for documenting output from risk identification.

Concept	Existing support UML	Not supported
Vulnerability	Guard	Particular expression power to specify vulnerabilities. Furthermore, it should be possible to express which entity the vulnerability is contained within.
Threat	Activity state	Particular expression power to specify threats.
Unwanted incident	Activity state	Particular expression power to specify unwanted incidents.

6.5.2 Sub-process 3: Risk analysis

According to the security assessment framework presented in Figure 6.8, a UML profile supporting documentation of aspects related to risk analysis must cover the four concerns *existing controls/safeguards*, *consequence*, *likelihood* and *risk level*.

Activity 3.1: Safeguards

Activity 3.1 deals with identification of safeguards. For each safeguard we need to document which system entity the safeguard is installed on, which vulnerability it protects, which threat it prevents, and the level of protection offered. Neither UML1.4 nor CORAS UML profile provides explicit modelling support for documentation of safeguards.

Let us again consider the example scenario described under sub-process 2. Now assume that the mail-gateway includes an anti virus-program reducing the likelihood that a virus is passed undetected. Figure 6.13 shows the scenario using UML activity diagrams. As illustrated in the figure, the virus is detected and discarded if the protection offered by the anti virus program is adequate.

In the rest of this section we concentrate our discussion on the part of the diagram involving the safeguard. As illustrated in the figure, safeguards can be documented as guards connected

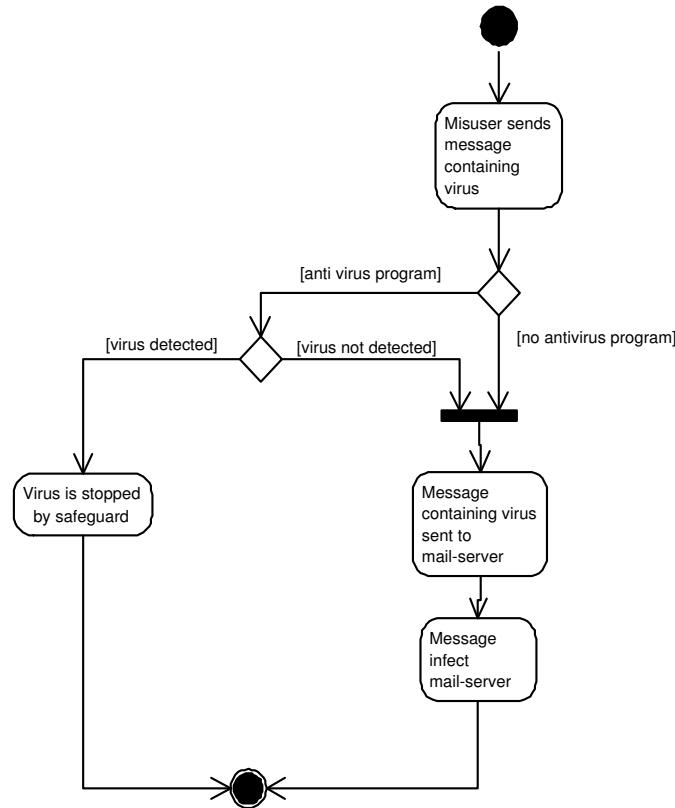


Figure 6.13: Activity diagram showing events using UML1.4

to branches. The context indicates which threat the safeguard prevents and which vulnerability it protects. However, no uniform notation for specifying the presence of safeguards is provided. Furthermore, from the diagrams it is not possible to identify the level of protection offered or which entity the safeguard is installed on. Requirements for specifying safeguards in UML activity diagrams are summed up in Table 6.4.

Table 6.4: Required modelling support in UML activity diagrams for documenting safeguards

Concept	Existing support UML	Not supported
Safeguard	Guard	Particular expression power to specify safeguards. Furthermore, it should be possible to specify information about the level of treatment offered by the safeguard and which entity the safeguard is placed on.

Activity 3.2: Frequency/likelihood

Activity 3.2 deals with the estimation of frequency/likelihood associated with an unwanted incident. In order to provide a better basis for discussing treatments to the system, we would like to document not only the final calculated frequency but also the basis for the estimated value. CORAS UML profile supports this activity through State analysis diagrams (see Chapter 5.6), which can be used as a basis for further analysis using Markov analysis or Monte Carlo simulation. As mentioned in the introduction to this requirement specification we want SecurityAssessmentUML to support calculation of frequency by including modelling support for fault tree notation into UML activity diagrams. The reason for this is that FTA is a well-known, widely used method for calculating the frequency of unwanted incidents. Figure 6.14 illustrates a simple fault tree depicting an example scenario, and Figure 6.15 illustrates how this fault

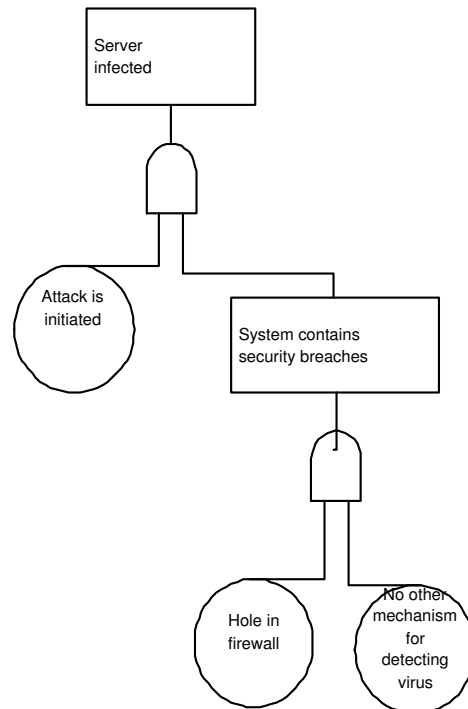


Figure 6.14: FTA notation for example scenario

tree could be modelled using existing UML activity diagrams. We believe that the traditional activity diagram may be harder to understand for non-technical people without knowledge of UML than the traditional fault tree. The complexity will increase when the size of the fault tree increases. Using FTA notation, however, it is possible to decompose a diagram into several smaller diagrams. Table 6.5 sums up existing and not supported requirements for including fault tree notation into UML activity diagrams.

Activity 3.3: Consequence

Activity 3.3 deals with assigning consequence values to unwanted incidents. Using UML1.4 this information can be attached by use of notes [9] or by using the tagged value documentation [9], but we aim for specific and uniform support. Table 6.6 presents existing and not supported requirements for specifying the consequence value of an unwanted incident in UML activity diagrams.

Activity 3.4: Estimate risk level

Assuming that SecurityAssessmentUML supports documentation of all aspects mentioned above,

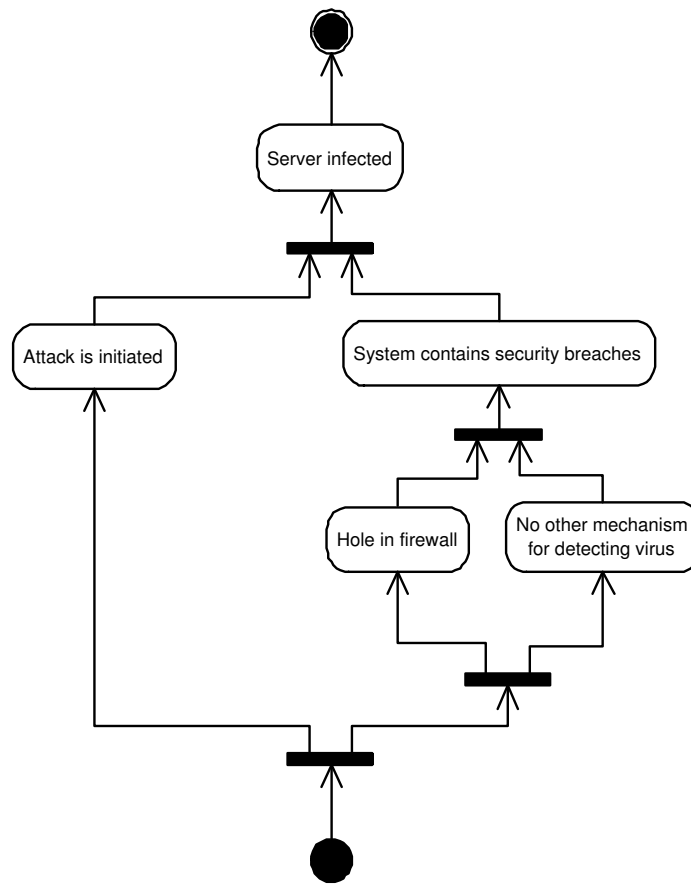


Figure 6.15: Fault tree of example scenario represented as a UML activity diagram

Table 6.5: Required modelling support for UML activity diagrams for specifying fault trees

Concept	Existing support UML	Not supported
Frequency/likelihood value	*Activity *Transition *Conditional branch	*AND-gate *OR-gate *Transfer symbols *Basic event (threats and vulnerabilities) *Top event (unwanted incident) *Particular expression power for specifying the frequency/likelihood value of an unwanted incident

Table 6.6: Required modelling support in UML activity diagrams for specifying consequence value

Concept	Existing support UML	Not supported
Consequence value		Particular expression power for specifying the consequence value of an unwanted incident.

what we need at this stage is a means for documenting the risk level value of the unwanted incident. We would like to represent estimated risk level in the same diagram as we represent

consequence and frequency values. Using UML1.4 the information can be attached using notes [9] or the tagged value documentation [9]. However, we want specific and uniform modelling support. Table 6.7 describes existing and not supported requirements for specifying the risk level of an unwanted incident in UML activity diagrams.

Table 6.7: Required modelling support in UML activity diagrams for specifying estimated risk level

Concept	Existing support UML	Not supported
Estimated risk level		Particular expression power for specifying the risk level of an unwanted incident.

Chapter 7

UML Profile for Security Assessment

This chapter presents SecurityAssessmentUML, the UML profile developed to support risk identification and risk analysis of security critical systems. The profile is specified as an extension to UML1.4 and provides support for UML sequence and activity diagrams for risk identification and UML activity diagrams for risk analysis. The profile, which is designed to be understood by non-technicians without prior knowledge of UML, is constructed based on the requirements presented in Chapter 6.5. The presentation of SecurityAssessmentUML is structured according to sub-processes.

When deciding on which UML diagrams to support, only the five types of diagrams representing behavioural information were considered. The reason for this is that security assessment focuses on behaviour aspects of the system under consideration. Collaboration diagrams are not included as they present similar information as sequence diagrams, except for the lack of timing information. Use cases are not suitable as they solely represent functionality and our intention is to specify concrete scenarios. Statechart diagrams are already supported by CORAS and thus not included here.

7.1 Risk Identification

This section presents stereotypes and diagrams used in SecurityAssessmentUML for specifying information related to risk identification of security critical systems. The table depicted in Figure 7.1 describes how the concepts identified in the requirement specification are represented in SecurityAssessmentUML, using existing UML semantics for sequence diagrams and the extension mechanisms defined by UML1.4. The four concepts are defined using stereotypes. Using these stereotypes, the example sequence diagram presented in Figure 6.11 can be specified as presented in Figure 7.2. The figure illustrates an attack of type fabrication where an outsider sends a virus to a mail-server through a mail-gateway. In this diagram, messages representing threats or unwanted incidents are distinguished from normal messages, and the misuser is distinguished from a normal actor. Furthermore, the diagram specifies that the misuser is an outsider and it also specifies which vulnerability the threat exploits.

Using the notation illustrated in Figure 7.3 it is possible to differentiate between the three threat categories interception, fabrication and modification, which was presented in Chapter 2. A notation for the fourth category, denial of service, can be specified in further work.

The table depicted in Figure 7.4 describes how the concepts identified in the requirement specification are represented in SecurityAssessmentUML, using existing activity diagrams and the





Concept	Mapping to existing UML	UML extension	Description
Vulnerability	Object	<<vulnerability>>  *{vulnerability}	This stereotype is used to distinguish objects containing vulnerabilities from normal objects. The tag vulnerability is used to specify the vulnerability. If more than one vulnerability is present, several vulnerability tags should be used (and numbered).
Unwanted incident	Message	<<unwanted incident>> 	This stereotype is used to distinguish unwanted incidents from normal messages.
Misuser	Actor instance	<<misuser>>  *{type} *{intention}	This stereotype is used to distinguish misusers from normal users (actors). The tag type refers to whether the misuser is insider or outsider. The tag intention refers to whether the misuse is intended or unintended.
Threat	Message	<<threat>> 	This stereotype is used to distinguish threats from normal and abnormal messages.

Figure 7.1: Mapping of concepts to UML sequence diagrams for risk identification

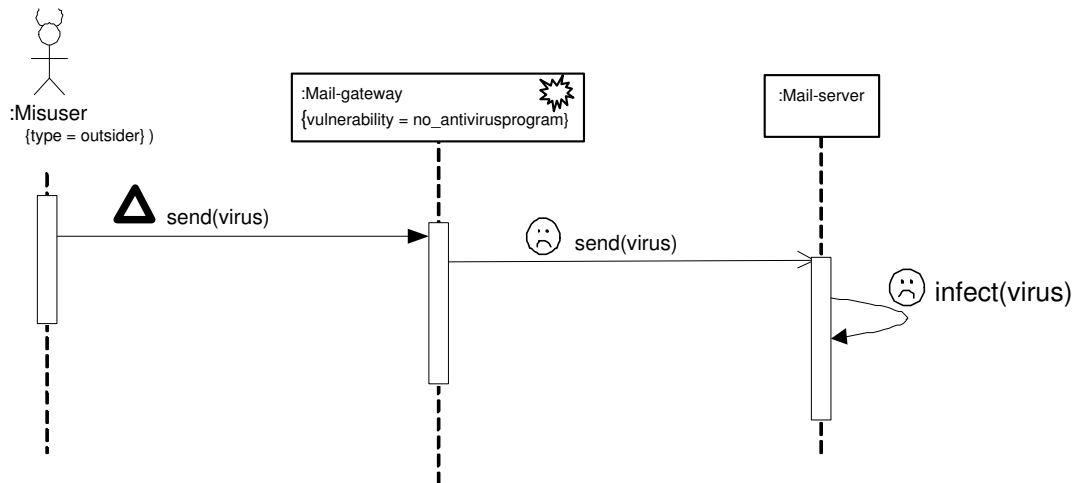


Figure 7.2: Example attack documented using extensions to sequence diagrams

extension mechanisms defined by UML1.4. The three concepts are defined using stereotypes. Using these stereotypes, the example activity diagram presented in Figure 6.12 can be presented as illustrated in Figure 7.5. The diagram specifies that the threat may exploit a vulnerability of the mail-gateway (no anti virus program installed) leading to unwanted incidents. In this diagram, activities posing a threat to the system and activities representing unwanted incidents are distinguished from activities representing normal system behaviour. Furthermore, the diagram indicates that the guard condition of the branch is a check for whether or not a vulnerability exists.

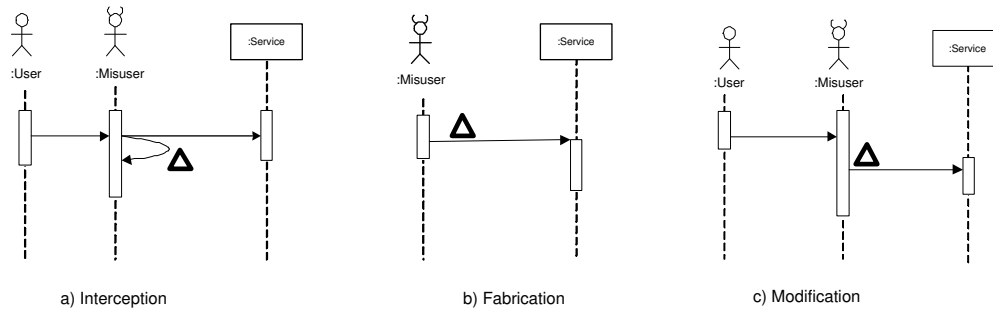


Figure 7.3: Categories of threats illustrated using extensions to sequence diagrams




Concept	Mapping to existing UML	UML extension	Description
Vulnerability	Guard	<<vulnerability>>  *{asset} *{vulnerability}	This stereotype is used to mark that a guard connected to a branch is a test for whether or not a vulnerability exists. The tag asset is used to represent the entity containing the vulnerability. The tag vulnerability can be used to further specify the vulnerability.
Unwanted incident	Activity state	<<unwanted incident>> 	This stereotype is used to distinguish activity states involving unwanted incidents from normal activity states
Threat	Activity state	<<threat>> 	This stereotype is used to distinguish activity states introducing threats to the system from normal activity states.

Figure 7.4: Mapping of concepts to UML activity diagrams for risk identification

7.2 Risk Analysis

7.2.1 Determine existing controls

This section specifies a stereotype for safeguards and presents an example of use of this stereotype. The stereotype and its associated tags are presented in Figure 7.6. In order to increase the readability of the diagram, the stereotype is represented by an icon illustrating a padlock. Using SecurityAssessmentUML, the example activity diagram presented in Figure 6.13 can be presented as illustrated in Figure 7.7. Compared to the diagram modelled using traditional activity diagrams as defined by UML1.4, this diagram specifies that the guard condition of the branch is a check for whether or not a specific safeguard is present. Furthermore the diagram specifies the level of protection offered by the safeguard (reduce likelihood) and which asset the safeguard is placed on (mail-gateway). In the example scenario there is a chance that the virus is undetected despite the presence of an anti virus program. Thus, as illustrated in the figure there is a chance that the message will exploit a vulnerability in the anti virus software. If this happens, the virus will be forwarded to the mail-server.

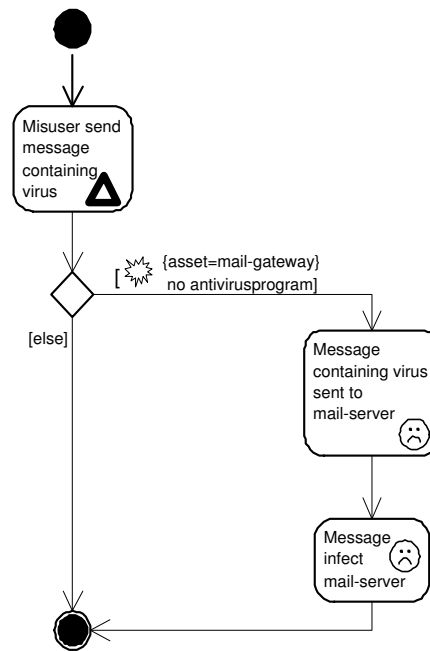


Figure 7.5: Example attack documented using extension to activity diagrams


Concept	Mapping to existing UML	UML extension	Description
Safeguard	Guard	<< safeguard >>  *{name} *{asset} *{level}	This stereotype is used to mark that a guard connected to a branch is a test for whether or not a safeguard exists. The three tags are used to specify the name of the safeguard, which entity the safeguard is placed on and which level of treatment the safeguard offers.

Figure 7.6: Mapping of the concept safeguard to UML activity diagram semantic

7.2.2 Determine consequence and likelihood of unwanted incidents

This section specifies stereotypes for including basic fault tree notation into UML activity diagrams, as well as tagged values for documenting values for consequence, frequency/likelihood and risk level. Stereotypes for representing a sub set of the fault tree concepts are presented in Figure 7.9. The remaining concepts used in FTA notation could be included in further work. Values for frequency/likelihood, consequence and risk level can be expressed in a uniform way using the tags presented in Table 7.1. The tag frequency/likelihood are attached to all activity states, while consequence value and risk level only are attached to the unwanted incident (the top event). Rules for estimating risk level using a combination of qualitative and quantitative values for likelihood/frequency and consequence are not considered in this report.

Figure 7.9 specifies the fault tree presented in Figure 6.14 and 6.15 using SecurityAssessmentUML. The figure shows that the unwanted incident "Server infected" will occur if there are holes in the firewall and no other security mechanism for detecting viruses are included given

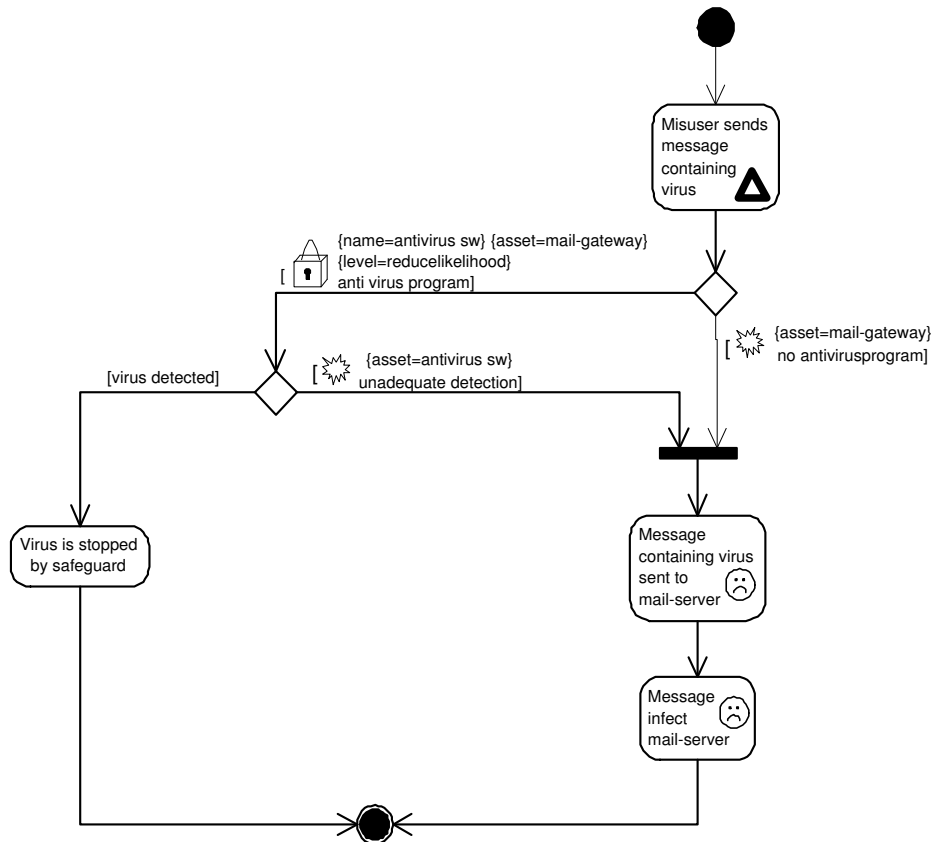


Figure 7.7: Example attack documented using extension to activity diagrams

Concept	Mapping to existing UML	UML extension	Description
AND-gate	Fork and join		This stereotype is used to substitute use of fork and join with an AND symbol in order to simplify the notation.
OR-gate	Branch		This stereotype is used to substitute use of branch with an OR symbol in order to simplify the notation.
Transfer down	Activity state		This stereotype is used to distinguish the top event in a decomposed tree from a "normal" top event.
Transfer up	Activity state		This stereotype is used to distinguish an event that is further decomposed from normal activity states.
Basic event	Activity state		This stereotype is used to distinguish basic events from normal activity states.

Figure 7.8: Mapping of concepts to UML activity diagrams for use of fault tree notation

that an attack is initiated. In contrast to ordinary activity diagrams as defined by UML1.4, our profile allows more than one transition from the initial state. In UML1.4 forks are used to

Table 7.1: Mapping of concepts to UML activity diagrams for risk analysis

Concept	Mapping to existing UML	UML extension	Description
Frequency/likelihood	Tagged value on activity state	{frequency = value} {likelihood = value}	This tagged value is used to specify the frequency/likelihood of an event. Frequency is used when quantitative values are available. In other cases likelihood is used.
Consequence	Tagged value on activity state	{consequence = value}	This tagged value is used to specify the consequence value of an unwanted incident.
Risk level	Tagged value on activity state	{risk level = value}	This tagged value is used to specify the risk level estimate of an unwanted incident.

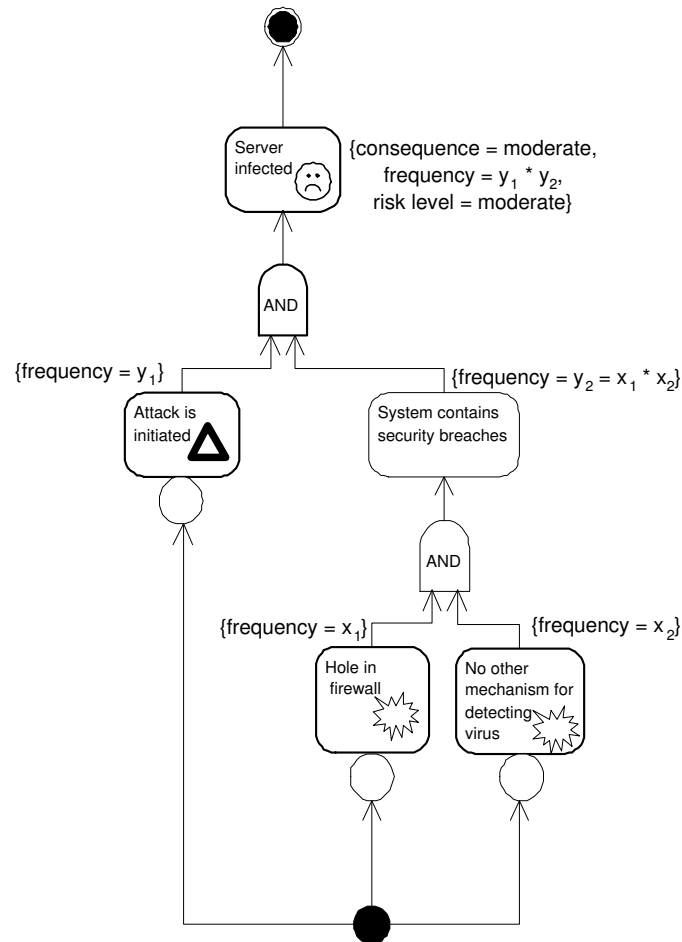


Figure 7.9: Example fault tree specified using extensions to activity diagrams

denote parallel transitions. Since the UML concepts fork and join is respecified in our profile, we do not use them in these diagrams. Furthermore, we believe use of fork and join symbols

might confuse stakeholders inexperienced with UML.

As illustrated in the figure all activities representing basic events are specified as either threats or vulnerabilities. We believe it is useful to distinguish between these concepts. The rationale for this is that the fault tree might be a useful means for identifying cost-effective treatment options. The higher up in the tree a treatment is placed, the more vulnerabilities it protects against. Since the threat will always exist, the main point during this process is to ensure that vulnerabilities are removed. Activities representing top events are specified using the stereotype for unwanted incidents.

It can be discussed whether all intermediate activities should have been classified as threats, vulnerabilities or unwanted incidents and specified using the appropriate stereotype. This notation is not used in this report. The reason for this is that the classification should be implicit from the combination of lower level activities. This guideline is contradictory to the use of the unwanted incident stereotype for the extended sequence and activity diagrams supporting risk identification. In these diagrams all events or activity states representing undesirable behaviour is specified by the stereotype.

As an alternative for drawing fault trees, values for frequency/likelihood, consequence and risk level could be attached to our extended sequence and activity diagrams. Examples are provided in Figure 7.10 and Figure 7.11 for sequence and activity diagrams, respectively. However, although supporting specification of risk analysis results, these diagrams do not support calculation of frequency. Hence, for calculation purposes we believe that the extended activity diagrams inspired by basic fault tree notation is preferable.

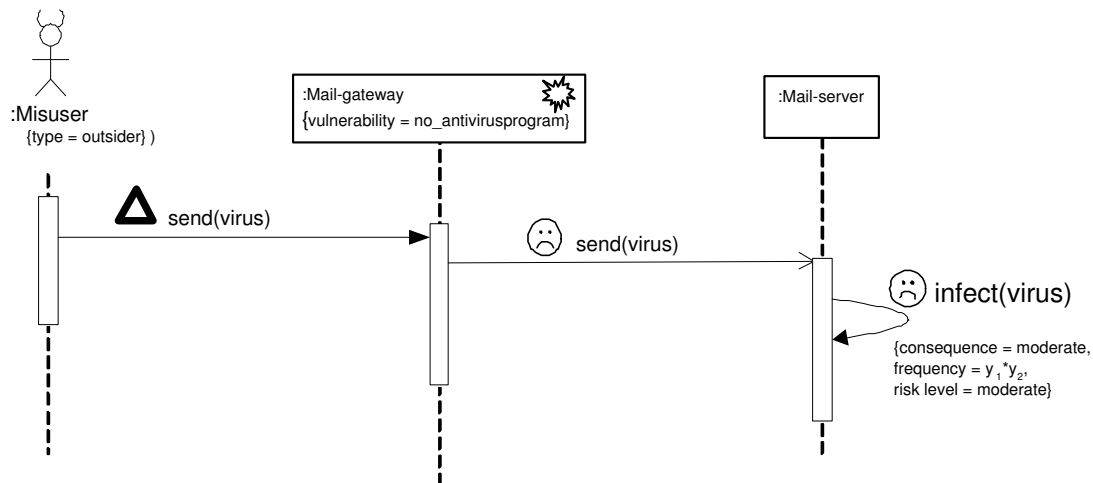


Figure 7.10: Sequence diagram tagged with output from risk analysis

7.3 Discussion

Although all stereotypes defined by SecurityAssessmentUML are defined by both text strings in brackets and by icons, all diagrams presented in this report are drawn using the iconic representation. The rationale for this is that we believe non-technical users will find these diagrams more understandable. Except for the icon representing vulnerability, all icons used are inspired by well-known concepts. The threat icon is inspired by a warning sign, familiar

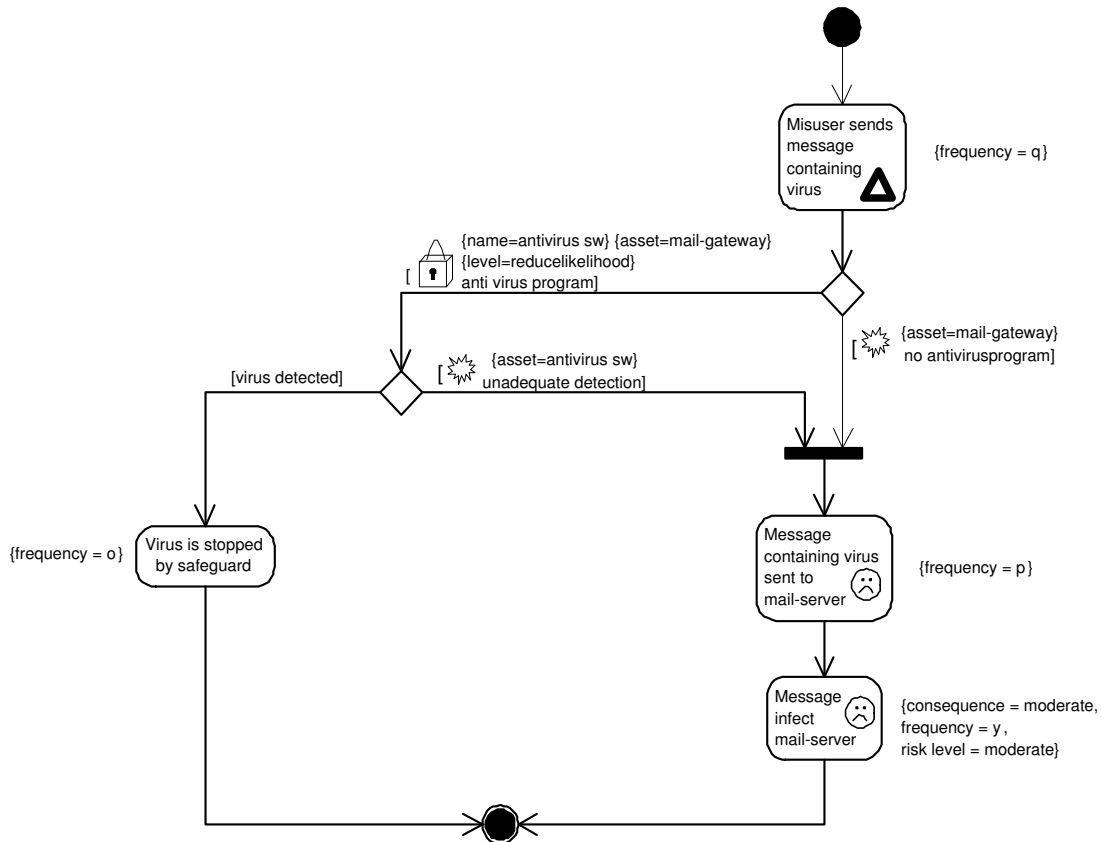


Figure 7.11: Activity diagram tagged with output from risk analysis

to all car drivers. The icon used to represent misusers is inspired by the devil, and the icon used to represent safeguards is inspired by a padlock. The icon representing unwanted incidents is inspired by a sad smiley, used to demonstrate that unwanted incidents represent situations that can make users unsatisfied. If read by designers or system developers, we believe that a representation using stereotypes in brackets would be more appropriate. A tool for drawing diagrams should support the translation between the different notations.

The example diagrams illustrate that use of stereotypes is more restrictive in the activity diagrams used for determining the likelihood/frequency of unwanted incidents than in the extended sequence and activity diagrams specifying output from risk identification. In the fault-tree inspired activity diagrams only the top-level, undesirable activity state is represented as an unwanted incident. The reason for this is that all activity states in these diagrams represent undesirable behaviour. This is not necessarily the case in the diagrams used to document results from risk identification, which may contain normal system behaviour as well.

SecurityAssessmentUML defines several tags. These can be used to add additional information to the diagrams in a uniform way. Normally, diagrams are tagged with all information available.

Chapter 8

Evaluation process

This chapter discusses aspects related to the choice of evaluation criteria, as well as methods for evaluation. Evaluation criteria for the UML profile is presented in Section 8.1. Choice of evaluation strategy is presented in Section 8.2, while a brief description of how the expert judgements were accomplished is included in Section 8.3.

8.1 Evaluation criteria for the UML profile

This section presents the seven evaluation criteria used to evaluate the UML profile for security assessment. Following the list of criteria, the rationales for choice of criteria are discussed.

1. **The profile must be able to represent all concepts in the domain**

As stated in the problem description the main objective of this thesis is to specify a profile that supports documentation of all required output from risk identification and risk analysis. As a minimum this corresponds to the nine concerns belonging to sub-process 2 and 3 in the security documentation framework (see Figure 6.8). However, as stated in the requirements specification we also want to support documentation of important relations between these concerns. Thus, we want the complete ontology for risk identification and risk analysis as presented in Figure 6.9 to be covered.

This criterion focus on avoiding construct deficit¹.

2. **The profile must only be able to express thing that are in the domain**

According to Wand and Weber [57] a model including constructs that do not represent any ontological concepts may undermine the ontological clarity of the constructs and the languages. The focus of this criterion is to avoid this situation, termed construct excess².

3. **An ontological concept must be represented by the same modelling construct throughout the profile**

The criterion is included as use of different modelling constructs for the same ontological concept might confuse the readers. The criterion corresponds to avoidance of construct redundancy³.

¹Construct deficit: An ontological concept is not represented by any modelling construct [57].

²Construct excess: A modelling construct does not represent any ontological concept [57].

³Construct redundancy: Several (overlapping) modelling constructs represent the same ontological concept [57].

4. **A modelling construct must represent the same ontological concept throughout the profile**

This criterion is included as use of the same construct for different phenomena or concepts will tend to make the language confusing [33]. The criterion corresponds to avoidance of construct overload⁴.

5. **Symbol discrimination must be easy**

This means that it should be easy to difference between the various symbols of the language. This is especially important since we assume that diagrams shown to non-technical stakeholders will use iconic representation for stereotypes.

6. **The use of emphasis in the notation must be in accordance with the relative importance of the statements in the given model**

7. **The symbols used in the report should strive for symbolic simplicity**

In this thesis, the term symbolic simplicity is used to denote intuitiveness of symbols used, as well as effort required to draw the symbols using pen and paper.

All criteria are taken from an evaluation framework used to consider the potential of a modelling approach to support the creation of high quality models. The framework, which is presented in [33] considers five quality areas for languages.

The criteria presented above cover aspects related to the two quality areas *domain appropriateness* [33] and *comprehensibility appropriateness* [33]. The former relates to the correspondence between the model and the modelling domain, the latter relates to the correspondence between the model and the audience interpretation of it. Requirement 1 and 2 constitute domain appropriateness, while requirement 3 to 7 is used to evaluate comprehensibility appropriateness. In particular, requirement 1 to 4 deals with avoiding the four ontological discrepancies, which according to Wand & Weber [57] may undermine the ontological clarity of modelling constructs and languages. The remaining criteria focus more on usability aspects of the profile.

Although not treated here, three other areas for language quality are defined by the framework. *Participant knowledge appropriateness* [33] and *knowledge externalizability appropriateness* [33] are not evaluated in this report as these criteria are highly dependent on the stakeholder. *Technical actor interpretation* [33] relates to whether the language lends itself to automatic reasoning for technical actors involved and is not evaluated because the profile is primarily a contribution to research on means for enhancing communication among non-technical stakeholders. For more information on the underlying evaluation framework, the reader is referred to [33].

8.2 Evaluation strategy

This section describes how the evaluation was performed and reflects the choice of evaluation strategy. When deciding upon suitable strategies, the eight strategies presented by McGrath were considered. These are laboratory experiment, experimental simulation, field experiments, field studies, computer simulations, formal theory, sample survey and judgement studies. Figure 8.1, which provides an overview of these methods, illustrates that the choice of evaluation method influences the three desired features of research [38]; A) Generalizability of the evidence, B) Precision of measurement, and C) Realism of context.

⁴Construct overload: A modelling construct corresponds to several ontological concepts [57].

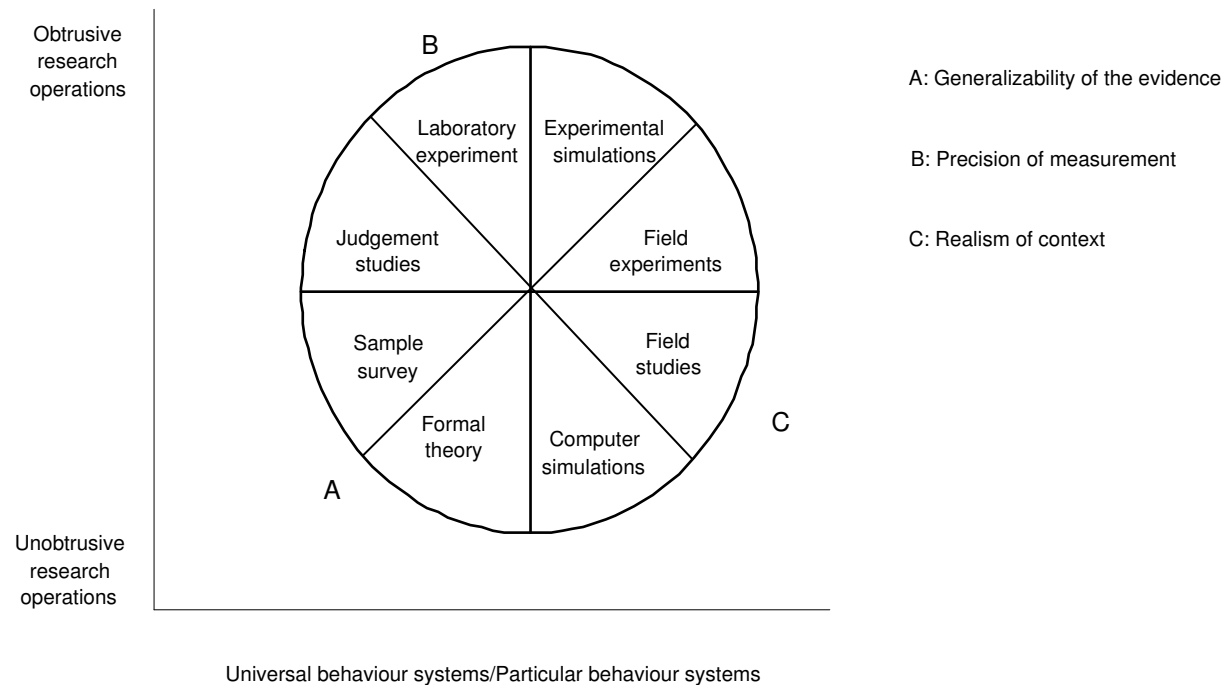


Figure 8.1: Research strategies [38]

Three different approaches were chosen; 1) simplified version of formal theory, 2) simplified version of field studies, and 3) judgement studies. As illustrated in the figure, use of a combination of these approaches should allow for both generalization (A), precision (B) and realism (C).

Formal theory [38] is used for examining criteria 1 to 4, which deals with ontological discrepancies. The evaluation will be performed using formal mapping between the ontology and modelling constructs. Although formal theory often is used to describe more formal, mathematical proofs, we classify this evaluation as a simplified version of formal theory as it is performed as a theoretical argumentation for compliance with the criteria.

Trials are used to examine whether SecurityAssessmentUML supports specification of example output results from risk assessments. Example results are tried documented using the extensions defined in the profile. We find that trials are a necessary addition to mapping between concepts, as it might reveal potential concepts or relationships not included in the ontology. Trials can be categorised as a simplified version of *field study* [38] because they are used to observe how SecurityAssessmentUML can be used to document existing risk assessment results.

Expert judgements [38] are used to evaluate usability aspects of the profile, as well as to get an indication of the understanding of the diagrams. The accomplishment of expert judgement is presented in Section 8.3.

The remaining five strategies, sample surveys, laboratory experiments, experimental simulations, field experiments and computer simulations are considered out of scope for this thesis. Some however, could be used in further evaluation. In the following, we discuss why these approaches were rejected.

Sample surveys [38] are highly related to judgement studies. Judgement studies were used in this thesis because these studies require a smaller set of participants and because these studies are believed to provide us with more extensive information than we could get in a sample survey.

Field experiments [38], *laboratory experiments* [38] and *experimental simulations* [38] could have provided valuable input about the suitability of SecurityAssessmentUML as a communication medium. However, an evaluation of the suitability of SecurityAssessmentUML as a communication medium was not part of the thesis and should be investigated in further work.

Computer simulations [38] are not suitable for our purpose and thus not further discussed in this thesis.

8.3 Accomplishment of expert judgements

The judgement studies were conducted by participants chosen in consultation with my supervisor. Ideally, the participants in the study should have been non-technical people possessing knowledge about security concepts. Knowledge about security concepts is important as we assume that the non-technical people supposed to read the final diagrams work in a setting where they understand the security concepts used in risk assessments. Due to lack of people from the target group, the evaluation form was sent to technical people with minor knowledge of UML.

Each participant were asked to answer questions about the three types of diagrams supported by SecurityAssessmentUML. For each diagram, the participants were asked to identify and describe the security elements presented and to explain the complete scenario. Furthermore, the judgement included questions about the intuitiveness and the discriminability of symbols used. In addition to this, the evaluation form included a short introduction to the profile as well as a description of normal system behaviour. The evaluation form is included in Appendix C.

The evaluation was performed by sending the evaluation form directly to the participants by e-mail. The evaluation form was sent to six persons, but only two returned the schema.

Chapter 9

Evaluation

This chapter presents an evaluation of SecurityAssessmentUML. Results from the expert judgement is included in Appendix D and discussed at appropriate places in this chapter. The mapping between concepts in the ontology and modelling constructs are presented in Section 9.1 and the trials are elaborated in Section 9.2. Fulfillment of the evaluation criteria are discussed in Section 9.3.

9.1 Mapping schema between ontology and modelling constructs

This section presents the mapping between concepts in the assessment ontology and modelling constructs specified in SecurityAssessmentUML. The mapping, which is presented in Figure 9.1, is used to evaluate criteria 1 to 4. The first column lists all concepts in the assessment ontology presented in Figure 6.9. The second column lists all modelling constructs presented in SecurityAssessmentUML. For each modelling construct, information on defined tags is included. Ontological concepts and modelling constructs presented on the same row map.

9.2 Trial

This section presents results from a trial examining the usefulness of SecurityAssessmentUML as a means to document results from risk assessment. The trial is conducted by specifying information provided by an example result using SecurityAssessmentUML. The analysis object used for evaluation is presented in Section 9.2.1. Section 9.2.2 and 9.2.3 present diagrams documenting results from risk identification and risk analysis, respectively.

For each type of diagram we consider two aspects. Firstly, we examine whether the diagram supports specification of all information provided by the example results. Secondly, we compare the final representation format against two alternative representations of results from security assessment; 1) the traditional documentation of risk assessment result, i.e. FMEA tables, fault trees, standard texts etc., and 2) the CORAS UML profile. This evaluation is performed with the target group of users in mind.











Concept in ontology	Representation in SecurityAssessmentUML	Comment
Undesirable behaviour	No specific counterpart in SecurityAssessmentUML, but supported through its three specialisations unwanted incidents, vulnerabilities and threats	
Unwanted incident	<<unwanted incident>>  *{asset}	The belonging tag asset specifies the asset containing the vulnerability.
Vulnerability	<< vulnerability>> 	
Threat	<<threat>> 	According to the ontology a security attack (threat) consists of misuse (action) performed by a misuser.
Misuser (insider/outsider) Misuse (intended/unintended)	<<misuser>>  *{type} *{intention}	The tag type is used to specify whether the misuser is external or internal to the organisation. The tag intention is used to specify whether the misuse is performed intentionally or unintentionally.
Risk	No specific counterpart in SecurityAssessmentUML, but covered by the concepts unwanted incident, consequence value and frequency/likelihood value	According to the ontology risk consists of an unwanted incident, consequence value and frequency/likelihood value.
Consequence	Tagged value	
Likelihood/frequency	Tagged value	
Risk level	Tagged value	
Safeguard	<< safeguard>>  *{name} *{asset} *{level}	The tag name specifies the name of the safeguard. The belonging tag asset specifies which entity the safeguard is protecting. The belonging tag level represents the level of protection offered by the safeguard (not shown in ontology)
Asset value	No specific counterpart in SecurityAssesmentUML	
Asset	No specific counterpart in SecurityAssessmentUML, but partly covered by the tagged value asset used in conjunction with safeguard and vulnerability.	
No specific counterpart in the ontology	<<AND>> 	
	<<OR>> 	
	<<basic event>> 	
	<<transfer up>> 	
	<<transfer down>> 	

Figure 9.1: Mapping between concepts in ontology and UML extensions defined by SecurityAssessmentUML

9.2.1 Description of analysis object

The example used for evaluating the profile is taken from a risk assessment of a mobile e-commerce system focusing on the purchasing process. The complete system including risk assessment results is presented in [20]. Due to time limitations only one example result is considered. Hence, trials on more specialized and detailed results should be subject for further work. Here, the scenario used as a basis for the risk identification trial is briefly described. A description of the risk assessment result used as a basis for the risk analysis trial is not included in the report, as the original fault trees should be clear from the diagrams.

The example risk assessment result specifies possible threats against a successful transmission of electronic service delivery from supplier to consumer after a successful purchase. Normal system behaviour is illustrated in Figure 9.2. As indicated in the figure, the electronic service is successfully transferred from supplier to consumer via the communication channel without any form of interference. The parameter `download_serviceID` represents the electronic service. The result of the risk assessment considering threats of type modification is included in Appendix E.

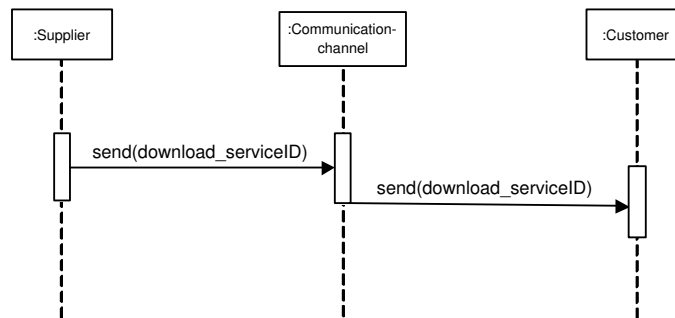


Figure 9.2: Normal system behaviour

9.2.2 Risk identification

In this section we discuss SecurityAssessmentUML as a means for documenting output from sub-process 2, risk identification. The discussion is based on the sequence diagram presented in Figure 9.3 and the activity diagram presented in Figure 9.4. In the activity diagram vertical solid lines are used to partition the activity states on the diagram into groups called swimlanes [9], where each group represents the responsible for actions within that group. The diagrams, which visualize the information provided by the FMEA schema in Appendix E, show an attack where a misuser modifies the message sent from the supplier to the consumer via the communication channel. Furthermore, both diagrams specifies that the misuser exploits the two vulnerabilities present in the communication channel: unauthorised access possible and insufficient encryption. Although not present in the FMEA table, the activity diagram also depicts that use of the safeguard encryption may prevent the attack from occurring.

In terms of the activity diagram, all information except from the classification of misuser into outsider is included. The reason for this is that activity diagrams do not focus on actors. In terms of the sequence diagram, the final system effects are not included. The reason for this is that information on how the final service is collected is not presented by [20]. However, given

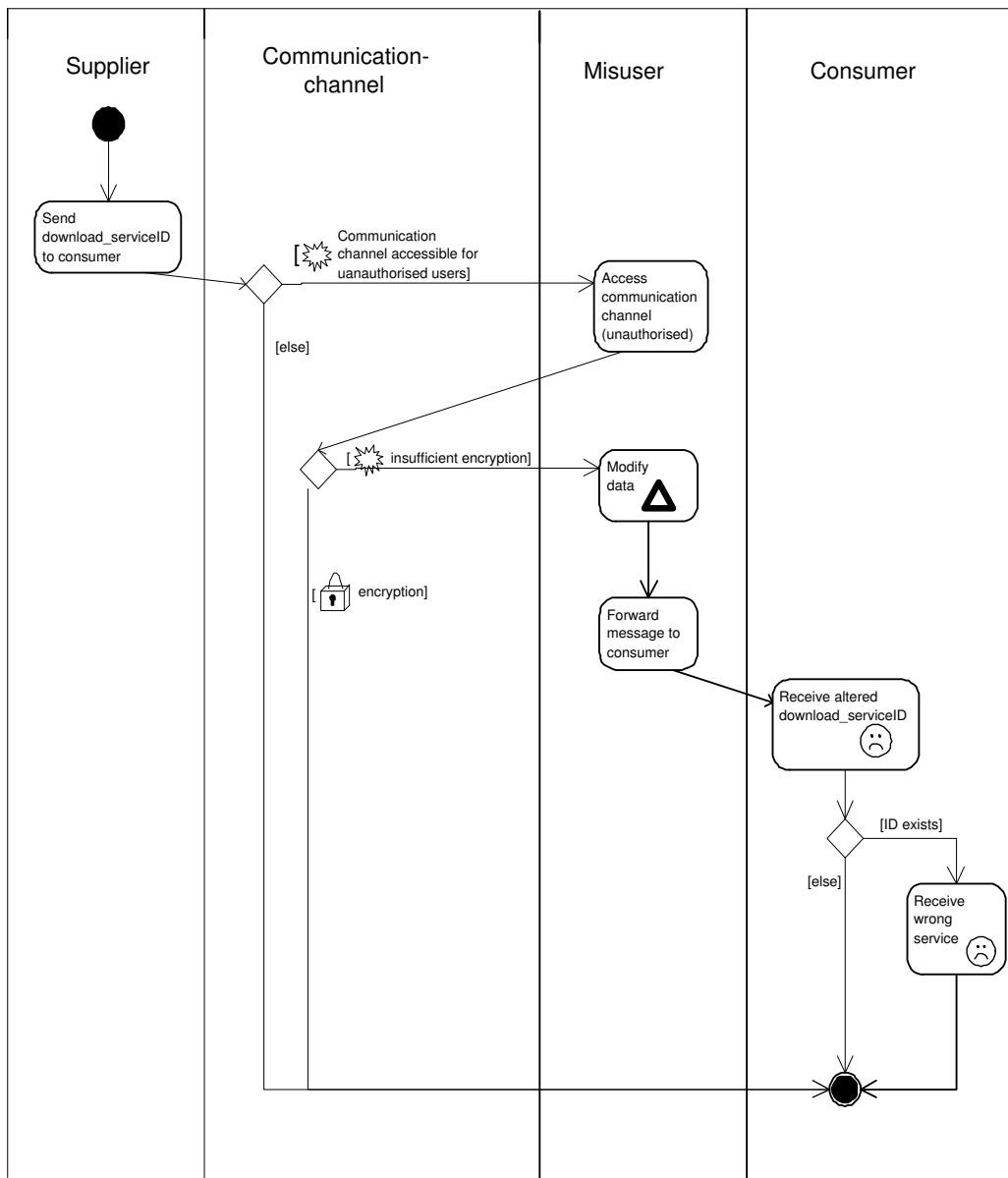


Figure 9.3: Manipulation-attack specified using extensions to activity diagrams

a description, the sequence diagram could have been extended to include this information. The column Remarks is not considered when documenting the results.

Compared to diagrams made using CORAS UML profile the diagrams made using SecurityAssessmentUML provide more detailed information. In particular, they specify how the attack actually occurs by providing an overview of the sequence of events/activities involved. Compared to the FMEA table, SecurityAssessmentUML offers a means for visualizing the information highlighting the connection between threats, vulnerabilities and unwanted incidents. The activity diagram illustrates activities involved in an attack, while the sequence diagram depicts the sequence of actions involved in the attack.

The results from the judgement study (see D.1 and D.2) indicate that the participants found

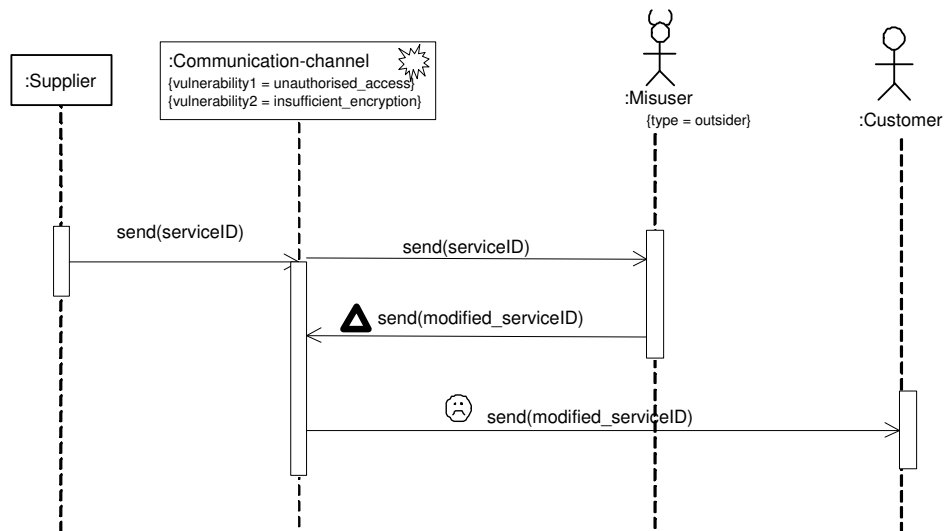


Figure 9.4: Manipulation-attack specified using extensions to sequence diagrams

the diagrams understandable. In terms of the sequence diagram, however, the participants had some problems describing the misuser. The reason for this is that type of misuser was not specified in the original diagram presented for the experts. This was corrected in a later version, allowing classification of the misuser according to the two categories type (insider/outsider) and intention (intentionally/unintentionally). Furthermore, one of the participants asked for more detailed information about why the communication channel is accessible. This information is not provided in the diagram because this information is not specified in the risk assessment result the diagrams are based on. If more detailed information had been provided by the FMEA scheme, it could have been specified in the diagram. Except for this, all security elements were identified and the scenario correctly described. In terms of the activity diagram, one of the participants only identified one of the two vulnerabilities in the system. Except for that, both participants seemed to understand the diagrams.

Although the result of the judgement study indicates that diagrams are understandable, no valid conclusions can be drawn. The main reason for this is that only two participants returned the schema.

9.2.3 Risk analysis

In this section, we discuss SecurityAssessmentUML as a means for documenting output from sub-process 3, risk analysis. The discussion solely focuses on the extended activity diagrams used for analysing frequency and specifying estimates for frequency, consequence and risk level. Hence, safeguards are not discussed here.

As a basis for the discussion we examine the three diagrams presented in Figure 9.5 to 9.7. Original fault trees used as a basis for drawing the diagrams can be found on page 279, 288 and 289 in [19]. For information about details in the fault trees the reader is referred to [20].

The diagrams specify how the unwanted incident "Fraud against consumer" might occur. The two generic fault trees, numbered 1,6 and 1,0,1, deal with issues related to more than one intermediate event. Example frequency and consequence values are attached to the figure to illustrate how frequencies are calculated. In the example quantitative values for frequency

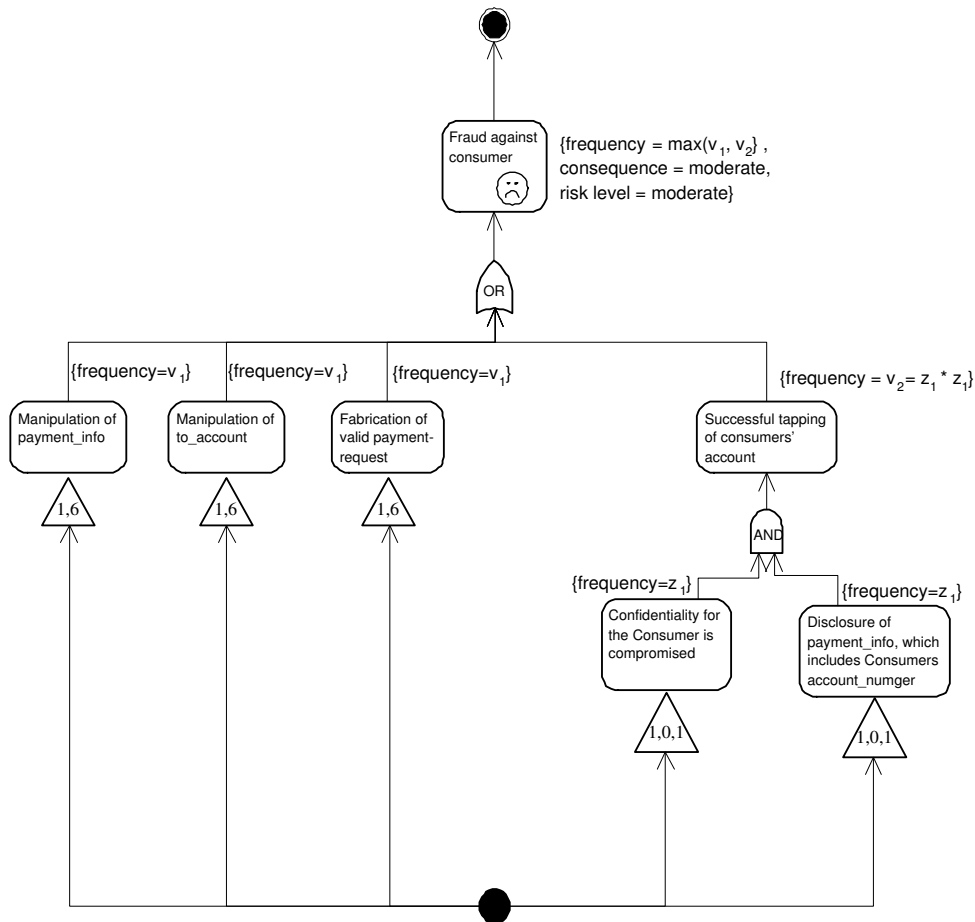


Figure 9.5: SecurityAssessmentUML notation for fault tree “Fraud against consumer”. Adapted from [19]

are used. Using UML1.4 this information could have been documented using normal activity diagrams with frequency values attached as notes. We believe that the notation inspired by fault trees is easier to understand for people without prior knowledge of UML than a normal activity diagram representing the same information using fork and join and branch. However, this must be tested on target users. Another difference from UML1.4 is that UML1.4 does not distinguish between unwanted incidents and threats and vulnerabilities.

CORAS UML profile supports documentation of frequency and consequence values in State Analysis diagrams. As discussed in Chapter 5.7, these diagrams may be hard to specify for complex systems as they require a complete list of normal and bad states, as well as on possible transitions between the states. One advantage by using fault trees is that the extended activity diagram using notation from fault trees is appropriate for use when considering treatments to the system. The higher up in the tree a safeguard is placed, the more threats/vulnerabilities it protects against. Thus, fault tree notation can be useful for cost-benefit analysis as it provides a means for comparing costs and benefits between treatment options.

It can be discussed whether traditional fault tree notation should be used instead of adapting the notation into UML diagrams. The reason the notation is adapted into SecurityAssessmentUML is that we want all diagrams to be supported by one modelling language.

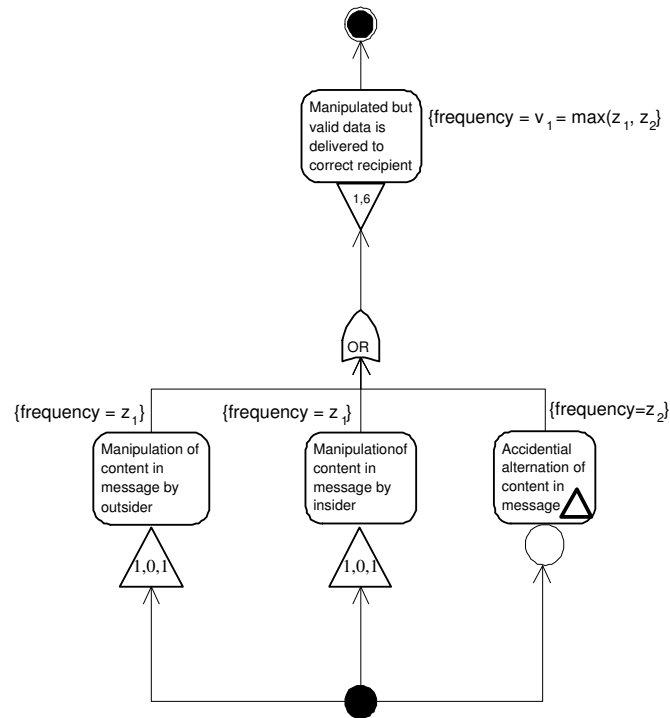


Figure 9.6: SecurityAssessmentUML notation for generic fault tree 1,6. Adapted from [19]

For the judgement study, the diagram presented in Figure 7.9 was used. The results of this study suggest that the participants understood how to read the diagram (see D.3). However, feedback from an expert of UML and fault tree analysis indicates that the diagram might be hard to understand for people familiar with UML activity diagram notation or fault tree notation. The reason for this is that the notation used in this diagram is a hybrid between the two notations. Thus, the diagram does not follow the syntax they expect. The main problem is to choose transition from the initial state. In traditional UML activity diagrams, fork, join and branches are used to indicate this. In our case, we want all transitions to be followed in parallel. In order to avoid confusion among UML experts, reader should be provided by an explanation on how to read and interpret the diagrams.

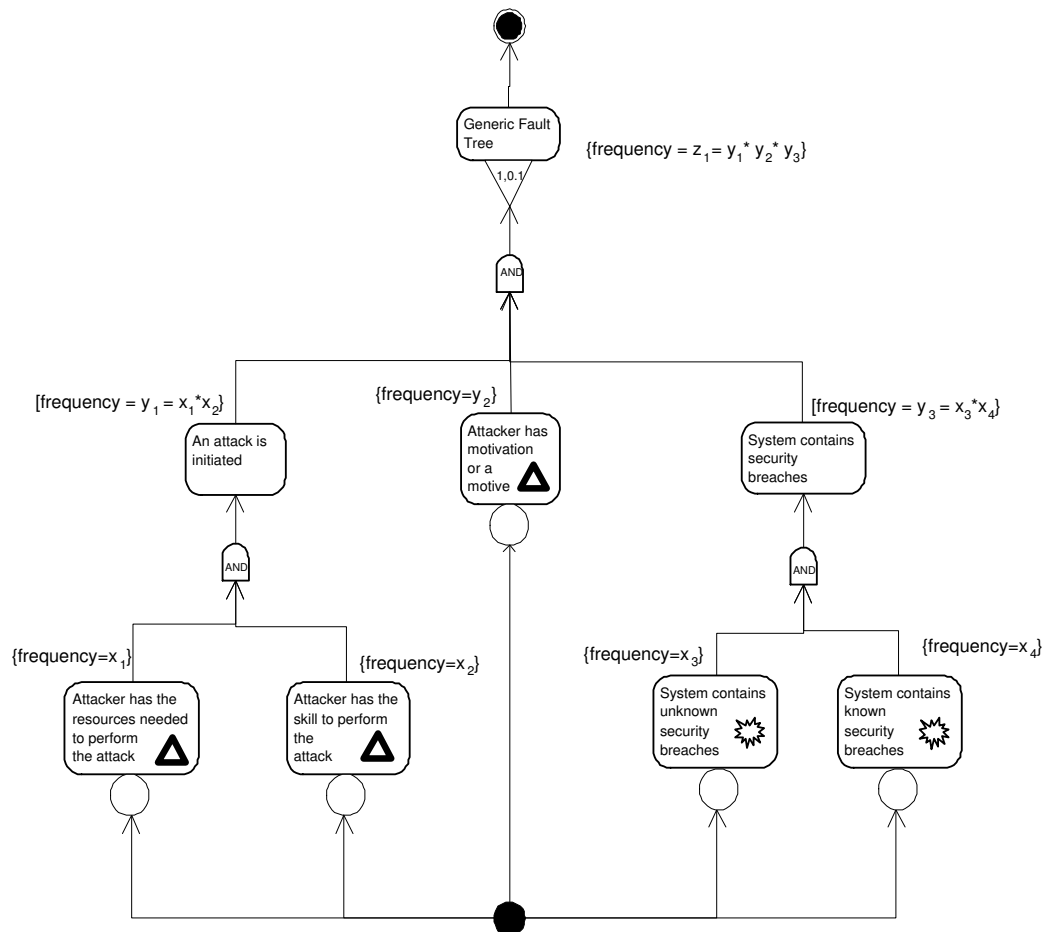


Figure 9.7: SecurityAssessmentUML notation for generic fault tree 1,0,1. Adapted from [19]

9.3 Evaluation criteria

In this section we discuss the seven evaluation criteria presented in Chapter 8.1. The criteria can be classified into two groups; 1) criteria considering formal aspects of the profile and 2) criteria considering usability aspects of the profile. Criteria 1 to 4, which deals with ontological discrepancies, belongs to the former group. To examine these criteria we examine the mapping between ontological concepts included in risk identification and risk analysis and modelling concepts specified in SecurityAssessmentUML. The mapping, which is included in Figure 9.1, considers all concepts present in the ontology for risk identification and risk analysis as presented in Figure 6.9. As described earlier, the figure includes more concepts than defined by the security documentation framework for these sub-processes.

When considering ontological discrepancies we only consider extensions defined by SecurityAssessmentUML. An evaluation including aspects inherited from UML1.4 is beyond the scope of this thesis. For an evaluation of UML1.4, the reader is referred to [41] or [33]. When evaluating criteria 5 and 7 we consider the results of the expert judgements.

Evaluation criteria 1: The profile must be able to represent all concepts in the domain

The objective of this criterion was to establish whether construct deficit is avoided in SecurityAssessmentUML. The criterion is fulfilled if all concepts in the ontology (left column of Figure 9.1) are represented by a modelling construct in the right column. The comparison identified four deficits in SecurityAssessmentUML. These are undesirable behaviour, risk, asset and asset value. None of these concepts are included in the list covering concerns that at a minimum must be supported by SecurityAssessmentUML. Despite of this, each of the discrepancies are briefly discussed below.

The concept risk is indirectly supported through representation of the modelling construct unwanted incident and its associated values for consequence and frequency/likelihood. Thus, we do not regard the lack of an explicit notation for risk problematic. The concept undesirable behaviour is supported through its three specialisations; unwanted incident, vulnerability and threat. As undesirable behaviour is an abstract term, we do not regard the lack of an explicit notation for this problematic. The concept asset should be covered by a UML profile supporting documentation of output from sub-process 1. However, the concept is included in the ontology to indicate which assets safeguard and vulnerabilities belongs to. SecurityAssessmentUML support specification of this through the tagged value asset.

The fourth deficit identified is lack of modelling support for specifying how the consequence of an unwanted incident affects the value of assets. This information provides additional information compared to the general consequence value of the unwanted incident, and it is thus important when communicating the seriousness of unwanted incidents to stakeholders. Thus, the lack of support for indicating the effect on the value of assets is regarded problematic and should be resolved in further work. CORAS UML profile support specification of asset value, but does not support specification of how the predefined value is affected by a risk.

Evaluation criteria 2: The profile must only be able to express things that are in the domain

The objective of this evaluation criterion is to examine whether construct excess is avoided. This discrepancy is only problematic if the construct is clearly intended to represent phenomena in the problem domain [41]. Thus, we looked at two types of excesses in SecurityAssessmentUML [41]; 1) modelling constructs which are intended for representing the security domain but have no counterparts in the ontology, and 2) modelling constructs that are used to represent non-domain oriented information and have no counterparts in the ontology. The mapping illustrated in Figure 9.1 identifies six SecurityAssessmentUML constructs that have no counterparts in the ontology. Five of these belong to the latter group as they are used to include fault tree notation into our extended activity diagrams supporting risk analysis. This type of construct excess is not problematic and thus, the inclusion of symbols to support fault tree notation does not decrease the language quality of the profile. The sixth construct is the tag used to represent the level of protection of a safeguard. This attribute of the safeguard could have been part of the ontology and thus, we do not regard it as a threat towards the quality of the model.

Evaluation criteria 3: An ontological concept must be represented by the same modelling construct throughout the profile

The objective of this evaluation criterion is to examine whether construct redundancy is avoided. This criterion is fulfilled by SecurityAssessmentUML.

Evaluation criteria 4: One modelling construct must represent the same ontological concept throughout the profile

The objective of this evaluation criterion is to examine whether construct overload is avoided. The mapping illustrated in Figure 9.1 indicates that all modelling constructs present in SecurityAssessmentUML corresponds to maximum one ontological concept. Hence, this evaluation criterion is fulfilled.

Ideally, the evaluation should have included a comparison with all modelling constructs present in UML1.4, in order to ensure that none of the constructs defined were already defined. There was no time for such an evaluation.

Evaluation criteria 5: Symbol discrimination must be easy

This evaluation criterion was examined by comparing the extensions defined in SecurityAssessmentUML with each other and by interpreting the results from the expert judgements. Except for the two symbols for transfer up and transfer down, we believe that discrimination should be very easy as no symbols are similar. Transfer up and transfer down will however, be clear from the context. Top level activity indicates transfer down while lower level activity indicates transfer up. Furthermore, the participants in the judgement study claimed that symbol discrimination was easy (see Table D.4).

Evaluation criteria 6: The use of emphasis in the notation must be in accordance with the relative importance of the statements in the given model

This evaluation criterion was examined by viewing example diagrams drawn using the extensions specified in SecurityAssessmentUML. When performing the evaluation it is necessary to have the main point of the profile in mind. The main point of the diagrams is to highlight the undesirable behaviour in forms of threats, vulnerabilities and unwanted incidents, and the relation between these aspects. Compared to ordinary UML activity and sequence diagrams, undesirable behaviour is highlighted. Thus, this criterion seems to be fulfilled. However, testing on users must be performed in order to thoroughly answer this question. Due to time and resource limitations this is considered further work.

Evaluation criteria 7: The symbols used in the notation should strive for symbolic simplicity

This evaluation criterion was examined by evaluating the symbols according to two criteria; 1) the intuitiveness of the symbols used, and 2) the effort required to draw the symbols using pen and paper. The intuitiveness of the symbols was evaluated by the participants in the judgement study. The results of this study (see D.4), suggests that icons used to represent misusers, safeguards, threats and unwanted incidents should be easy to understand as the symbols are based on well-known metaphors. The misuser is inspired by the devil horn, threat is inspired by warning sign, safeguard by a lock and unwanted incident by a sad smiley. The symbol used for specifying vulnerabilities could be improved, however. In fact, one of the participants associates this icon with an explosion, which again could be associated with a catastrophic event. Hence, this icon should be improved in further work.

In terms of effort to draw the symbols, all symbols can easily be drawn using pen and paper, and all symbols can be drawn in black and white.

Chapter 10

Discussion

This report has addressed three main issues, SecurityAssessmentUML, security assessment process and the security documentation framework. The main result of the thesis is SecurityAssessmentUML, a UML profile for documenting output from risk assessment of security critical systems. The profile supports documentation of concerns connected to the two sub-processes risk identification and risk analysis. In particular, the profile supports specification of concrete threat scenarios, as well as a specification of how frequency of risk is calculated. The remaining two issues, a security assessment process refined from AS/NZS 4360 and its associated security documentation framework were developed as a means for eliciting requirements for the profile.

10.1 Completeness of profile

The main objective of the profile was to cover all required output from risk identification and risk analysis. Evaluation against the defined security ontology encountered one major limitation; the profile does not support specification of how the asset value is affected by an unwanted incident. Instead only documentation of estimated consequence value is supported. Inclusion of support for specifying effect on asset value should be subject for further work. Assuming that the documentation framework used as the basis for specifying requirements for the profile covers all necessary output, the ontological evaluation performed indicates that all other concepts are covered by the profile. This is further supported by the trial using SecurityAssessmentUML to document an example of risk assessment results. The profile allowed documentation of all information presented in the trial. However, in order to draw valid conclusions more extensive trials covering a representative sample of possible results from risk identification and risk analysis should be carried out. Furthermore, it is necessary to check whether the level of detail offered by the diagrams is appropriate for the target users.

10.2 Comparison to existing methods

A comparison between diagrams offered by CORAS UML profile and SecurityAssessmentUML suggests that SecurityAssessmentUML may be useful for complementing CORAS UML profile in cases where concrete descriptions of threat scenarios are required. The sequence and activity diagrams used for documenting output from risk identification highlight how threats may actually

occur. Furthermore, the activity diagram extended with basic fault tree notation provides an alternative to the State analysis diagrams defined by CORAS UML profile for specifying values for frequency/likelihood and consequence of unwanted incidents. In particular, SecurityAssessmentUML supports calculation of frequency, as well as input to the cost-benefit analysis performed when considering treatment options for risk. Compared to traditional textual assessment documentation, often presented in tables, diagrams defined by SecurityAssessmentUML highlight important information and provide a visual illustration of the relation between threats, vulnerabilities and unwanted incidents.

10.3 Trade-off between intuitive description and standard syntax

When specifying the profile, we focused on two aspects, intuitive description and preservation of standard UML syntax. Intuitive descriptions are important to increase the readability of the diagrams. The rationale for preserving standard syntax is twofold. Firstly, people familiar with UML syntax may be confused if UML concepts are used in untraditional ways. Although not in the target group, these people will have to draw the diagrams and explain them to other stakeholders. Secondly, one of the main aims of model-based risk assessment is that undesirable behaviour should be specified in the same way as normal behaviour.

However, these features are sometimes in conflict. Increasing the readability of diagrams for non-technical stakeholders by making them more intuitive may affect the use of standard syntax. In general, we tried to preserve the standard syntax. Instead, we focused on highlighting undesirable behaviour by use of icons and by removing detailed information from the diagrams. In terms of the fault-tree inspired activity diagram, which is a hybrid between traditional activity diagram notation and traditional fault tree notation, use of standard UML syntax was compromised. Thus, readers should be provided with description on how to read the diagrams.

10.4 Validity of evaluation

Evaluation was not the main issue in this thesis and was only conducted in order to achieve a preliminary indication of the quality and the usefulness of the UML profile developed. Despite of this, three different methods were used for evaluating the profile; judgement studies and simplified versions of formal theory and field studies. Ideally, the combination of these three evaluation strategies should allow for generalization, precision and realism. However, due to limited time and resource, extensive evaluation was impossible. Hence, further evaluation must be performed in order to draw valid conclusions.

The main problem with the ontological evaluation was that the evaluation was performed by the person developing the profile. This situation is not ideal. There are two reasons for this. First, the evaluator had deep knowledge about the profile. More importantly, however, due to lack of objectivity there is a danger of "fishing" for a specific result.

The main problem with the trial was that only one example was used. In order to draw valid conclusions, further evaluation using a representative sample of assessment results must be performed. To find a representative sample, we also need to check whether the level of detail for documentation is appropriate for stakeholders knowledge.

When it comes to the expert judgements two main factors represent a threat to the validity of the results; the limited amount of persons participating in the study and the fact that all participants were technical people.

10.5 Suitability of UML for non-technicians

The work performed in this thesis is a contribution to research focusing on enhancing communication among stakeholders participating in security assessments. In particular, focus was on specifying a means for communication among non-technical people. Knowing that UML diagrams originally are developed for use by people involved in the production, deployment, and maintenance of software, we find it necessary to question the suitability of using UML as basis for communication among non-technicians. In particular, we believe that UML sequence and activity diagrams may be hard to understand for non-technical people. Use case diagrams are traditionally easier to understand, but these diagrams does not provide sufficient level of detail for describing concrete scenarios and is therefore not included in the profile. However, we do not disregard that the diagrams may be useful assuming that stakeholders get sufficient training in how to read them. Hence, testing of the profile on non-technical users will be interesting and should be subject for further work.

Diagrams created using SecurityAssessmentUML is possibly more suitable for technical experts and people familiar with UML. However, a problem is that one of the diagrams (the fault-tree inspired extension to activity diagram used for risk analysis) has compromised the normal UML syntax in order to make it more intuitive for non-technical people. The syntax used in this diagram does not completely conform to neither normal fault tree nor traditional activity diagram interpretation. In this case it is of crucial importance that the readers are provided with detailed descriptions on how to read and interpret the diagrams. Otherwise, people familiar with UML might be confused because they expect the diagrams to follow traditional UML syntax. Technical stakeholders might also require more detailed modelling. Furthermore, technical people and people familiar with UML might prefer use of text strings in brackets instead of icons for representing stereotypes. Thus, a translation between icon and text-stereotype representation should be supported by modelling tools.

10.6 Adaption to the safety domain

Although developed for use in the security domain, the profile can probably be used in the safety domain as well. The reason for this is that all methods for risk identification and risk analysis are adapted from the safety domain. Even though the safety ontology is not completely similar, we believe that output from risk identification and risk analysis of safety critical systems will produce the same list of concerns as for the security domain. Investigation of the suitability of SecurityAssessmentUML to document results from risk identification and risk analysis of safety critical systems could be considered in further work.

Chapter 11

Conclusion and further work

11.1 Conclusion

In this thesis, we have specified SecurityAssessmentUML, a UML profile supporting model-based risk identification and risk analysis of security critical systems. The profile aims at specifying concrete scenarios demonstrating the relationship between outputs from risk identification, and at providing a means for documenting risk analysis results.

Based on the extensions to UML1.4, three types of diagrams can be made. For risk identification the profile supports two types of diagrams, an extension to sequence diagrams and an extension to activity diagrams. For risk analysis the profile supports an extension to activity diagrams, which uses notation from the risk analysis method FTA. All stereotypes are defined both by text strings in brackets and by icons, in order to suit both technical and non-technical readers. Ideally, diagrams should be understandable for non-technicians without compromising on the standard syntax as defined by UML1.4. This is not the case for the fault-tree inspired activity diagram, which uses a hybrid between traditional fault tree and traditional activity diagram notation.

Due to lack of extensive testing SecurityAssessmentUML is in a draft version. However, preliminary evaluation of the profile has identified some areas for improvement. The most significant limitation is the lack of support for specifying unwanted incidents effect on asset value. Thus, the profile does not entirely meet its stated goal to cover all required output from risk identification.

The work is a contribution to research on model-based risk assessments of security critical systems. The critical point is that the diagrams should be understandable for non-technical users. We believe sequence and activity diagrams might seem confusing for people inexperienced with UML. Thus, we believe education in reading the diagrams might be necessary.

11.2 Further work

Based on the discussion and constraints made when working with the thesis, two main areas for further work are identified; extensive evaluation of the profile and further development of the profile. The list below includes both aspects.

- **Support for documenting effect on asset value**

Currently, SecurityAssessmentUML lacks support for specifying how asset value is affected

by an unwanted incident. However, since this is essential in security assessments it should be included in a later version of the profile.

- **Extension to support documentation of threats categorised as system failure**
Currently, SecurityAssessmentUML only supports documentation of threats categorised as attacks. Support for specifying system failure should be subject for further work.
- **Extension to support all sub-processes in the risk management process**
Currently, SecurityAssessmentUML supports documentation of output from two of the sub-processes in the risk management process. The profile should be extended to allow documentation of output from all sub-processes. Requirements for support of sub-process 1, 3 and 4 can be specified based on the security documentation framework presented in Chapter 6. In order to specify requirements for the two parallel sub-processes, the security documentation framework must be extended as well.
- **Extension to support documentation appropriate for system designers and developers**
The current version of SecurityAssessmentUML should be extended to facilitate documentation suitable for system designers and developers. In order to achieve this, the profile might need to cover additional details.
- **Extensive evaluation**
In order to draw valid conclusions about the usefulness of SecurityAssessmentUML, whether it further communication and interaction among stakeholders without prior knowledge of UML and whether all necessary concepts are covered, extensive evaluations must be performed. The evaluation should include trials using more detailed and varied assessment results, as well as evaluation on target users of the profile.

Appendix A

Glossary

Accountability The property that ensures that the actions of an entity may be traced uniquely to the entity [26]

Asset - Something to which an organisation directly assigns value and, hence, for which the organisation requires protection [5].

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 loss, loss of revenue, market share, or company image [5].

Authenticity The property that ensures that the identity of a subject or resource is the one claimed. Authenticity applies to entities such as users, processes, systems and information [26]

Availability The property of being accessible and usable upon demand by an authorized entity [26]

Confidentiality The property that information is not made available or disclosed to unauthorized individuals, entities or processes [26]

Consequence - The outcome of an event expressed qualitatively or quantitatively, being a loss, injury, disadvantage or gain. There may be a range of possible outcomes associated with an event [4].

Context - The strategic, organisational and risk management context in which the rest of the risk management process will take place [4].

Data integrity The property that data has not been altered or destroyed in an unauthorized manner [26]

Entity - An entity that becomes an asset when assigned value by a stakeholder [37].

Fault tree analysis A systems engineering method for representing the logical combinations of various system states and possible causes which can contribute to a specified event (called the top event) [4].

Frequency - A measure of the rate of occurrence of an event expressed as the number of occurrences of an event in a given time [4].

Integrity See data and system integrity

IT security All aspects related to defining, achieving, and maintaining confidentiality, integrity, availability, non-repudiation, accountability, authenticity, and reliability. [26]

IT security policy Rules, directives and practices that govern how assets, including sensitive information, are managed, protected and distributed within an organization and its IT systems [26].

Likelihood - Used as a qualitative description of probability or frequency [4]

Mis-actor - An actor who initiates functions that the system should not allow [50].

New definition: **Misuser** - An actor who initiates system misuse. The misuser may be either internal or external to the organisation.

Misuse - A function that the system should not allow. Some kinds of misuse are most likely to be performed by intent whereas other may happen accidentally. Some require insiders or people with enormous skill, others not [50].

New definition: **Misuse** - An action that violates the security of a system. Some kinds of misuse are most likely to be performed by intent whereas other may happen accidentally. Some require insiders, other types of misuse may be performed from external locations.

Non-repudiation The ability to prove that an action or event has taken place, so that this event or action cannot be repudiated later [26].

Organization - A company, firm, enterprise, or other legal entity or part thereof, whether incorporated or not, public or private, that has its own function(s) and administration [4].

Probability - The likelihood of a specific event or outcome, measured by the ratio of specific events or outcomes to the total number of possible events or outcomes. Probability is expressed as a number between 0 and 1, with 0 indicating an impossible event or outcome and 1 indicating that an event or outcome is certain [4].

Reliability The property of consistent intended behaviour and results [26].

Risk - The potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organisation [26]. It is measured in terms of consequence and likelihood [4].

Risk analysis - A systematic use of available information to determine how often specified events may occur and the magnitude of their consequences [4]

Risk assessment - The overall process of risk analysis and risk evaluation [4].

Risk evaluation - The process used to determine risk management priorities by comparing the level of risk against predetermined standards, target risk levels or other criteria [4].

Risk evaluation criteria - A criteria against which risk is to be evaluated [4].

Risk identification - The process of determining what can happen, why and how [4].

Risk level- Classification of risk associated with a particular unwanted incident. The classification is calculated based on estimated values for consequence and likelihood/frequency [55].

Risk management - The culture, processes and structures that are directed towards the effective management of potential opportunities and adverse effects [4].

Risk management process - The systematic application of management policies, procedures and practices to the tasks of establishing the context, identifying, analysing, evaluating, treating, monitoring and communicating risk [4].

Risk priority - Value indicating the necessity for further treatment of the risk [4].

Risk treatment - Selection and implementation of appropriate options for dealing with risk [4].

Safeguard - A practice, procedure or mechanism that reduces risk [26].

Security attack - Any action that comprises the security of information owned by an organisation [52]

IT Security policy - Rules, directives and practices that govern how assets, including sensitive information, are managed, protected and distributed within an organization and its IT systems [26].

New definition: **Security policy** - Rules, directives and practices that govern how assets, including sensitive information, are managed, protected and distributed within an organization and its information systems.

Stakeholders - Those people or organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity [4].

System integrity The property that a system performs its intended function in an unimpaired manner, free from deliberate or accidental unauthorized manipulation of the system [26].

Target of Evaluation (ToE) - An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation [14]).

Threat - A potential cause of an unwanted incident which may result in harm to a system or organisation [26].

Unwanted incident - An undesired event that may reduce the value of an asset [37].

Vulnerability - A weakness of an asset or group of assets which can be exploited by one or more threats [26].

Appendix B

Research Agenda

The work on this thesis started with a literature study. The literature research was comprised of collecting and structuring relevant information about security critical systems, UML, AS/NZS 4360, CORAS, and other relevant areas.

Based on the findings in the literature study, a risk assessment framework particularly supporting security was specified. The main sources in this process were the study of AS/NZS 4360, CORAS risk documentation framework and general knowledge about the security domain. Questions posed when tailoring AS/NZS 4360 to the security domain were: 1) Which changes must be made to the generic standard in order to particularly support security assessments? and 2) What are the required input and output to each activity within the security management process? A necessary first step in this process was to specify the ontology for security assessments.

The next step was to specify requirements for the UML profile. As a basis for the specification, information about output from the activities within the security assessment process was used. Having defined the requirements, the knowledge of UML was used to extend the language. The main goal was to look at how output from a security assessment could be documented using UML and its standard extension mechanisms.

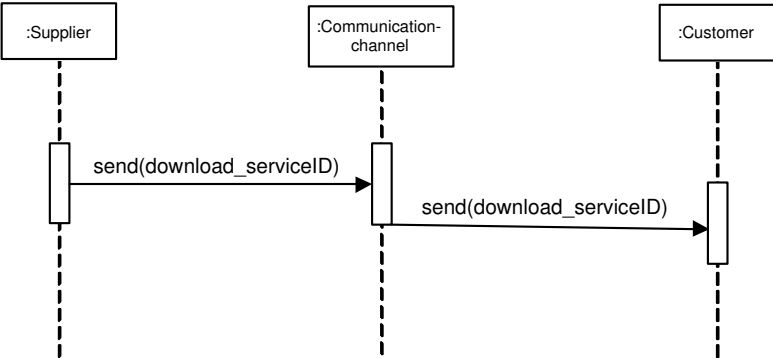
The last phase in the project was to evaluate the profile.

Appendix C

Evaluation schema used in judgement study

Normal system behaviour






Diagram 1 and 2 visualize an example result from risk assessment of an e-commerce system. The example risk assessment result specifies possible threats against a successful transmission of electronic service delivery from supplier to consumer after a successful purchase. Normal system behaviour is illustrated below. As indicated in the figure, the electronic service is successfully transferred from supplier to consumer via the communication channel without any form of interference. The parameter `download_serviceID` represents the electronic service.



Normal system behaviour

Presentation of the UML profile

The tables below provide a description of symbols used in the diagrams. The former table presents symbols used to highlight security aspects in the diagrams, the latter table presents symbols used to include fault tree concepts into activity diagrams. In the diagrams used for evaluation only the icons are used, i.e. textual descriptions in brackets are not included.

Modeling construct	Description
<< vulnerability >> 	Vulnerability
<< unwanted incident >> 	Unwanted incident
<< threat >> 	Threat
<< misuser >> 	Misuser
<< safeguard >> 	Safeguard






Modeling construct	Description	Comment
	AND-gate	Output occurs only if all inputs occur
	OR-gate	Output occurs if at least one of the inputs occurs
	Transfer down	Symbol used to denote further development in a cause-chain. The symbol is used when the same branch is involved in several paths and when the fault tree spans more than one page.
	Transfer up	
	Basic event	A basic event that requires no further development.
{frequency = value}		
{consequence = value}		
{risk level = value}		

Diagram 1: Sequence diagram

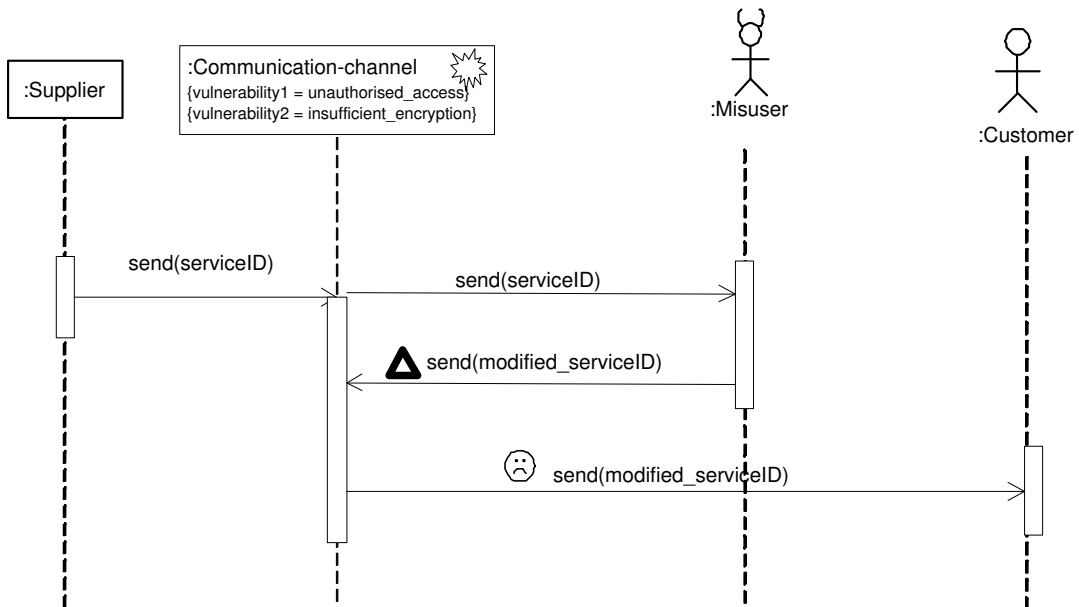


Diagram 1: Sequence diagram		
Identify and describe	Threat	
	Vulnerability	
	Unwanted incident	
	Misuser	
Explain the scenario in own words		

Diagram 2: Activity diagram

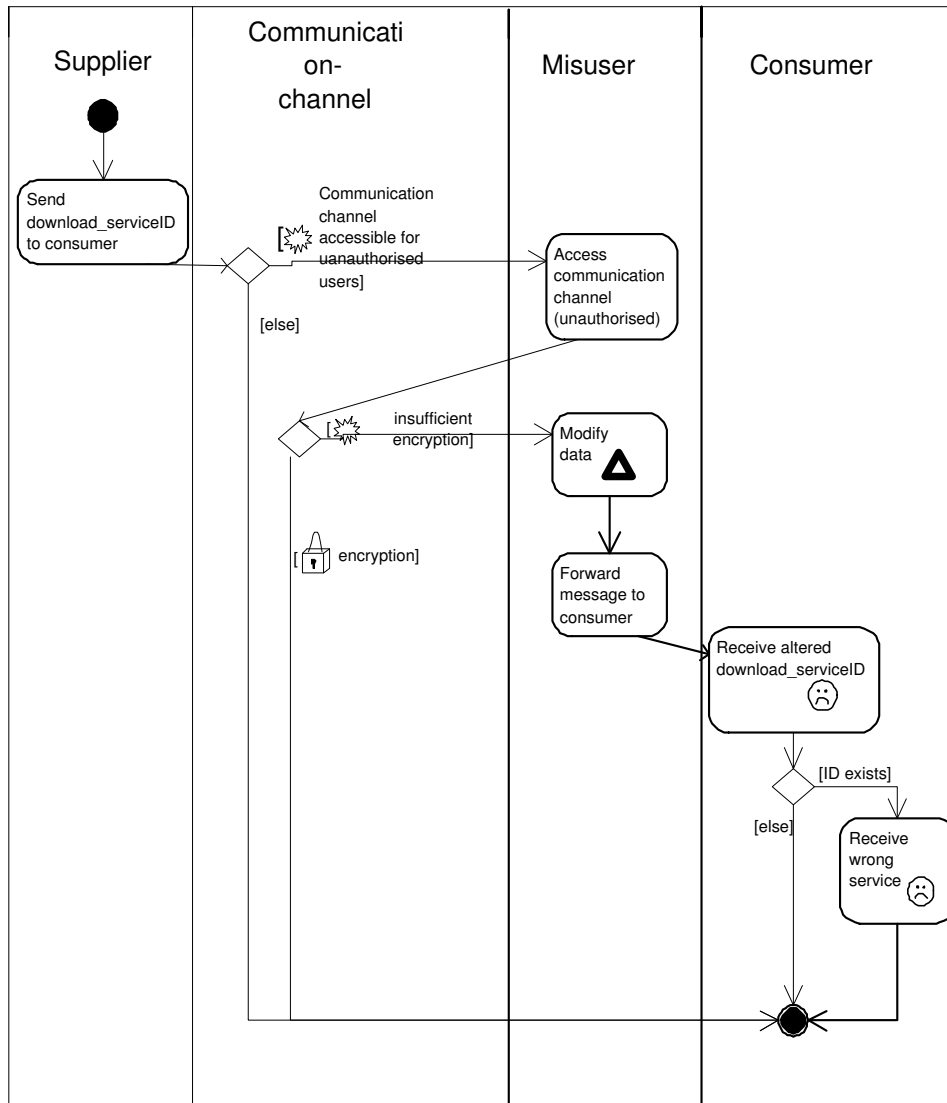


Diagram 2: Activity diagram		
Identify and describe	Threat	
	Vulnerability	
	Unwanted incident	
	Safeguard	
Explain the scenario in own words		

Diagram 3: Activity diagram inspired by fault tree notation

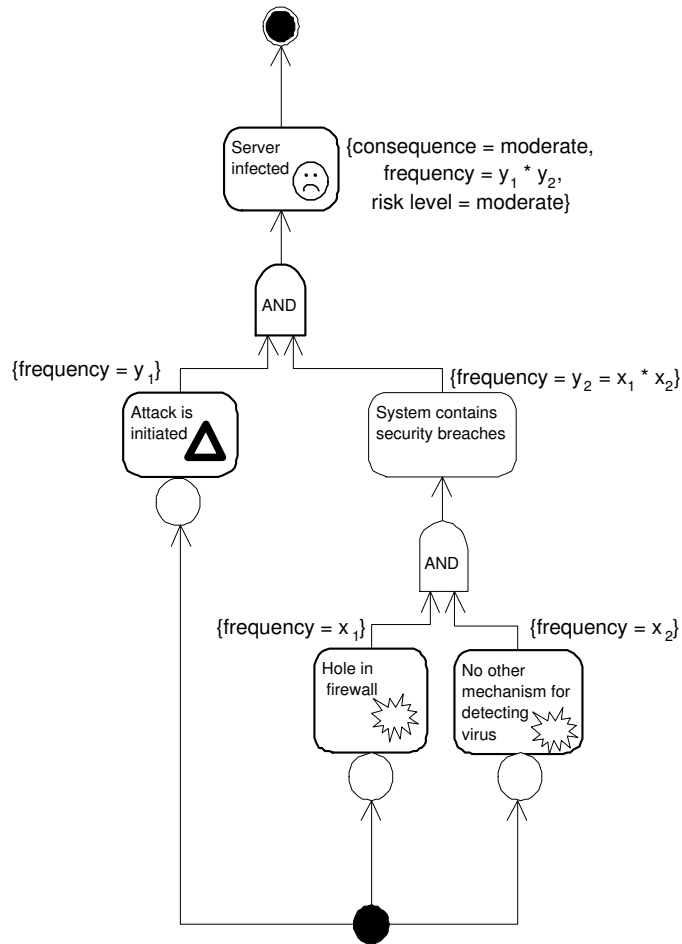


Diagram 3: Activity diagram using fault tree notation		
Identify and describe	Threat	
	Vulnerability	
	Unwanted incident	
	Safeguard	
	Consequence of unwanted incident	
	Frequency of unwanted incidents	
	Risk level of unwanted incident	
Explain the scenario in own words		

General questions about the symbols used in the profile		
Did you find the following symbols intuitive?	Threat	
	Vulnerability	
	Misuser	
	Safeguard	
	Unwanted incident	
Did you find it difficult to distinguish between the symbols?		

Appendix D

Results from the judgement study

The results from the judgement studies are included in the language they were conducted.

Table D.1: Results of judgement study on extension to sequence diagram for documenting results from risk identification

Question	Judge 1	Judge 2 (in norwegian)
Identify and describe threat	Misuser can access and modify data (download_serviceID)	Uønsket bruker kan endre meldingen som sendes fra leverandør til kunde
Identify and describe vulnerability	Insufficient encryption, unauthorized access, note: it is not clear to me why unauthorized access to the communication channel is possible	Kommunikasjonskanalen muliggjør uautorisert tilgang og det er utilstrekkelig kryptering.
Identify and describe unwanted incident	Download_serviceID is modified by an unauthorized user	En uautorisert bruker endrer meldingen som sendes fra leverandør til kunde
Identify and describe misuser	??? hacker	
Describe the scenario in own words	A hacker eavesdrops on the communication channel and masquerades as an authorized user in order to modify Download_serviceID (whatever that is?)	Mulighet for uautorisert tilgang kombinert med utilstrekkelig kryptering gjør at en uvedkommende bruker endrer p meldingen som sendes fra leverandør til kunde.

Table D.2: Results of judgement study on extension to activity diagram for documenting results from risk identification

Question	Judge 1	Judge 2 (in norwegian)
Identify and describe threat	Modification of data	Misuser kan modifisere melding som sendes fra Supplier til Customer
Identify and describe vulnerability	Insufficient encryption, possibility for unauthorised access to the communication channel (again this needs to be defined more clearly how/why this is possible)	Kommunikasjonskanal tilgjengelig for uautoriserte brukere
Identify and describe unwanted incident	Received altered download_serviceID; Wrong service received, DoS for authorised user	Meldingen som customer mottar er ikke som tiltenkt fra supplier; ønsket tjeneste mottas ikke og uønsket tjeneste mottas.
Identify and describe safeguard	Encryption	Kryptering i kommunikasjonskanalen
Describe the scenario in own words	A hacker has access to the communication channel and eavesdrops on the communication channel and modifies the intercepted data so that access to the service by the authorized user is denied	Dersom uautoriserte brukere har tilgang til kommunikasjonskanalen og det er utilstrekkelig kryptering, kan en uautorisert bruker sørge for at ønsket service stoppes og uønsket service mottas for Customer.

Table D.3: Results of judgement study on extension to activity diagram for documenting results from risk analysis

Question	Judge 1	Judge 2 (in norwegian)
Identify and describe threat	Attack is initiated	Et angrep initieres
Identify and describe vulnerability	Holes in the firewall and no mechanisms for detecting virus	Hull i brannvegg og ingen annen mekanisme for ådetektere virus.
Identify and describe unwanted incident	Server is infected by virus	Server infisert
Identify and describe consequence of unwanted incident	moderate	Moderat
Identify and describe likelihood of unwanted incident	$y1*y2$	$y1*y2$
Identify and describe risk level of unwanted incident	moderate	Moderat
Describe the scenario in own words	There are holes in the firewall and no mechanisms for detecting virus so that it is possible to attack the server and infect the server with a virus	Hvis det er hull i brannmur og samtidig mangler andre mekanismer for detektere virus vil systemet inneholde sikkerhetsbrudd. Dersom det samtidig initieres et angrep, vil serveren bli infisert. Sannsynligheter, konsekvenser og risikonivåer er som beskrevet.

Table D.4: Results on general questions about the profile

Question	Judge 1	Judge 2 (in norwegian)
Did you find the symbol for threat intuitive?	Yes	Ja
Did you find the symbol for vulnerability intuitive	Yes	Nei. Jeg assosierer dette med en eksplosjon, som igjen assosieres med en katastrofal hendelse (f.eks safety brudd). Dessverre kommer jeg i farten ikke p noen gode alternativer. (Hus uten tak, skilpadde uten skall etc. blir jo fort for kompliserte tegninger...).
Did you find the symbol for misuser intuitive?	Yes	Ja, veldig!
Did you find the symbol for safeguard intuitive?	Yes	Ja
Did you find the symbol for unwanted incident intuitive?	Yes	Ja
Did you find it difficult to distinguish between the symbols?	No	Nei (men se kommentaren over)

Appendix E

FMEA used for evaluation of SecurityAssessmentUML

Table E.1: FMEA table documenting result of FMEA analysis on the message Res(download_serviceID), sent from supplier to consumer. Adopted from [20]

ID	Parameter	Attribute	Intrudersailure source	Local effects	Failure modes	System Effects	Cause	Remarks
1b	download_serviceID	Manipulation	Outsider (hacker)	Manipulation of download_serviceID (the electronic file representing the requested service)	Wrong or corrupt file being transferred	1)Consumer is not able to receive the requested service; Or 2) Consumer receives wrong service.	Outsider (hacker) gains access to the communication channel between the supplier and the consumer altering data being transmitted.	This could lead to serious issues concerning trust relationship between consumer and supplier.

Bibliography

- [1] OMG (2001). Unified Modeling Language Specification. Version 1.4. <http://www.omg.org>.
- [2] C. J. Alberts, S. G. Behrens, R. D. Pethia, and W. R. Wilson. Operationally critical threat, asset, and vulnerability evaluation (octave) framework, version 1.0. Technical report, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, June 1999.
- [3] J. Allen, C. Alberts, S. Behrens, B. Laswell, and W. Wilson. Improving the security of networked systems. In Pam Bowers, editor, *LOCK OUT Network Predators*, volume 13 of *CrossTalk, The Journal of Defense Software Engineering*, pages 7–11. Reuel S. Alder, October 2000.
- [4] Australian/New Zealand Standard AS/NZS 4360:1999: Risk Management. Strathfield: Standards Australia.
- [5] Australian/New Zealand Standard AS/NZS 4444:1999: Information Security Management. Strathfield: Standards Australia.
- [6] T. Aven. *Pålitelighets- og risikoanalyse*. Universitetsforlaget, 2 edition, 1998. ISBN: 82-00-42527-4.
- [7] B. Barber and J. Davey. The use of the ccta risk analysis and management methodology CRAMM. In *Proc. MEDINFO92, North Holland*, pages 1589–1593, 1992.
- [8] G. Booch. *Object-oriented analysis and design with applicaitons*. Addison-Wesley, 2 edition, 1994. ISBN: 0805353402.
- [9] G. Booch, J. Rumbaugh, and Jacobsen. I. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999. ISBN: 0-201-57168-4.
- [10] A. Bouti and D. Ait Kadi. A state-of-the-art review of FMEA/FMECA. *International Journal of Reliability, Quality and Safety Engineering*, 1:515 – 543, 1994.
- [11] F. den Braber, T. Dimitrakos, B. A. Gran, K. Stølen, and J. Ø. Aagedal. Model-based risk management using UML and UP. To appear in the book titled UML and the Unified Process. IRM Press, 2003.
- [12] P. Clements, R. Kazman, and M. Klein. *Evaluating software architectures: Methods and case studies*. Addison-Wesley, 2002. ISBN: 020170482X.
- [13] CORAS IST-2000-25031 Web Site. <http://www.nr.no/coras>. 24 February 2003.
- [14] Common Criteria. Common Criteria for Information Technology Security Evaluation, 1999. <http://www.commoncriteria.org/>. 24 February 2003.

- [15] T. Dimitrakos, B. Ritchie, D. Raptis, and K. Stølen. Model based Security Risk Analysis for Web Applications: The CORAS approach. In *Electronic Workshops in Computing (eWiC): EuroWeb 2002 Conference - From e-science to e-business*, December 2002.
- [16] P.D. Goldis. Questions and answers about tiger teams. *The EDP Audit, Control and Security Newsletter*, 27:1–10, 1989.
- [17] B. Goodwin. Cybercrime - an inside job. *ComputerWeekly.com*. 31 August 2000.
- [18] I. S. Herschberg. Make the tigers hunt for you. *Computers and Security*, 7:197–203, 1988.
- [19] S. H. Houmb. Stochastic Models and Mobile E-Commerce: Appendix. Master’s thesis, Østfold University College, 2002.
- [20] S. H. Houmb. Stochastic Models and Mobile E-Commerce: Are stochastic models usable in the analysis of risk in mobile e-commerce. Master’s thesis, Østfold University College, 2002.
- [21] S. H. Houmb, F. den Braber, M. S. Lund, and K. Stølen. Towards a UML Profile for Model-Based Risk Assessment. In *Critical systems development with UML - Proceedings of the UML’02 workshop*, pages 79–91, September 2002.
- [22] S. H. Houmb and K. Stølen. Model-based security assessment. Unpublished work. Will be published as a technical report at NTNU in 2003.
- [23] IEC 61508: 2000 Functional Safety of Electrical/Electronic/Programmable Electronic (E/E/PE) Safety-Related Systems.
- [24] IEC 1025: 1990 Fault Tree Analysis (FTA).
- [25] ISO/IEC 10746 series: 1995 Basic reference model for open distributed processing.
- [26] ISO/IEC 13335: Information Technology - Guidelines for the management of IT Security. <http://www.iso.ch>.
- [27] ISO/IEC 17799: 2000 Information technology - Code of practise for information security management.
- [28] I. Jacobson, M. Christerson, P. Jonsson, and G. Övergaard. *Object-Oriented Software Engineering - A Use Case Driven Approach*. Addison-Wesley, 1992. ISBN: 0-201-54435-0.
- [29] B. Jung, I. Han, and S. Lee. Security threats to internet: a korean multi-industry investigation. *Information & Management*, 38:487–498, 2001.
- [30] J. Jurjens. UMLsec: Extending UML for Secure Systems Development. Software & Systems Engineering, Dep. of Informatics, Munich University of Technology.
- [31] J. Jürjens. *Principles for secure systems design*. PhD thesis, Wolfson College, 2002.
- [32] Jan Jürjens. Towards development of secure systems using UMLsec. In Heinrich Hussmann, editor, *Fundamental Approaches to Software Engineering, 4th International Conference, FASE 2001, held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001, Genova, Italy, April 2-6, 2001, Proceedings*, volume 2029 of LNCS, pages 187–200. Springer, 2001.

- [33] J. Krogstie. Evaluating uml using a generic quality framework. in Liliana Favre 'UML and the Unified Process', IDEA Group, to be published 2003.
- [34] K. Lano, K. Androutsopoulos, and D. Clark. Structuring and design of reactive systems using RSDS and B. proc. fase 2000. In *LNCS*, volume 1783, pages 97–111, 2000.
- [35] N. G. Leveson. *Safeware: System safety and computers*. Addison-Wesley, 1995. ISBN: 0-201-11972-2.
- [36] Torsten Lodderstedt, David A. Basin, and Jürgen Doser. SecureUML: A UML-Based Modeling Language for Model-Driven Security. In Jean-Marc Jézéquel, Heinrich Hussmann, and Stephen Cook, editors, *UML 2002 - The Unified Modeling Language. Model Engineering, Languages, Concepts, and Tools. 5th International Conference, Dresden, Germany, September/October 2002, Proceedings*, volume 2460 of *LNCS*, pages 426–441. Springer, 2002.
- [37] M. S. Lund, F. den Braber, and K. Stølen. Maintaining results from security assessments. In *Proceedings of the Seventh European Conference On Software Maintenance And Reengineering (CSMR'03)*, IEEE Computer Society, 2003.
- [38] J. E. McGrath. *Groups: Interaction & Performance*. Prentice Hall, 1984. ISBN: 0-13-365700-0.
- [39] Ministry of Defence. Interim defence standard 00-58/1: HAZOP Studies on Systems Containing Programmable Electronics. Directorate of Standardization (2000).
- [40] Ministry of Defence. Defence Standard 00-58 Issue 2: Hazop Studies on Systems containing Programmable Electronics, part 1: Requirements, 2000.
- [41] A. L. Opdahl and B. Henderson-Sellers. Ontological evaluation of the UML using the Bunge-Wand-Weber Model. *Software and Systems Modeling*, 1:43–67, 2002.
- [42] C. P. Pfleeger. *Security in Computing*. Prentice-Hall, 1997. ISBN: 0-13-185794-0.
- [43] R. K. Rainer, C. A. Snyder, and H. H. Carr. Risk analysis for information technology. *Journal of Management Information Systems*, 1:129–147, 1991.
- [44] D. Raptis, T. Dimitrakos, B. A. Gran, and K. Stølen. The coras approach for model-based risk analysis applied to the e-commerce domain. In *Proceedings Communication and Multimedia Security Conference, 2002. Sept. 26-27, 2002, Portoroz, Slovenian*, 2002. ISBN: 82-7017-397-5.
- [45] M. Rausand. *Risikoanalyse: Veiledning til NS 5814*. Tapir, 1991. ISBN: 82-519-0970-8.
- [46] F. Redmill, M. Chudleigh, and J. Catmur. *System safety: HAZOP and software HAZOP*. John Wiley and Sons, 1999. ISBN: 0-471-98280-6.
- [47] S. M. Ross. *Introduction to probability models*. Academic Press, 6 edition, 1997. ISBN: 0-12-598470-7.
- [48] J. Rumbaugh, M. Blaha, W. Pomerlani, F. Eddy, and W. Lorensen. *Object-oriented Modelling and Design*. Prentice Hall, 1991. ISBN: 0-13-630054-5.
- [49] J. Rumbaugh, Jacobsen. I, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999. ISBN: 0-201-30998-X.

- [50] G. Sindre and A. L. Opdahl. Capturing security requirements through misuse cases. In D. Langmyhr, editor, *Proceedings of Norsk informatikkonferanse - NIK'2001, Trondheim, Norway, 2001*, pages 219–230. Tapir, 2001.
- [51] C. Sluman, J. Ø. Aagedal, M. S. Lund, and E. F. Jr. Ecklund. Response to the OMG RFP for modeling quality of service and fault tolerance characteristics and mechanisms, September 2002. OMG document number realtime/2002-09-02.
- [52] W. Stallings. *Network Security Essentials: Applications and Standards*. Prentice-Hall, 2000. ISBN: 0-13-016093-8.
- [53] K. Stø len, F. den Braber, R. Fredriksen, B. A. Gran, S. H. Houmb, M. S. Lund, Y. C. Stamatiou, and J. Ø . Aagedal. Model-based risk assessment - the CORAS approach. In *In Proc. 1st iTrust Workshop, 2002*, 2002.
- [54] K. Stø len, F. den Braber, T. Dimitrakos, R. Fredriksen, B. A. Gran, S. H. Houmb, Y. C. Stamatiou, and J. Ø. Aagedal. Model-based risk assessment in a component-based software engineering process: The CORAS approach to identify security risks. In F. Barbier, editor, *Business Component-Based Software Engineering*, pages 189–207. Kluwer, 2003.
- [55] N. Storey. *Safety-critical computer systems*. Addison-Wesley, 1996. ISBN: 0-201-42787-7.
- [56] Udo Voges, editor. *Security Assessments of Safety Critical Systems Using HAZOPs*, volume 2187 of *Lecture Notes in Computer Science*. Springer, 2001.
- [57] Y. Wand and R. Weber. On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information systems*, 3:217–237, 1993.