A Graphical Approach to Security Risk Analysis

Doctoral Dissertation by

Ida Hogganvik

Submitted to the Faculty of Mathematics and Natural Sciences at the University of Oslo in partial fulfillment of the requirements for the degree Ph.D

October 2007

Kolofonside (Copyright-side som settes av forlaget)

Dedikasjoner
Acknowledgements

List of Original Publications

I. Ida Hogganvik and Ketil Stølen. ***On the Comprehension of Security Risk Scenarios***. In proceedings of the 13th International Workshop on Program Comprehension (IWPC'04), pages 115-124. IEEE Computer Society, 2004.

II. Ida Hogganvik and Ketil Stølen. ***Risk Analysis Terminology for IT-systems: Does it match intuition?*** In proceedings of the 4th International Symposium on Empirical Software Engineering (ISESE'05), pages 13-23. IEEE Computer Society, 2005.

III. Ida Hogganvik and Ketil Stølen. ***A Graphical Approach to Risk Identification, Motivated by Empirical Investigations***. In proceedings of ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS'06) (formerly the UML series of conferences), volume 4199 of *Lecture Notes in Computer Science,* pages 574-588. Springer Verlag, 2006.

IV. Heidi E. I. Dahl, Ida Hogganvik and Ketil Stølen. ***Structured Semantics for the CORAS Security Risk Modelling Language***. In proceedings of 2nd International Workshop on Interoperability Solutions on Trust, Security, Policies and QoS for Enhanced Systems (IS-TSPQ'07), pages 79-92. Helsinki University Printing House, 2007.

V. Folker den Braber, Ida Hogganvik, Mass Soldal Lund, Ketil Stølen and Fredrik Vraalsen. ***Model-based security analysis in seven steps – a guided tour to the CORAS method***. Vol. 25 (1) of BT Technology Journal, pages 101-17. Springer Verlag, 2007.

VI. Fredrik Vraalsen, Tobias Mahler, Mass Soldal Lund, Ida Hogganvik, Folker den Braber, Ketil Stølen. ***Assessing Enterprise Risk Level: The CORAS Approach***. In *Advances in Enterprise Information Technology Security,* pages 311-333. Djamel Khadraoui and Francine Herrmann (Eds.), Information Science Reference, 2007.

VII. Ida Hogganvik and Ketil Stølen. ***Investigating preferences in graphical risk modeling***. Technical report, SINTEF A57, 2007. Submitted to the Journal of Empirical Software Engineering.

The publications I-VII are available as Appendices A-G. In the case of publication I, II and IV we have included the full technical reports which are extended, slightly revised versions of the published papers.

Table of Contents

1 INTRODUCTION 1

1.1 OUR WORK 1
1.2 THE CONTRIBUTION OF THE THESIS 2
1.2.1 New Artifacts 2
1.2.2 Results in the Form of Evaluations 4
1.3 THE STRUCTURE OF THE THESIS 6
1.4 HOW TO READ THE THESIS 7

2 PROBLEM AREA 9

2.1 INTRODUCTION 9
2.2 THE NEED FOR A SPECIAL PURPOSE LANGUAGE 9
2.3 PROBLEM CHARACTERIZATION 11
2.3.1 A Common Basis for Security Analysis Terminology 11
2.3.2 A Language for Specifying and Documenting Security Analysis Findings 12
2.3.3 Modeling Instructions for the User 13
2.3.4 A Way of Evaluating the Quality of Security Risk Modeling Approaches 13

3 STATE OF THE ART 15

3.1 RISK MANAGEMENT 15
3.2 SECURITY RISK ANALYSIS METHODS 16
3.2.1 The CORAS Security Analysis Method 16
3.2.2 OCTAVE 17
3.2.3 CRAMM 17
3.2.4 Facilitated Risk Assessment Process 18
3.2.5 Microsoft's Security Risk Management 19
3.2.6 Other Methods 19
3.2.7 Summary 20
3.3 STANDARDS RELEVANT TO SECURITY- AND RISK ANALYSIS 20
3.3.1 AS/NZS4360 20
3.3.2 ISO/IEC13335 21
3.3.3 BS7799/ISO17799 22
3.3.4 ISO15408 (The Common Criteria) 22
3.3.5 IEC61508 22
3.3.6 ISO21827 23
3.3.7 Other Standards and Guidelines 23
3.3.8 Summary 24
3.4 RISK- AND SYSTEM ANALYSIS TECHNIQUES 24
3.4.1 SWOT 25
3.4.2 FMEA/FMECA 25
3.4.3 HazOp 25
3.4.4 Fault Tree 26
3.4.5 Event Tree 27
3.4.6 Cause-consequence Diagram 28
3.4.7 Attack Tree 28
3.4.8 OCTAVE Threat Tree 29
3.4.9 Bayesian Network 30
3.4.10 Markov Analysis 31
3.4.11 Block Diagram 31
3.4.12 Other Techniques 32
3.4.13 Summary 33
3.5 RISK DOCUMENTATION TECHNIQUES 34
3.5.1 The UML Profile 34
3.5.2 Misuse Case 38
3.5.3 UML Variants for Security 39
3.5.4 Simple Technique for Illustrating Risk 40
3.5.5 Riskit Graph 40
3.5.6 Microsoft's Threat Modeling and Related Approaches 41

3.5.7 Summary ... *41*
3.6 INFORMATION VISUALIZATION TECHNIQUES ... 42
3.6.1 Representing Graph Navigation ... *42*
3.6.2 Representing Vulnerabilities ... *43*
3.6.3 Representing Risks ... *44*
3.6.4 Summary ... *46*

4 RESEARCH METHOD ... 47

4.1 TECHNOLOGY RESEARCH ... 47
4.1.1 Methods for Evaluating the Results ... *48*
4.2 OUR RESEARCH METHOD ... 49
4.2.1 Developing a Common Basis for Security Analysis Terminology ... *50*
4.2.2 Developing a Language for Specifying and Documenting Security Analysis Findings ... *51*
4.2.3 Developing Modeling Instructions for the User ... *55*
4.2.4 Developing a Way of Evaluating the Quality of Security Risk Modeling Approaches ... *56*

5 INITIAL INVESTIGATIONS ... 59

5.1 THE ORIGINAL CONCEPTUAL MODEL OF THE CORAS PROJECT ... 59
5.2 INVESTIGATING CONCEPTS AND TERMINOLOGY ... 60
5.2.1 The First Terminology Investigation ... *60*
5.2.2 The Second Terminology Investigation ... *61*
5.3 INVESTIGATING THE USE OF SPECIALIZED GRAPHICAL SYMBOLS ... 63

6 LANGUAGE REQUIREMENTS DEFINED WITHIN A QUALITY FRAMEWORK ... 65

6.1 THE CORE SECURITY RISK SCENARIOS ... 65
6.1.1 Phase I: Establishing the Context ... *66*
6.1.2 Phase II: Identifying Risks ... *68*
6.1.3 Phase III: Estimating Risks ... *71*
6.1.4 Phase IV: Evaluating Risks ... *72*
6.1.5 Phase V: Identifying Treatments ... *72*
6.2 QUALITY REQUIREMENTS ... 74
6.3 ADAPTING SEQUAL TO THE SECURITY ANALYSIS SETTING ... 75
6.3.1 Domain Appropriateness ... *76*
6.3.2 Participant Language Knowledge Appropriateness ... *81*
6.3.3 Knowledge Externalizability Appropriateness ... *82*
6.3.4 Comprehensibility Appropriateness ... *82*
6.3.5 Technical Actor Interpretation Appropriateness ... *86*
6.3.6 Organizational Appropriateness ... *87*

7 EVALUATION OF THE UML PROFILE ... 89

7.1 MODELING THE CORE SECURITY RISK SCENARIOS (SUMMARY) ... 89
7.2 EVALUATION OF THE LANGUAGE QUALITY (SUMMARY) ... 90
7.3 CONCLUSION ... 91

8 THE CORAS SECURITY RISK MODELING LANGUAGE AND GUIDELINE ... 93

8.1 THE SYNTAX AND SEMANTICS ... 93
8.1.1 The Conceptual Foundation ... *93*
8.1.2 The Graphical Syntax ... *94*
8.1.3 The Textual Syntax ... *94*
8.1.4 The Structured Semantics ... *96*
8.2 THE GUIDELINE (AN EXAMPLE DRIVEN INTRODUCTION) ... 98
8.2.1 Context Establishment ... *98*
8.2.2 Risk Identification ... *101*
8.2.3 Risk Estimation ... *103*
8.2.4 Risk Evaluation ... *105*
8.2.5 Treatment Identification ... *108*
8.2.6 Additional Explanations, Tips and Hints ... *110*

9 EVALUATION OF THE CORAS LANGUAGE AND THE GUIDELINE ... 113

9.1 EVALUATION USING QUESTIONNAIRES ... 113

9.2 EVALUATION BASED ON EXPERIENCE REPORTS FROM FIELD TRIALS 115
9.2.1 he Conceptual Foundation 117
9.2.2 The CORAS Language 118
9.2.3 The Guideline 122
9.3 EVALUATION USING THE QUALITY FRAMEWORK 125
9.3.1 Modeling the Core Security Risk Scenarios 125
9.3.2 Evaluating Against the Quality Requirements 128
9.4 THREATS TO THE VALIDITY OF THE EVALUATIONS 130
9.4.1 The Validity of the Design Investigation 130
9.4.2 Strengths and Weaknesses Related to the Field Trials 130
9.4.3 Threats to the Validity of the Quality Framework Evaluation 131

10 DISCUSSION 133

10.1 FULFILLING THE SUCCESS CRITERIA 133
10.1.1 The First Success Criterion 133
10.1.2 The Second Success Criterion 134
10.1.3 The Third Success Criterion 135
10.1.4 The Fourth Success Criterion 136
10.2 THINGS WE COULD HAVE DONE DIFFERENTLY AND FUTURE WORK 137
10.2.1 Strategies and Solutions Related to Design 137
10.2.2 Strategies for Evaluation 138

REFERENCES 141

TABLE OF FIGURES 153

LIST OF TABLES 155

Appendix A – Empirical Investigations of the CORAS Language for Structured Brainstorming

Appendix B – Risk Analysis Terminology for IT systems: Does it Match Intuition?

Appendix C – A Graphical Approach to Risk Modeling, Motivated by Empirical Investigations

Appendix D – Structured Semantics for the CORAS Security Risk Modeling Language

Appendix E – Model-based Security Analysis in Seven Steps – a Guided Tour to the CORAS Method

Appendix F – Assessing Enterprise Risk Level: The CORAS Approach

Appendix G – Investigating Preferences in Graphical Risk Modeling

Appendix H – Quality Evaluation of the CORAS Language

1 Introduction

Over time, we have gradually become more and more dependent on computerized systems of various kinds. Infrastructure like mobile phones, Internet, e-mail, online bank services etc. are important elements in our everyday life. In many cases we cannot imagine how to manage without them. As the systems increase in complexity, the number of security risks is most likely to rise. Security incidents occur on a daily basis within most companies, and the systems we surround us with are vulnerable to attacks [95]. In CSI/FBI Computer Crime and Security Survey for 2005 [35] 74% of the companies reported security incidents. The worst security breaches within UK companies are estimated to cost about 90.000£ (≈133.000€), and the companies experience about 19 incidents of that kind per year [128]. Virus attacks are the major source of the largest financial losses, with unauthorized access on second place [35]. Security incidents may have critical effects also for systems traditionally not associated with security breaches. In 2003 the Davis-Besse nuclear plant (Ohio, US) was hit by the SQL Slammer worm. The worm disabled crucial control systems for about five hours. Fortunately, the systems had analogue backups that remained unaffected. The so-called "northeastern blackout" in 2003 was a failure in the electrical power grid that left 50 million people in North America without power [127]. The incident affected transportation, communication, industry, water supply etc., and it took several days to restore everything back to normal operation. The failure was not caused by any form of malicious attack, but a contributing factor to the incident was a software bug.

The request for safe, secure and reliable systems have lead to a demand for good security risk analysis methods. The methods must handle detailed analyses of technical aspects, as well as more high-level enterprise level analyses. This means that the methods must be general, but not too general in order to provide sufficiently support to the analysis process. Many security risk analysis techniques include people familiar with the target of analysis as participants, under the guidance of an analysis leader. A well designed security risk analysis method should support the analysis process by facilitating communication, interaction and understanding between these participants. Security risk analysis is often considered to be complicated and time-consuming. The participants in a security risk analysis often have to deal with complex systems and advanced technology. In many cases the only visible part of the system analyzed is its interfaces and effects, while the internal structure is hidden. Consequently, the participants may experience difficulties in understanding and analyzing the system with respect to security risks. This makes room for errors and misunderstandings which may hamper the analysis process. Another important challenge of security risk analysis methods is how to facilitate the involvement of people with different backgrounds and competences. People who uses or maintain the system on a daily basis posses vital knowledge about how the system is actually used, not only how it is intended to be used. Combining this type of information with the knowledge of the system designers, developers and owners is believed to benefit the security risk analysis results. The challenge is then to facilitate communication and collaboration between these different groups of people when they meet in the security risk analysis setting. In this thesis both "security risk analysis" and its abbreviation "security analysis" will be used.

1.1 Our Work

Our work has been to develop a graphical approach to threat and risk modeling that supports the security analysis process. There exists no such thing as *the* correct

representation of threats, vulnerabilities, risks etc. since this highly depends on the situation at hand, but it makes sense to look for the *preferable* representation with respect to a specific security analysis approach. In this work we target security analyses that use brainstorming sessions to identify and analyze security risks. These analyses are characterized by their involvement of people with thorough knowledge of the target of the analysis, like users, designers, developers and decision makers.

Our graphical approach to security risk modeling contributes to solving three issues related to security analysis. The first is *how to facilitate communication in a group consisting of people with different backgrounds and competences?* Our aim has been to provide the participants with a mean for communication that covers both technical and more high-level information, without being too complicated to understand. Offering a common communication basis will hopefully reduce misunderstandings and thereby give a more correct risk picture.

Second, *how to estimate the likelihood and consequences of the risks*? The existence of reliable data on which this can be based on is unlikely. The participants must use their expert knowledge, experience and familiarity with the domain to estimate both the likelihood and the consequences of incidents that might never happened to this day. The estimation process may easily become complex and difficult to follow, particularly when the number of risks is high. Our aim has been to offer a structured, graphical risk picture to make the complexity more manageable. A graphical representation may illustrate who or what caused the incidents and the weaknesses in the system that made them possible.

Third, *how to document the security analysis in a comprehensible manner?* The findings of a security analysis constitute vital information not only to the participants in the analysis, but to the organization as a whole. The results need to be documented in a way that is understandable even for those not involved directly in the analysis. Our aim has been to define a documentation method that should be more or less "self-explaining" and not rely on extensive training to be understood.

We believe our approach to security risk modeling will contribute towards making the above challenges more manageable. We have aimed to make a language that is easily understandable, even for people without training in modeling or security analysis.

1.2 The Contribution of the Thesis

The main contributions of this thesis are a set of new artifacts and the results from evaluating the new artifacts. First we summarize the artifacts, then the evaluation results:

1.2.1 New Artifacts

A) A conceptual foundation for security risk analysis:
We have specified main security risk analysis concepts and their relationships in a conceptual model. The conceptual foundation originates from the CORAS method and is based on international standards within security and risk analysis. A conceptual foundation may help to reduce misunderstandings with respect to interpretation of terminology. It may also contribute to a consistent use of terms both in the security analysis process and in security risk analysis modeling.

Described in: *Chapter 6 – Language Requirements Defined within a Quality Framework.*

B) A framework for evaluating the quality of security risk modeling languages:
To assess the quality of security risk modeling languages we have constructed a quality evaluation framework. The framework covers both practical modeling tasks as well as the more theoretically pats of a language like domain appropriateness and comprehensibility appropriateness. The framework consists of two main parts: (1) a detailed description of core security risk scenarios which a security risk modeling language should be able to express used as a "benchmarking test", and (2) a set of quality requirements based on an extended version of SEQUAL (quality framework for modeling languages). The framework has been designed with two goals; first to capture the rationale for our analytical evaluation. Second the framework makes it possible to compare the quality of different security risk modeling languages.

Described in: *Chapter 5 – Initial Investigations.*

C) A language for security risk modeling:
We present in our work a security risk modeling language that through its customized diagram types supports the entire security analysis process. Since it accompanies the CORAS method for security analysis it will in this thesis be referred to as the CORAS security risk modeling language (abbreviated to the CORAS language). The CORAS language has five diagram types: (1) asset diagram, (2) threat diagram, (3) risk diagram, (4) treatment diagram, and (5) treatment overview diagram. The asset diagram is used initially in the analysis to support asset identification, valuation and specification of risk acceptance levels. The threat diagram is the most central diagram type in the language. First, it is used in risk identification to support specification of threats, vulnerabilities, threat scenarios and unwanted incidents. Then, during risk estimation the threat diagram forms the basis for estimating likelihood and consequences of the different unwanted incidents. Finally, the threat diagram is the starting point for the treatment identification where treatments are added to the diagram. The risk diagram supports risk evaluation by specifying the risks and including which are acceptable and which are not. The treatment overview diagram is based on the treatment diagram and gives an overview of non-acceptable risks and the suggested treatments. This can be seen as a plan for mitigating the risks, and is suitable for presentation purposes. The CORAS language is not restricted to security analyses of information systems. We show this in *Appendix F* by using an early version of the language in an analysis of security-, legal- and trust issues related to collaborative engineering in virtual organizations within the aerospace industry.

Described in: *Chapter 8 – The CORAS Security Risk Modeling Language and Guideline, Examples of use in: Appendix H – Quality Evaluation of the CORAS Language, Appendix C – A Graphical Approach to Risk Modeling, Motivated by Empirical Investigations, Appendix E – Model-based Security Analysis in Seven Steps – a Guided Tour to the CORAS Method and Appendix F – Assessing Enterprise Risk Level: The CORAS Approach.*

D) A structured semantics for the security risk modeling language:
To ensure that the semantics of the CORAS security risk modeling language is understood in the same manner by everyone reading them, we have developed a textual syntax that maps from the graphical syntax to precise English. The mapping is based on the Backus–Naur form notation (EBNF) [81], and may also be seen as the first step towards a formal semantics.

Described in: *Chapter 8 – The CORAS Security Risk Modeling Language and Guideline (summary), Appendix D – Structured Semantics for the CORAS Security Risk Modeling Language*.

E) A guideline for the use of the security risk modeling language:
The full potential of a graphical security risk modeling language cannot be reached without providing a comprehensive guideline for how to use it. The guideline describes in detail how to create the different diagrams for each step during the security analysis process. In addition to realistic examples, it also provides guidance on how to conduct the analysis and what target descriptions that may be used. In *Appendix E* we provide a thorough, example driven introduction to a complete security analysis using the CORAS security modeling language, especially targeting practitioners. *Appendix C* contains a shorter example.

Described in: *Chapter 8 – The CORAS Security Risk Modeling Language and Guideline, Appendix C – A Graphical Approach to Risk Modeling, Motivated by Empirical Investigations (short version) and Appendix E – Model-based Security Analysis in Seven Steps – a Guided Tour to the CORAS Method (targeting practitioners).*

1.2.2 Results in the Form of Evaluations

We present two types of evaluations: investigations made during development to improve the artifacts, and evaluations of artifacts in their final form.

F) Empirical support for the conceptual foundation:
The conceptual foundation was revised on basis of the results from two empirical investigations with focus on ease of understanding. We first explored how risk analysis and security analysis concepts are understood based on their existence in every day language. The findings made us aware of particular difficult concepts that were considered when redesigning the conceptual foundation. In the second investigation we decided to use subjects with highly relevant backgrounds and competences, similar to potential users of a security risk modeling language. One of the most interesting findings was the discovery of the same pattern of difficult and easy concepts/relations as in the first study. This means that security risk analysis employs a conceptual foundation that may cause misunderstandings and confusion, even for highly skilled people.

Described in: *Chapter 5 – Initial Investigations (summary), Appendix A – Empirical Investigations of the CORAS Language for Structured Brainstorming and Appendix B – Risk Analysis Terminology for IT systems: Does it match Intuition?*.

G) Results from applying the quality framework:
We have applied the quality framework twice, first to evaluate the original UML profile [111, 112, 124], then to evaluate the new CORAS security risk modeling language. The results include complete modeling efforts of the core security risk scenarios and analytical evaluations according to the SEQUAL-based quality requirement framework. The results make it possible to compare the quality of the two security risk modeling languages.

Described in: *Chapter 7 – Evaluation of the UML Profile (summary, for full details we refer to [57]), Chapter 9 – Evaluation of the CORAS Language and Guideline, Appendix H – Quality Evaluation of the CORAS Language*.

H) Results from applying the language:
The CORAS security risk modeling language has from 2004 to 2006 been tested, evaluated and refined in six major industrial field trials. The security analyses have been conducted as part of the research project SECURIS (www.sintef.no/securis) which aims to investigate and improve the CORAS method by applying it in real security analyses. Each analysis required about 250 person hours from the analysis team and 50-100 hours from the client. The representatives of the client argued that the graphical language made it easier to actively involve the participants in the analysis and helped ensuring an effective communication between the analysis team and the participants. They also found the notation itself easy to understand and remember. It was considered to be a good way of visualizing threat scenarios and very suitable for presentations. According to one of the participants this type of visualization emphasizes the "message" or the purpose of the analysis.

Described in: *Chapter 9 – Evaluation of the CORAS language (field trials)*.

I) Empirical support for the use of visualization mechanisms:
It is well known that using visualization techniques correctly can improve a graphical representation. Information visualization techniques include among other the use of size, shape and color to make information more accessible to the reader. The study reported in *Appendix A* convinced us to use special risk-related symbols in our language since this improved the subjects' ability to complete tasks. The background for the investigation was related studies which claimed that graphical symbols in modeling languages make the diagrams more understandable. We compared basic UML icons with special risk related symbols. The results showed that one may further improve a UML based language by replacing the standard UML symbols with more specialized symbols that reflects the domain it is used for. An interesting finding in the second study (*Appendix G*) of graphical mechanisms was that people prefer textual information labels rather than just visual means like size, color and shape. We let a group of professionals within system and software engineering make the final design decision with respect to how the diagrams should look like. The results indicate that if using visual means like shape, size or color to convey meaning in diagrams they must be accompanied with an explanatory textual information label.

Described in: *Chapter 9 – Evaluation of the CORAS Language and Guideline (language design decisions), Appendix A – Empirical Investigations of the CORAS Language for Structured Brainstorming, Appendix G – Investigating Preferences in Graphical Risk Modeling*.

J) Results from applying the guideline:
The most complete example of using the guidelines is in the quality evaluation of the CORAS language, where it is used when modeling the core security risk scenarios (*Appendix H*). The description of the scenarios is provided as text only with no guiding or restrictions on to how it should be modeled. A smaller example is given in *Appendix C*. Since the guideline has been developed iteratively and improved for each of the SECURIS field trials, every field trial analysis report contains the results from a full application of the guideline, however these reports are confidential.

Described in: *Appendix H – Quality Evaluation of the CORAS Language, Appendix C – A Graphical Approach to Risk Modeling, Motivated by Empirical Investigations (smaller example)*.

1.3 The Structure of the Thesis

This thesis consists of the following chapters and appendices:

Chapter 1 – Introduction: this chapter provides an introduction to the research domain, the thesis contribution and the structure of the thesis.

Chapter 2 – Problem Area: this chapter introduces the main problems within our area of research, focusing on specification, documentation and communication of security risks. This is also where our success criteria are specified.

Chapter 3 – State of the Art: this chapter introduces the state of the art within security analysis methods and standards, graphical modeling techniques and information visualization mechanisms.

Chapter 4 – Research Method: this chapter describes the research methods we have used.

Chapter 5 – Initial Investigations: this chapter contains the finding from the initial empirical investigations made of security risk analysis terminology and the symbols in the previous version of the CORAS language. This is based on III and IV in page iii, and a full version can be found in *Appendices B* and *C*.

Chapter 6 – Language Requirements defined within a Quality Framework: this chapter gives the detailed requirements to an ideal security risk modeling language inspired by quality requirements for modeling languages within system development and also based on experiences from real security analyses in field trials.

Chapter 7 – Evaluation of the UML profile: this chapter presents a summary of the quality evaluation of the UML profile which was the starting point for our work on a graphical security risk modeling language. A full version can be found in [57].

Chapter 8 – The CORAS Security Risk Modeling Language and Guideline: this chapter presents the syntax and semantics of the language, including a detailed example driven modeling guideline. This is particularly based on II, IV and V, and the full versions can be found in *Appendices C, E,* and *F*.

Chapter 9 – Evaluation of the CORAS Language and the Guideline: this chapter summarizes the evaluations of the CORAS language using questionnaires, in field trials and applying the quality framework.

Chapter 10 – Discussion: the final chapter discusses our contribution with respect to the success criteria from *Chapter 2*.

References, Table of Figures and *List of Tables*.

Appendix A – Empirical Investigations of the CORAS Language for Structured Brainstorming. This is an initial exploration of the understanding of risk analysis concepts. It also includes the experiment with special graphical symbols versus traditional UML symbols. Corresponding to I in page iii.

Appendix B – Risk Analysis Terminology for IT systems: Does it match intuition? This technical report presents the investigation of the understanding of the revised conceptual foundation. Paper II in page iii corresponds to this report.

Appendix C – A Graphical Approach to Risk Modeling, Motivated by Empirical Investigations. This paper provides an introduction to the complete CORAS language, including the conceptual foundation and the guideline. Corresponds to III in page iii.

Appendix D – Structured Semantics for the CORAS Security Risk Modelling Language. This technical report defines the structured semantics for the CORAS language. Corresponds to IV in page iii.

Appendix E – Model-based Security Analysis in Seven Steps – a Guided Tour to The CORAS Method. An example of a complete security analysis, including modeling guidelines for the CORAS language. Corresponds to V in page iii.

Appendix F – Assessing Enterprise Risk Level: The CORAS Approach. An example of a full security analysis, using the previous version of the CORAS symbols. Corresponds to VI in page iii.

Appendix G – Investigating Preferences in Graphical Risk Modeling. Reports the design decisions based on the findings from the experiment. Corresponds to VII in page iii.

Appendix H – Quality Evaluation of the CORAS language: contains the evaluation of the CORAS language with respect to its ability to model a set of core security risk scenarios and fulfillment of language quality requirements.

1.4 How to Read the Thesis

According to which artifact you are interested in, we recommend four alternative strategies for reading the thesis:

A) The development of the conceptual foundation is described in *Chapter 1, 2, 4, 5, 9* and *10*.

B) If you are mainly interested in the CORAS language itself, you should read *Chapter 1*, *2, 4, 8, 9* and *10* since they take you through the most essential chapters.

C) For a description of the development of the modeling guideline, including the guideline itself you should read the same chapter as for strategy B.

D) If you like to explore the quality evaluation framework for security risk modeling languages, we recommend *Chapter 6*. The development strategy is described in *Chapter 4* and examples of how we have applied it are found in *Chapter 7* (a summary, for full details we refer to [57]) and in *Chapter 9* (full details in *Appendix H*).

In the figure below, we have illustrated the different strategies for reading the thesis using the letters A, B, C and D. The gray color indicates chapters that are recommended for all strategies.

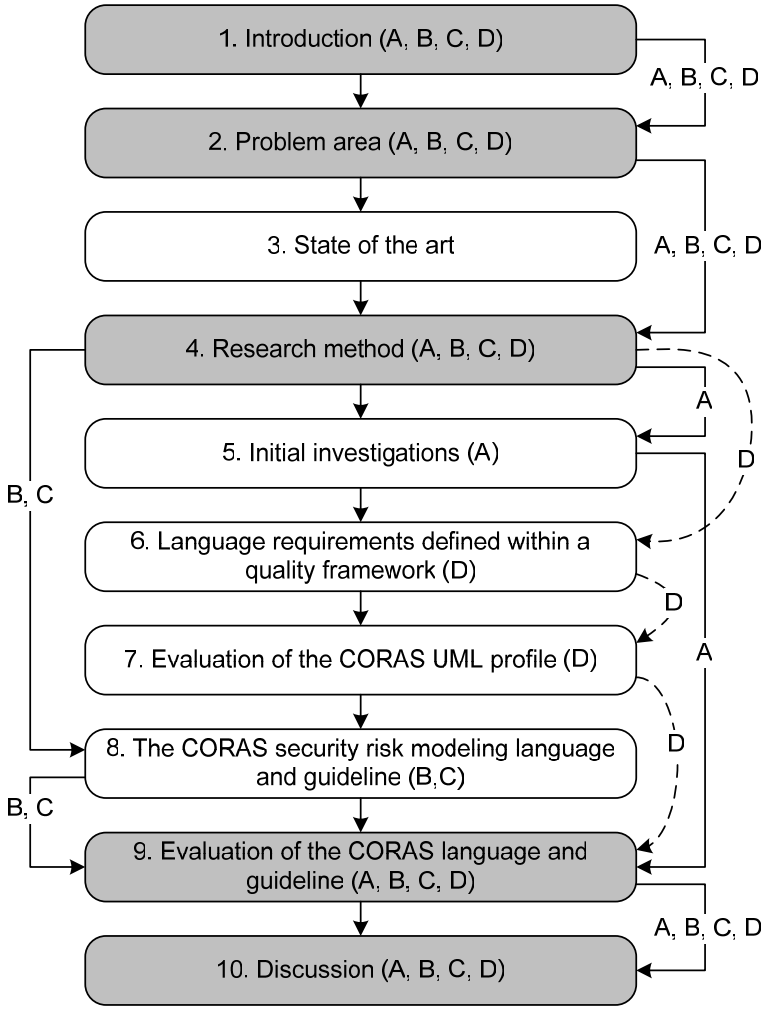


Figure 1 – Reading guide

To avoid confusion while reading, it is important to remember the difference between the UML profile and the CORAS language:

The UML profile	Refers to the *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms* [124] that was developed as part of the CORAS project (2000-2003). The language is based on the use case notation from the *Unified Modeling Language (UML)*.
The CORAS language	Refers to the language presented in *Chapter 8* developed as part of this thesis. The language does not conform to UML, but is inspired of the UML profile and therefore have similarities with the UML use case notation.

2 Problem Area

In this chapter, we first introduce the problem area, and then we formulate and motivate a list of success criteria for the work presented in this thesis.

2.1 Introduction

We all take actions to avoid security risks in our daily life. It may be as simple as locking the office door when leaving for the day, protecting the laptop with a password or avoiding insecure web browsers when accessing the net bank. For your home computer, it is quite manageable to enforce good security routines like updating the virus protection, activating the firewall and keeping updated on relevant security patches. The situation becomes more complex if the system is a major information system, providing business critical information through several interfaces to a large user base. To ensure a sufficient level of security for such a system, there is need for a thorough security risk analysis method.

The security field is rapidly changing and it can be challenging, if not impossible for a single person to keep updated on all relevant security issues for a complex system. This is why a full-scale security risk analysis is typically conducted by a group of people with competence on different areas, for example experts in security technology, system experts, users and decision makers. The idea is that a group of people with different competences will view the system from different perspectives, and therefore identify more and possibly other risks than a more heterogeneous group. When we in the following speak of security risk analysis, we refer to this kind of analysis. It is typically used in cases where security is crucial to an organization, either in their daily business or with respect to the product or service they deliver. The analysis may take hundreds of hours and is consequently an expensive operation. Any means that may contribute to reduced costs, without sacrificing the quality of the security risk analysis results, are more than welcome.

A security risk analysis (or just security analysis) may consist of several "structured brainstorming" sessions. Brainstorming is often associated with a creative process of coming up with new ideas to solve a problem. A structured brainstorming is a methodical and step-wise "walk-through" of the target of analysis, particularly useful for identifying potential security risks and estimating likelihoods and consequences of these risks. The brainstorming session may be structured according to special security aspects, the physical organization of the system, the work processes or similar. In brainstorming sessions within security analyses, the participants are asked to put forward as many threats, vulnerabilities and risks as they may think of within the context of the analysis. These are in turn assessed to find the best way to deal with them. Structured brainstorming may be used to assess systems of various size and complexity, including organizational procedures or routines in which the system is used. This does not mean that the same brainstorming approach is applied *directly* to all types of systems; some adaptation is needed to fit the particular domain.

2.2 The Need for a Special Purpose Language

A security analysis based on structured brainstorming is a team effort that involves a group of people that must communicate and cooperate with each other. The topics discussed may be complex and difficult. A common method for explaining complicated problems is to draw sketches or illustrations on a blackboard. For instance, in software engineering graphical models are often used to ease the explanation of complex systems (e.g. [105]).

The effectiveness of such drawings depends however on the artistic skills of the modeler, the group members' individual interpretations of the drawings and other factors. The potential sources of misunderstandings regarding a drawing are therefore numerous. To reduce this weakness and increase the comprehension and preciseness of security analysis documentation, we see the need for a well-defined and easily understandable graphical notation for documenting security relevant scenarios. Moreover, this notation (or language) must be supported by a carefully designed modeling guideline, which connects the language to the security analysis method. In the remaining of this chapter, we refer to this language and its guideline as the "security risk modeling approach".

Documentation from one step in a security analysis typically provides input to the next step. The security risk modeling approach should also follow this structure. This implies that the different steps in the analysis should be documented using customized diagram types. The diagrams should build on each other like the steps in the security analysis process, meaning that it should be possible to extend the diagrams from one step with more information in the next step. Since the brainstorming session is an interactive process, it must be possible to modify and update diagrams "on-the-fly". This may be very challenging for the modeler, since it is carried out while the participants are watching. Nevertheless, it gives the participants an opportunity to instantly validate and correct the diagrams, possibly providing additional information, which in turn gives a representation that is as complete, and correct as possible.

Threat scenario identification and risk estimation require particular support. Threat scenarios describe all the different ways a risk may occur, something that can be complex and difficult to follow (illustrated in Figure 2).

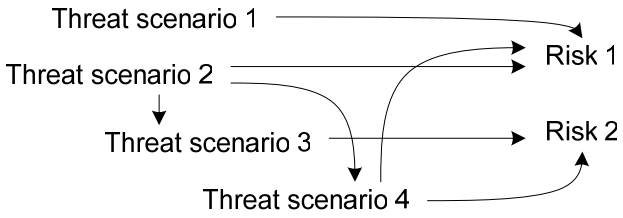


Figure 2 – Risks and threat scenarios

Risk estimation is concerned with determining the likelihoods and the consequences of risks. The likelihood of a risk is based upon the combined likelihood of all threat scenarios leading up to the risk. In theory, this may seem unproblematic, but practice has shown that reliable data to base estimates upon hardly exists. Sometimes statistical material can be used as a starting point or a suggestion, but someone familiar with the target of analysis and the domain must always judge its relevance. Quite often, the participants are unable to estimate the likelihood of a threat scenario, and consequently the likelihood of the risk, in terms of an exact numeric value. A solution may be to use intervals, where the participants only judge in which interval category a likelihood belongs to. This implies that a suitable modeling approach must support likelihood estimation on the basis of interval values, as well as exact, numeric values if they exist.

Security analysis documentation should serve multiple purposes, for example as highlighted in [6, 131]. As already explained, it should support specification of intermediate findings at the various methodological steps. The documentation should also show that the security analysis process has been conducted properly, and provide evidence

of a systematic approach to risk identification and analysis. This may be required in the case of certification according to quality standards like ISO9001 [71] or BS7799-2 [26]. The documentation represents a record of security issues that can be shared with the rest of the organization and thereby extend the organization's knowledge base to become more robust against security threats. Furthermore, often the results from a security analysis should be presented to the management, board of directors or other parties that have not been involved directly in the analysis. The documentation should ideally provide them with a risk management plan, showing the risks, the main vulnerabilities and how they should be mitigated. This highlights of course the importance of having documentation in a form that requires only a minimum of explanation to understand.

Traditional risk analysis documentation has mainly focused on tables and text. This may require the reader to go through an extensive amount of documentation in order to comprehend the security analysis findings. Getting an overview of the findings may be especially challenging for people who only participate in parts of the analysis. Text and tables do not provide sufficient support to the participants during the analysis, since it requires careful attention to everything that is written. When a table grows it may become difficult to keep record of all relevant information and the participants may easily loose track of what has been documented. A graphical documentation approach may provide an overview of the complete risk picture in a more compact and space effective manner than text and tables, which is especially suitable for use in group work. Well established modeling techniques within risk analysis (e.g. fault trees [66] and event trees [64]) are more commonly used for calculating probabilities, than describing the overall risk picture. A new modeling approach must relate to these existing modeling techniques, by either using them as complementary modeling techniques or being able to express the same information. In fact, it may be said that the new modeling approach is an effort to better integrate existing notations. We do not argue for a complete replacement of text, tables and existing notations with something new, but rather build on existing techniques to bring security risk analysis documentation one step further. The intention is to combine traditional risk documentation techniques with state-of-the-art modeling techniques, to support the entire security analysis process.

2.3 Problem Characterization

Above we provided an overview of the overall problem area. In the following, we narrow this down to four main "work packages" addressed by this thesis. For each of them we formulate a set of criteria, providing a characterization of what it means to successfully complete the work package.

2.3.1 A Common Basis for Security Analysis Terminology

As mentioned previously, the participants in a structured brainstorming may have very different backgrounds and competences. The authors of [131] recommend these roles to be represented when assessing IT systems: (1) user or intended user of the system assessed, (2) experts on relevant aspects of the system and (3) system designer. There must also be a person responsible for facilitating the process (analysis leader), and a secretary whose responsibility is to document the findings. To ensure an effective analysis process, the participants must quickly become familiar with the security analysis terminology. This means that the terminology should be as easy to understand as possible. Moreover, this terminology must be reflected in the security risk modeling approach. There are several international standards relevant to risk- and security analysis (see *Chapter 3*). However, these standards have often been made for specific purposes by different standardization

organizations, and may not focus on the same aspects. To help achieving a consistent and clear terminology for use in security analysis, there is a need for a conceptual foundation of security risk related terms, building on the most relevant of these standards. An important prerequisite for the usefulness of such a conceptual foundation is that the definitions are as close as possible to how the concepts are interpreted in everyday language. At the same time, the definitions should not directly conflict with relevant standards. This will facilitate integration with other security analysis methods and avoid misunderstandings among the users. A conceptual foundation defined as a list of definitions is not sufficient. There should also be a description of how the various concepts relate to each other. A concept may have relations to several other concepts, and this must be expressed in a simple manner, preferably illustrated graphically. This motivates the following success criterion:

1. The work of this thesis should include a conceptual foundation for security analysis, that:
    a. specifies the main security analysis concepts and their relationships
    b. uses a terminology that is easy to understand
    c. uses a terminology that is in accordance with international standards
    d. is described in a simple and understandable manner
    e. provides a solid, underlying basis for a security risk modeling approach

2.3.2 A Language for Specifying and Documenting Security Analysis Findings

The need for a customized security risk modeling approach that includes both a language/notation and a set of modeling instructions has already been highlighted. The language should have a well-defined syntax that specifies its components and rules for how they may be combined in diagrams. This will ease the learning, since the models will build on a known set of rules that is recognizable for the reader, even if the actual information modeled in the diagram is unknown. The syntax should make use of best practice within information visualization to help illustrating particular important parts of the security analysis findings. This means that mechanisms like size or color may be used to convey specific meanings in the models. The language should also be designed in interaction with real users to include their preferences, and to ensure that it fulfils their needs. Well-known risk modeling notations should be considered for inclusion in the language. If this is impractical, one should try to include their underlying ideas even though they will be expressed with a different syntax. The language also needs a precise semantics that defines how the diagrams should be understood. The semantics should be structured in the sense that the meaning of an arbitrary, syntactically correct diagram can be generated schematically. Moreover, to make the semantics accessible for regular users of the language we would like the generated meaning of a diagram to be expressed in English. The structured semantics will help ensuring that all readers understand the models in the same manner. This motivates the following success criterion:

2. The work of this thesis should specify a language for describing security risks, that:
    a. is suitable for use in structured brainstorming sessions
    b. is easily understandable for the participants in the brainstorming, including those who receive the analysis results afterwards
    c. has a precise syntax, meaning its design should be based on:
        i. best practice within information visualization
        ii. experiences with realistic security risk scenarios
        iii. users' preferences
        iv. existing risk modeling techniques
    d. has a structured semantics that translates arbitrary diagrams into English

e. supports and documents the different steps in the security analysis process

2.3.3 Modeling Instructions for the User

To utilize a language to its full potential, the person who uses it (typically the one who is responsible for documenting the analysis) must be provided with instructions for how it should be used. This is not only valuable to the modeler, but also important for the quality of the models. A set of instructions for how to model, may help achieving models that are more complete and consequently the analysis will be better documented. The instructions should be in the form of a guideline, which in addition to rich and detailed modeling instructions also provides realistic modeling examples from real security analysis situations. The guideline should first of all address inexperienced modelers and those unfamiliar with security analysis. To give this group as much support as possible, the guideline should if possible include recommendations from practical use of the modeling approach. Since the modeling tasks are tightly coupled to the analysis process, the guideline should follow the same structure, with relevant guidance for each step. This motivates the following success criterion:

3. This thesis work should provide a modeling guideline for the modeler, that:
    a. is rich and detailed with realistic examples
    b. addresses non-experienced modelers
    c. has recommendations based on user experiences
    d. follows the security analysis process step by step

2.3.4 A Way of Evaluating the Quality of Security Risk Modeling Approaches

When developing this security risk modeling approach, the requirements should be properly defined, similarly to a requirement specification for e.g. a software product. The degree of fulfillment of this specification can then be used as a measure of the quality of the modeling approach. A considerably portion of these requirements will however be of a general character and can therefore be seen as a quality assessment framework for any security risk modeling approach. People who need to judge the quality of, or compare, different security risk modeling approaches, may use this framework. There exist frameworks for comparing *security analysis methods* [158] and for evaluating the quality of *modeling languages in general* (Sect. 6.2), but to our knowledge there are none for evaluating the quality of security risk modeling approaches. Such a quality evaluation framework should cover model quality in general, and aspects that are specific to security risk analysis. Examples of the latter are the strong relationship between the different steps in the analysis process and the modeling approach, the special terminology, and the focus on structured brainstorming. The framework should also have a modeling test case that may be used as a "benchmarking test" to evaluate the approach's ability to model typical and realistic security risk scenarios. This motivates the following success criterion:

4. This thesis work should provide a framework for evaluating the quality of security risk modeling approaches, that:
    a. covers quality requirements related to modeling languages in general
    b. includes quality requirements special to security risk analysis
    c. includes a modeling test case representing core security risk scenarios

3 State of the Art

This state of the art is divided into 6 sub sections: (1) risk management, (2) security risk analysis methods, (3) standards relevant to security- and risk analysis, (4) risk- and system analysis techniques, (5) risk documentation techniques and (6) information visualization techniques.

3.1 Risk Management

Domains like finance, safety/reliability, and security all use the term risk. Even though our work addresses security risk analysis, it will cover aspects of the safety/reliability domain since we use or build on their methods, terminology and more. When speaking of approaches, methods or techniques that do not belong to a particular domain we will use the terms risk analysis, risk analysis technique etc. If the focus is particularly on security, we will use "security risk analysis" or just "security analysis". Financial risk analysis will not be covered by this thesis.

It may be difficult for the reader to clearly understand the differences between risk management, risk analysis, risk analysis techniques etc. The problem lies in the lack of precise definitions. Something presented as a method in one context may be regarded a framework, methodology or technique in another. To make this easier we will use the following distinctions and classifications in this state of the art (framework and methodology fall within method):

- **Risk management** is the culture, processes and structures that are directed towards realizing potential opportunities whilst managing adverse effects [6]. In practice, this means the use of one or more methods to identify and evaluate new risks, and also monitor, review, and communicate information about existing risks.
- **Risk analysis method/security risk analysis method** refers to a method used in the systematic process of understanding the nature of and deducing the level of risk [6]. In practice, this refers to the overall assessment of a system, or part of a system, that includes risk identification and evaluation, and also suggestions for risk mitigation. This kind of method often makes use of one or more risk analysis techniques.
- **Risk analysis technique/security analysis technique** refers in this thesis to analysis techniques that may be used in various parts of a risk analysis method. These techniques are typically used to e.g. estimate the likelihood of risks, analyze potential outcomes of a risk, look for particular vulnerabilities, and more.

The domain, objective and the level of detail of an analysis influences which approach for managing and controlling risks that is appropriate to use. Often it is necessary to tailor a risk management approach or strategy to meet specific usage requirements. The two main strategies within risk management are *asset-based* and *threat- and vulnerability-based* risk management [133]. The asset-based risk management focuses on identifying the valuable aspects of the system (assets), and then assesses how they may be protected from threats and risks. Asset-based risk management typically involves so-called bottom-up risk analysis techniques like FMEA/FMECA (Failure Mode Effect Analysis / Failure Mode Effect and Criticality Analysis) (Sect. 3.4.2) and HazOp (Hazard and Operability) analysis (Sect. 3.4.3). Threat- and vulnerability-based risk management aims to identify the threats and vulnerabilities of a system first, and then look at the risk they pose in a top-down manner. An example of a threat and vulnerability-based risk analysis technique is SWOT (Strengths, Weaknesses, Opportunities and Threats) which helps identifying the general

areas of concern regarding a system (Sect. 3.4.1).The two categories of risk management have their strengths and weaknesses and the optimal solution is probably to use a combination of them. Often a top-down threat and vulnerability assessment is conducted initially to establish the scope of the analysis and characterize the overall risk picture. Then a more detailed bottom-up approach can be used to identify assets and their risks, using the findings from the top-down approach.

3.2 Security Risk Analysis Methods

In this section we first present the starting point of our work, the CORAS security analysis method (3.2.1), and then other relevant security analysis methods. We survey the main differences and similarities between these methods and CORAS, and summarize this in the end of the section. One important common factor for these methods is that they employ some form of structured brainstorming in the identification, estimation, evaluation or treatment of risks.

3.2.1 The CORAS Security Analysis Method

The security analysis method CORAS (http://coras.sourceforge.net) was developed in the EU-funded CORAS project (IST-2000-25031) from 1999 to 2003. The project had eleven partners representing the United Kingdom, Greece, Germany and Norway. One of the aims of the project was to gather well-known risk analysis techniques into an integrated security risk analysis method called the CORAS security analysis method. The method should make it easier to apply several risk analysis techniques in an integrated manner with particular focus on security risks analysis.

The CORAS method employs several international standards [7, 65, 66, 76, 82, 86] and the process follows the Australian / New Zealand Standard for Risk Management [7] (Sect. 3.3.1). This CORAS method has five main phases: *(1) establish context, (2) identify risks, (3) analyze risks, (4) evaluate risks and (5) treat risks.* Each phase has different purposes and produces different types of documentation. Structured brainstorming is used in risk identification, risk estimation and in the risk treatment phases.

Another major result from the CORAS project was the graphical security risk modeling language based on UML, the UML profile (Sect. 3.5.1). In addition to the methodological approach and the graphical language, the CORAS project resulted in a library of reusable experience packages, a computerized integration tool, an XML mark-up for exchange of risk assessment data and a vulnerability assessment report format (Figure 3), but these are not relevant to this thesis work. The work with the CORAS method has been continued within the research project SECURIS (NFR 152839/220) which runs from 2003 to 2007. One of the main objectives of SECURIS is to improve the security analysis method by applying it in industrial field trials.

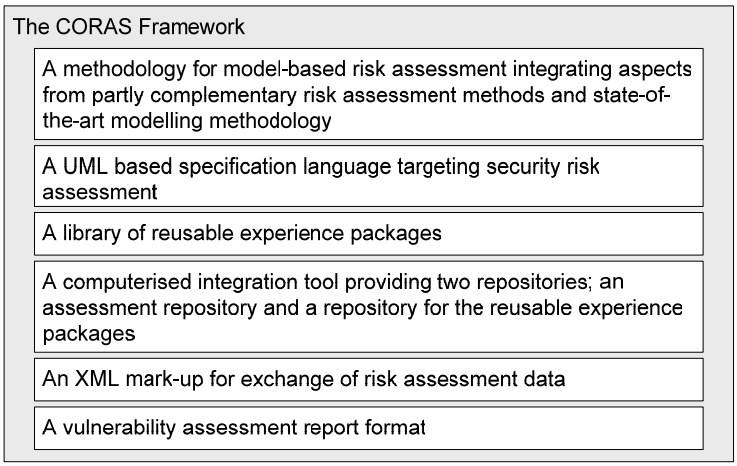


Figure 3 – Results from the CORAS project

3.2.2 OCTAVE

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [3] is a risk-based strategic assessment and planning technique for security. OCTAVE is conducted in three phases: (1) identify critical assets and the threats to those assets, (2) identify the vulnerabilities that expose the assets to threats, and (3) develop an appropriate treatment strategy (Figure 4).

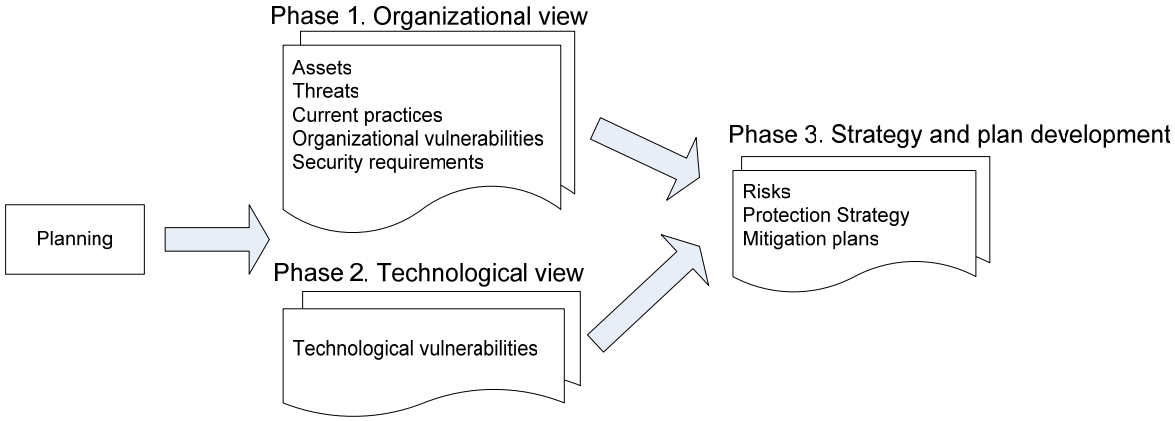


Figure 4 – The OCTAVE process

Phase 1 normally involves two workshops, the first with senior management to define the scope of the analysis, and the second with staff that have a more technical expertise on the target of analysis. The workshops may have a form of a structured brainstorming where people with different competences and backgrounds participate. The intermediate findings are documented in tables and form the basis for developing asset-based threat profiles using a simple graphical tree-structure (described in Sect. 3.4.8). The approach in OCTAVE is quite similar to the one used in CORAS.

3.2.3 CRAMM

CRAMM (CCTA Risk analysis and Management Method), is the UK Government's Risk Analysis and Management Method [8]. It is owned by the UK government's Security Service, but managed by Siemens/Insight (http://www.cramm.com). In CRAMM, *risk analysis* is identification and assessment of security risks while *risk management* is concerned with identifying appropriate countermeasures, or treatments for those risks. In

our work the term security analysis covers the complete process from risk identification to treatment identification. Risk management according to CRAMM includes three phases: (1) asset identification and valuation (including dependencies between assets), (2) threat and vulnerability identification and (3) treatment (countermeasure) identification (Figure 5). The information is gathered through interviewing the owners of the assets, the users of the system, the technical support staff, and the security manager. In this manner, CRAMM is more like a review of the security of a product, conducted during system development or for an already running system.

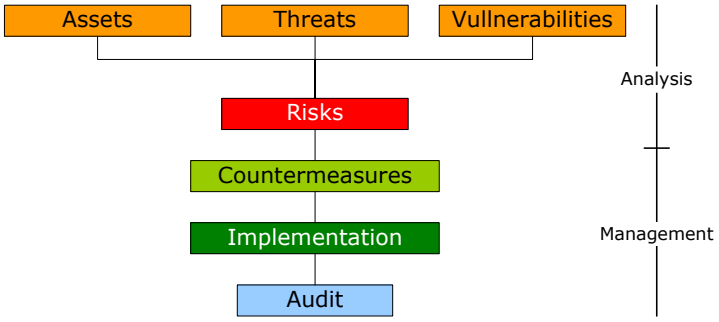


Figure 5 – The CRAMM process

The documentation produced during a CRAMM review uses a standardized CRAMM format, mostly in the form of specialized tables. CRAMM may help an organization to achieve compliance with ISO17799 [82], and the outcome is compliant with the mandatory documentation needed to achieve ISO27001 certification (BS7799-2) [26, 75]. The concepts and activities in CRAMM were a source of inspiration in the CORAS project.

3.2.4 Facilitated Risk Assessment Process

Peltier Associates (http://www.peltierassociates.com) develops the Facilitated Risk Assessment Process (FRAP). A risk assessment according to FRAP focuses on security aspects of systems or business processes. The assessment team consists of representatives with competence on technical aspects, as well as business- and management aspects. The FRAP has focus on threats, vulnerabilities and consequences towards data integrity, confidentiality and availability. A FRAP consists of the following three phases:

- Pre-FRAP meeting: the objective of this meeting is to decide on the system description and scope of the assessment, as well as assembling an assessment team.
- The FRAP session: this phase consists of three activities; first one decides the roles each participant will have in the brainstorming session, reviews and agrees on the definitions and scope of the risk assessment. Second, one conducts the actual brainstorming to identify potential risks within the scope of the assessment, and third one prioritizes the identified risks according to how vulnerable the system is, and what impact the risks may have. When the risks are sufficiently specified, the participants may also suggest possible controls or treatments for the risks.
- The post-FRAP meeting(s): these meetings aim to further analyze the information gathered at the FRAP session. The outcome is an overview of risks and how they should be mitigated by existing or new controls (treatments). The final report contains a complete documentation of the process, including an action plan for the recommended treatments.

3.2.5 Microsoft's Security Risk Management

As the name indicates, the Microsoft security risk management process [120] includes more than just a risk analysis method. The process consists of four phases (Figure 6, from [120]), where the first and the second correspond to our interpretation of a risk analysis method.

- 1st phase - assessing risk: during this phase one gathers data about assets, threats, vulnerabilities, existing security controls and suggested treatments. This information is then analyzed in facilitated discussions (what we call structured brainstorming sessions) and the outcome should be a list of risks.
- 2nd phase - conducting decision support: the list of risks from the previous phase function as input to an assessment of the various control or treatment solutions that are proposed. The outcome of this phase is a set of treatment options that are considered to be appropriate for mitigating the risks.
- 3rd phase - implementing controls: the decided risk treatments are implemented.
- 4th phase - measuring program effectiveness: in this phase the implemented treatments are monitored to verify their effectiveness. This phase also covers the ongoing process of watching out for new, potential risks.

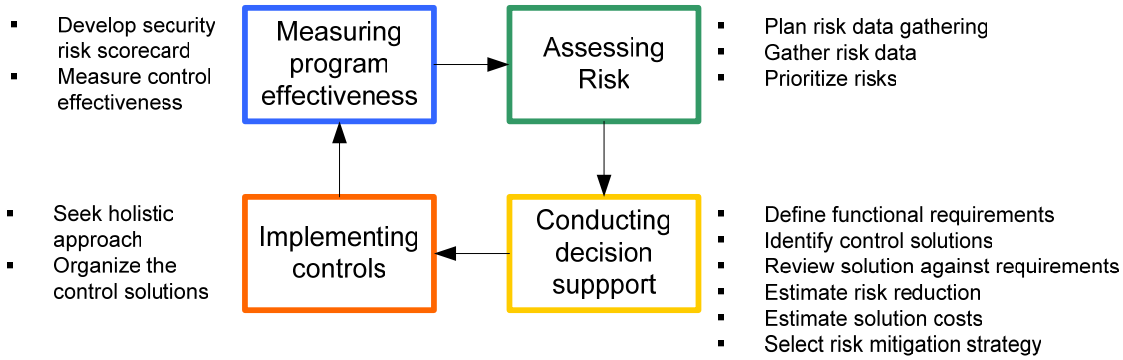


Figure 6 – Microsoft's security risk management process

3.2.6 Other Methods

We also include some security related methods and framework that are worth mentioning.

3.2.6.1 COBIT

ISACA (http://www.isaca.org) develops *COBIT* (Control Objectives in IT) that started as a guide on best practice IT management controls, intended for computer auditors. COBIT is a framework that focuses on assisting managers in the implementation and quality control of IT processes and systems in their organization, and not so much a risk analysis method according to our definition. To obtain the latest version of the COBIT (v4) documentation you must be a member of ISACA.

3.2.6.2 FIRM

The Information Security Forum (ISF) is an international association of more than 260 leading companies and public sector organizations (http://www.securityforum.org). ISF provides a set of methods and tools related to risk analysis and risk management. Among others they have a methodology for the monitoring and control of information risk at the enterprise level called *FIRM*. *SARA* is their risk analysis method for critical information systems, and *SPRINT* is a method for less critical systems and assessment of business risks. The methods are however only available through membership in ISF.

3.2.6.3 MEHARI

MEHARI (Méthode Harmonisée d'Analyse de Risques Informatiques) [119] is a French risk analysis method, designed by CLUSIF (https://www.clusif.asso.fr/en/clusif/present/), an organization consisting of security experts. MEHARI provides a method, tools and knowledge bases for use in controlling the risks of an organization. The method is compliant with ISO/IEC13335 (see Sect. 3.3.2) and its results can be used as input to ISO27001/BS7799-2 certification (see Sect. 3.3.3). MEHARI is not well known outside France and can only be purchased from CLUSIF.

3.2.7 Summary

This section summarizes the relations between CORAS and other security analysis methods.

Method	Compared to CORAS
OCTAVE	Similar process, similar vocabulary. Comes with more specific guidelines and predefined templates for documentation than CORAS. Limited use of graphical risk modeling.
CRAMM	Similar process, similar vocabulary. More focused on certification and the requirements needed for this purpose than CORAS. No graphical risk modeling.
FRAP	Similar process, similar vocabulary. Makes no particular use of graphical modeling.
Microsoft's Security Risk Management	Covers the entire risk management process, where the two first phases constitute a method according to our definition. The analysis process is similar to CORAS, but it does not use graphical modeling.
Other methods	COBIT is internationally well known, but it is more an audit framework than a security analysis method. FIRM, SARA, SPRINT and MEHARI are not widely used outside their organizations countries, and therefore of less interest. We have however no indications that any of these methods are fundamentally different in their risk analysis process than CORAS.

3.3 Standards Relevant to Security- and Risk Analysis

There exist several international standards relevant to security and risk analysis. In this section, we give a presentation of the ones relevant to our work, and how they relate to the conceptual foundation in CORAS. The section concludes with a short summary of each of the standards' relations to CORAS.

3.3.1 AS/NZS4360

The Australian New Zealand Standard for Risk Management (AS/NZS4360) [6, 7] defines a complete risk management process. It defines risk management as *the total process of identifying, controlling, and eliminating or minimizing uncertain events that may affect IT system resources*. The standard is accompanied with a handbook that gives a "generic guide for the establishment and implementation of a risk management process for information security risks" [54, 55]. The process has five main phases (Figure 7): *(1) establish context, (2) identify risks, (3) analyze risks, (4) evaluate risks and (5) treat risks*:

1. When establishing the context, the purpose is to decide what parts of the system that will receive attention, and why. This phase also aims to specify the stakeholders and vulnerabilities of the system.

Risk assessment is the collective term for identification, analysis and evaluation of risks. These three phases have the following purpose:

2. During risk identification, one should identify potential threats and threat scenarios that can constitute risks towards the system. A commonly used risk identification technique is some form for structured brainstorming.
3. The purpose of risk estimation is to estimate the risks' likelihoods and potential consequences. This activity is mostly centred on applying statistical methods or more qualitative judgments to establish the likelihood and consequences of the risks, and modeling notations like fault trees [66] and event trees [64] may be used.
4. Risk evaluation includes prioritization of the risks according to their severity and selecting the ones that cannot be tolerated which will be subject to treatment identification.
5. In the risk treatment phase the purpose is to identify the risks that need treatments, i.e. risks that are too serious to leave unattended.

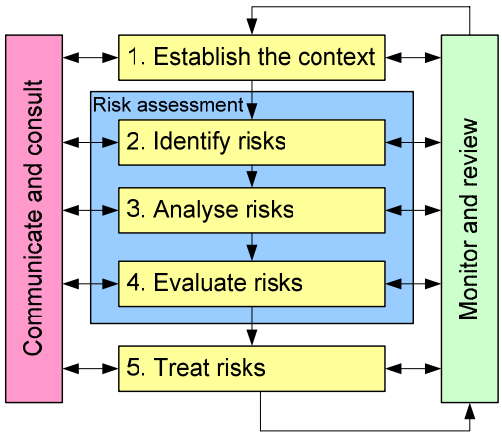


Figure 7 – Risk management process overview

The standard also defines two parallel activities *communicate and consult*, and *monitor and review*, which are related to the continuous risk management effort within an organization. CORAS takes much of its terminology from AS/NZS4360 and consequently the conceptual foundation is to a high degree in accordance with this standard.

3.3.2 ISO/IEC13335

The standard *ISO/IEC13335 Information technology* [76-79] provides help and guidance on how to manage security of information systems. The standard consists of five parts. *Part 1- Concepts and models for information and communications technology security management* [76] addresses security and security management in and organizational view, providing guidance on planning, implementation and operation of security measures. *Part 2- Information security risk management* has been revised by Part 1. *Part 3- Techniques for the management of IT Security* [77], *Part 4- Selection of safeguards* [78] and *Part 5- Management guidance on network security* [79] are more technical security standards. ISO/IEC13335 has in its latest revision been adapted to align with AS/NZS4360 and it is expected that several of its parts will be released under the new standard: Information Security Management (ISO/IEC 27000-family). Since ISO/IEC13335 has become aligned with AS/NZS4360, there is a substantial agreement between the conceptual foundation in CORAS and the terminology used in the ISO standard.

3.3.3 BS7799/ISO17799

BS7799 [25] was published in 1995 by the British Standards Institution, as a standard to guide the development and implementation of information security management system. It was a so-called *code of practice*, which means that it provides help and guidance on the selection and implementation of controls, but cannot be used in third party verification or certification of products. The demand for such a certification option lead to the development of *BS7799-2 (ISO27001) Information technology – Security techniques, Information security management systems – Requirements* [26, 75] which has requirements specified in the form "shall". In addition it integrates the process-based approach of *ISO9001 Quality management systems – Requirements* [71] and *ISO14001 Environmental management systems – Requirements with guidance for use* [72]. The original code of practice is now known as *BS7799-1 (ISO17799)* [25, 82]. Even though the standard has a broad international accept, it does not define concepts in the level of details like the CORAS conceptual model. It is more like a high-level guideline for security risk management.

3.3.4 ISO15408 (The Common Criteria)

The Common Criteria (ISO15408) [33, 73] gives a set of security requirements a product or system must conform to in order to match a specific level of security. This makes it possible for a vendor to explicitly state the level of security he/she require from a system or product. The requirements are divided into functional requirements and assurance requirements. The functional requirements are security aspects that a product must conform to. Assurance requirements are used to evaluate whether the security aspects of a product or system work and whether these are implemented correctly. There are seven assurance levels, each with a stricter requirement to how the product or system should be evaluated:

1. EAL1: the product must be functionally tested
2. EAL2: the product must be structurally tested
3. EAL3: the product must be methodically tested and checked
4. EAL4: the product must be methodically designed, tested and reviewed
5. EAL5: the product must be semiformally designed and tested
6. EAL6: the product must have semiformally verified design and be semiformally tested
7. EAL7: the product must have formally verified design and be formally tested

Although the Common Criteria does not come with an implementation method, various attempts have been made to specify a method that supports integration of its security aspects into the software engineering process [157]. The common criteria evaluation is conducted by authorized entities as an assessment of the product or system against the defined criteria. This is more like a security audit than a complete security analysis process, and focuses on specification and testing of the product or system. Except for "asset" the standard does not use any of the risk related concepts in CORAS.

3.3.5 IEC61508

IEC61508 *Functional safety of electrical/electronic/programmable electronic safety-related systems* [69] is the international standard for all safety lifecycle activities for systems comprised of electrical and/or electronic and/or programmable electronic components (electrical/electronic/programmable electronic systems (E/E/PESs). The safety lifecycle of a system covers all safety related activities made in phases from concept, through design, implementation, operation, maintenance until decommissioning. Typical

examples of E/E/PESs are control and supervisory systems used in domains like industry, health and transportation. The key features of IEC61508 can be summarized like this [15]:

1. It facilitates development of sector specific standards for safety related E/E/PESs that builds on the same underlying principles, terminology etc.
2. It helps specifying safety requirements for safety related E/E/PESs.
3. It introduces safety integrity levels (SIL) that are used to specify the level of safety required by a function in a safety related E/E/PESs.
4. It has a risk-based approach to determine SIL requirements.
5. It sets numerical target failure measures for safety related E/E/PESs that are linked to the SIL's.
6. It sets a lower limit on the target failure measures that can be claimed for a single safety related E/E/PES.

Compared to the concepts in CORAS, IEC61508 concepts are much more system oriented with terms like safety function, module, safe state etc. The standard does not focus on traditional security oriented aspects like risks posed by someone/something with malicious intents like hackers, intruders or malicious code. The lack of such concepts makes it less comparable to a conceptual model for security risk terminology like in CORAS.

3.3.6 ISO21827

The standard *Information technology - Systems Security Engineering - Capability Maturity Model (ISO21827)* [74] defines a complete development process for security engineering. Its concepts and definition are mainly taken from ISOIEC13335, except for "threat" which is defined as *Capabilities, intentions and attack methods of adversaries, or any circumstance or event, whether originating externally or internally, that has the potential to cause harm to information or a program or system or cause those to harm others*. Nevertheless, this interpretation of threat does not conflict with the more general definition used in the CORAS method: *a potential cause of an unwanted incident*. Due to its near coupling with ISO/IEC13335, which is already used in CORAS' method and underlying foundation, the terminology used in ISO21827 is considered to be sufficiently covered.

3.3.7 Other Standards and Guidelines

The standards *Information technology – Security techniques – Guidelines for the use and management of Trusted Third Party services (ISO/IEC14516)* [80] and *Information technology – Security techniques – IT Network security (ISO/IEC18028)* [83] are standards targeting technical security solutions in a detailed level with no focus on security analysis as a process. Their objectives are to extend the security management guidelines provided in ISO/IEC13335 and ISO/IEC17799.

Information technology – Security techniques – Information security incident management (*ISO/IEC TR 18044)* [85] provides advice and guidance on information security incident management for information security managers and for information system managers. The standard can be seen as a supplement to the security risk management process and not a stand-alone security analysis standard.

Information technology – Software life cycle processes – Risk management (ISO/IEC16085) [63] defines a process for the management of risk during a complete software lifecycle process with no particular focus on security. This includes risks related to both buying and supplying software, as well as developing, operating and maintaining software. It may be used to guide standards like *Software life cycle processes (ISO12207*)

(a high-level standard addressing all processes of the software life cycle) or *Systems Engineering - System Life Cycle Processes* (*ISO15288)* (a framework for describing the life cycle of systems created by humans).

There are also quality management standards that organizations may be audited against to prove their compliance with specific quality requirements. These includes among others: the *ISO9000 quality management family* [71], *IC9700* and *IC9200 business certifications* (www.icharter.org), and *ISO20000* for IT service management [84]. None of these is targeting risk analysis or security analysis.

The U.S. National Institute of Standards and Technology publishes standards and best-practice guidelines for a wide range of IT security related topics. The *NIST SP800-30 Risk Management Guide for Information Technology Systems* [123] provides *a foundation for the development of an effective risk management program, containing both the definitions and the practical guidance necessary for assessing and mitigating risks identified within IT systems*. The publication is therefore more like a guideline than a standard and in a comparison with OCTAVE the authors claim [162] "following the OCTAVE guidance will meet the spirit and intent of the NIST guidance for conducting the risk assessment as part of a total risk management program described in NIST SP 800-30"

3.3.8 Summary

This section summarizes the state-of-the-art with respect to standards for security and risk analysis that are relevant to the conceptual model for security risk analysis defined in CORAS.

Standard	Relation to CORAS
AS/NZS4360	CORAS employs many of the concepts in this standard in its underlying foundation.
ISO/IEC13335	CORAS employs many of the concepts in this standard in its underlying foundation.
BS7799 and related standards	Has a stronger relation to the CORAS method than the conceptual foundation, since it deals with risk management without detailed risk analysis concept definitions.
Common Criteria	The standard does not provide as detailed concept definitions like the one used in CORAS' conceptual model.
IEC61508	Employs much more system oriented concepts than the security focused conceptual model in CORAS.
ISO21827	Sufficiently covered by CORAS, since it is based on ISO/IEC13335.
Other standards	The other standards and guidelines are either specialized towards particular security technologies, too high-level or lacks focus on security analysis.

3.4 Risk- and System Analysis Techniques

Traditionally risk analysis techniques document their findings in tables only, but our claim is that many of them may benefit from using graphical documentation techniques as well. In this section, we present risk analysis techniques that have in common that they may be used in structured brainstorming, and we evaluate whether they are appropriate for graphical risk documentation techniques. The evaluation is summarized in Sect. 3.4.13.

3.4.1 SWOT

A *SWOT* analysis [62] is used to evaluate the Strengths, Weaknesses, Opportunities, and Threats associated with a project or business activity in a top-down manner. Used in project or business planning the objective is to define a goal, strategy or actions, and find means to achieve it (e.g. increase income, reduce development time). In security analysis it is used to get an initial overview of the risk picture and helps scoping the analysis to focus where it is most needed. In a security analysis the objective of a SWOT will be to protect the assets within the target of analysis and then find:

1. Strengths: attributes of the target that are helpful in protecting assets.
2. Weaknesses: attributes of the target that are harmful to achieve sufficient protection of the assets.
3. Opportunities: external conditions that are helpful in protecting assets.
4. Threats: external conditions that are harmful to achieve sufficient protection of the assets.

A SWOT analysis takes form as a brainstorming session involving a cross-functional team consisting of people with different background and view of the target system. The information they come up with may be used as input to a more detailed security analysis. The findings are typically documented in various forms of table formats, adjusted to the particular need of the analysis, but often structured according to the four SWOT aspects. A security risk focused SWOT analysis may benefit considerably from using an easily understandable graphical approach to model its findings. This will for instance make it possible to illustrate how a threat in one area may exploit a weakness in another, or how an opportunity may become a weakness if seen from a different point of view.

3.4.2 FMEA/FMECA

FMEA/ FMECA (Failure Mode Effect Analysis / Failure Mode Effect and Criticality Analysis) [17] is a method that assess potential failures of individual components within a system. The method is usually conducted in two steps, first the failure modes and their effects are identified (FMEA). Then the failure modes are ranked according to their criticality and their probability (FMECA). The basis of the FMEA/FMECA is functional description of the system, where each component is analyzed to identify all possible or failure modes and classify them according to their criticality. The FMEA/FMECA is a "bottom-up" approach, especially suitable for detecting a system's possible failure modes, and determining their consequences. The failure identification is normally organized as a brainstorming, structured by the system's functional descriptions. The findings are documented in a table where each separate module's potential failure modes are investigated with respect to failure detection method, failure effect and how critical it may be. It is not obvious how to document relations between failure modes in different modules, neither how the effects may be common for several modules. This is a common problem of tables, where relations between different rows are difficult to show. In this regard, a graphical language may be useful to document relations between the findings instead of, or in addition to the conventional FMEA/FMECA tables.

3.4.3 HazOp

HazOp (Hazard and Operability) analysis [70] is a well known risk identification technique used in all forms for risk analyses. A HazOp is a structured brainstorming with the aim of finding ways system behavior may deviate from design intention, and whether threes deviation can lead to unwanted incidents (hazards). The participant all must have through knowledge of one or more aspects of the system analyzed. The input to the analysis is

system documentation of any kind, and in addition the analysis leader uses specialized guidewords to ensure that all aspects are covered. The guidewords are used in questions like "what if the service delivers too much data?", "what if the service delivers too little data?", "too slow response or too early?" and so on. This is meant to mitigate the weakness that the information gathered during a HazOp is restricted to the already existing knowledge within the group. The idea is that the guidewords can make people think of aspects they have not been thinking of before. A similar technique that is commonly used within safety analysis is called *HazId* (Hazard Identification). This is basically a simplified HazOp that uses checklists rather than guidewords, and it is often used early in the analysis process or for smaller risk analyses. HazOp can be tailored to fit any domain and system, in for instance [131] the method is especially targeting software HazOp. These kinds of methods represent particular suitable situations for using graphical security risk modeling languages since a common understanding and communication between the participants are crucial to the quality of the findings. Similar to FMEA/FMECA tables, also HazOp tables are unsatisfactory for showing relationships between the findings (the different rows in the table). There is clearly a need for both documentation methods, where one keeps detailed information about each risk in tables while the relationships between the risks are documented graphically.

3.4.4 Fault Tree

The *fault tree* notation is used in fault tree analysis (FTA) [66] to describe the causes of an event. Fault trees are well known and widely used within risk analysis, and become more commonly in security analysis, typically of systems that may have consequences for safety. The notation provides an excellent way of structuring the order of events, and is particular useful if there exist numerical statistical data to use in calculations. Fault trees may among others be used to model the findings of HazOp analyses [152]. The top node represents an unwanted incident, or failure, and the different events that can lead to the top event are modeled as intermediate nodes or leaf nodes (Figure 8, taken from [133]). The probability of the top node is calculated based on the probability of the leaf nodes and the logical gates "and" and "or".

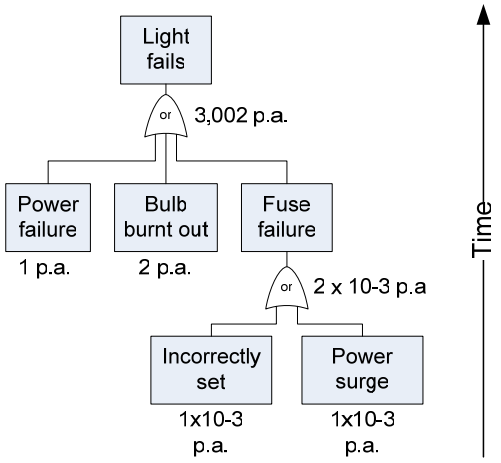


Figure 8 – Fault tree example

Fault trees can be used both qualitatively to specify the different paths that lead to the unwanted incident, as well as quantitatively to estimate the likelihood of the top node incident [2]. The leaf nodes in a fault tree must be independent of each others, otherwise one has to apply special methods for computing likelihood values [107]. An incident model that takes the fault tree notation a step further into a more complex structure is the MORT

(Management Oversight and Risk Tree) [87] which is more common within safety risk modeling. There exist specialized methods for quantitative analysis of fault trees (e.g. [43]) and also methods that takes into account uncertainty regarding the likelihood estimates (e.g. [92]), but these topics are beyond the scope of this thesis.

The modeling notation used in FTA is quite easy to understand and particularly useful for systems consisting of hardware/software modules. Whenever the system also includes people's behavior, the notation becomes too "rigid". It is not feasible to set numerical fault rates for humans in the same manner as for e.g. hardware components. FTA does cover the outcome of the unwanted incident and provides therefore only one side of the risk picture. The underlying idea of illustrating the chain of events that may lead to a fault or unwanted incident is however very useful, and can be used as a basis for a modeling language that is less strict than the fault tree. The

3.4.5 Event Tree

Event trees [64] use a tree notation to represent the outcome (or consequences) from an event and the probability of the various consequences (Figure 9, example from [133]). In the same manner as a fault tree, also the event tree is both qualitative (shows the outcomes from and event) and quantitative (estimates the likelihood of each outcome). When constructing an event tree it is normal to use a binary split from the initial event, towards the final consequences (success/failure). Event trees can be used as an extension of fault trees to create a "cause consequence" diagram (Figure 10). The event tree lets the modeler specify the "barriers", or the mechanisms that shall prevent the consequences of an unwanted incident from escalating. It also describes what the outcome will be if the barriers fail to work.

The event tree lets you specify every detail about the expected outcome from an unwanted incident. It also includes the barriers that should be in place to minimize the consequences. Similar to the fault tree, also event trees provides half the risk picture, excluding the chain of events that may lead to the incident. As one may see from the small example in Figure 9, the tree will grow rapidly when the number of barriers is high, and it does not allow for showing how a failure in a barrier may initiate a new unwanted incident. Nevertheless, the underlying idea of event trees is very valuable, but there is room for an improved and possible more flexible notation.

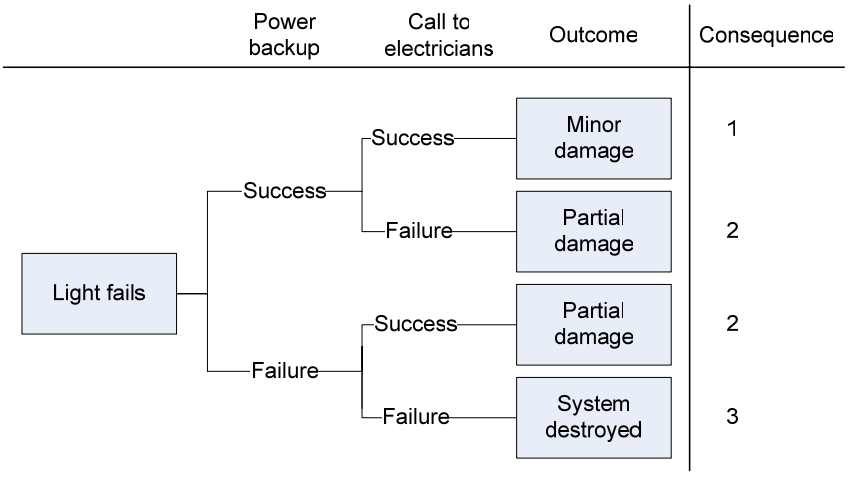


Figure 9 – Event tree example

3.4.6 Cause-consequence Diagram

The *cause-consequence diagram* [122] combines the features of both fault tree and event tree. When constructing a cause-consequence diagram one starts with an unwanted incident and develops backwards to find its causes (fault tree) and forwards to find its consequences (event tree). Figure 10 shows and example of a cause-consequence diagram which is a combination of the two previous diagrams (Figure 8 and Figure 9).

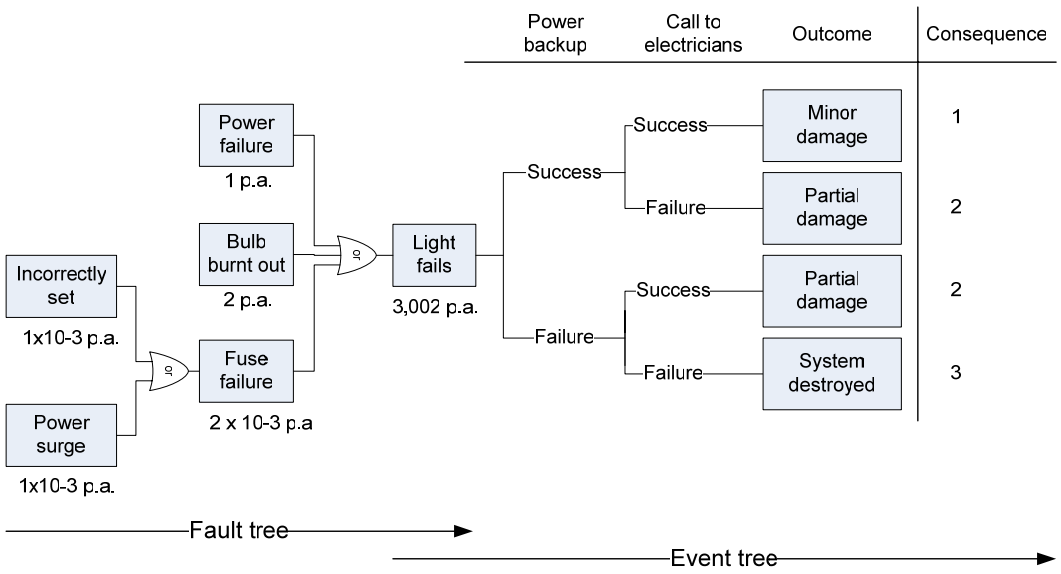


Figure 10 – "Cause consequence" diagram

The cause-consequence diagram provides the complete risk picture, which both FTA and ETA lacks. It illustrates the chain of events from the very beginning with the initiators of unwanted incidents, to their final consequences towards assets. Since it builds on FTA and ETA it also inherits their weaknesses (as discussed in the two previous sections), but it captures much of the main idea behind what we consider as the ideal way of presenting a risk picture. The notation should be optimized with respect to presentation, but it should also be able to model the risk picture in a high level manner, without demanding all details about the chain of events (order of events, probabilities, logical gates etc.). For instance, detailing each path through the diagram may be left for subsequent analyses.

3.4.7 Attack Tree

Attack trees [138, 139] is a modeling notation that aims to provide a formal and methodical way of describing the security of a system based on the attacks it may be exposed to. The notation uses a tree structure similar to FTA, with the attack goal as the top node and different ways of achieving that goal as leaf nodes (Figure 11, example from [138]).

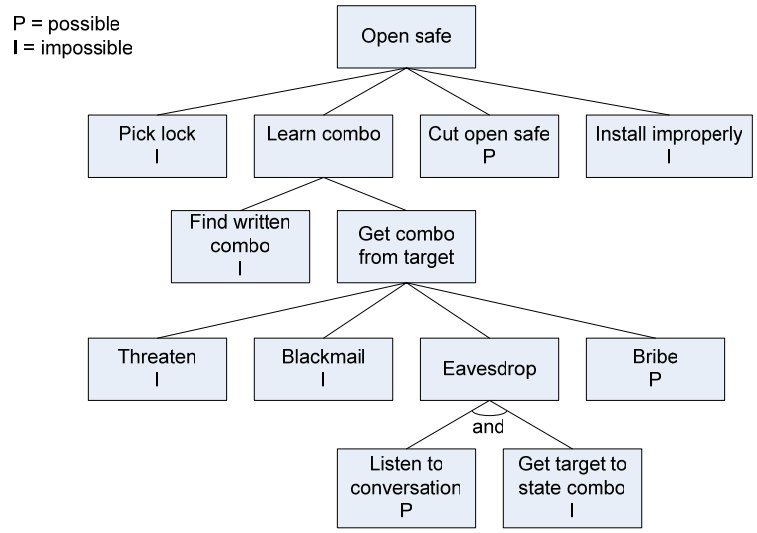


Figure 11 – Attack tree example

There exist an extension of attack trees called *defense trees* [16] which in addition to representing attack strategies performed by an attacker, also represents the possibly countermeasures that may be implemented in the target system to mitigate the attacks.

Since the notation is based on fault trees we get the same difficulties with respect to expressing logical gates, assigning likelihood/probability values to threat scenarios and computing precise likelihoods. Attack trees do however allow for less precise likelihood values (e.g. possible/impossible like in Figure 11) which makes it easier to use for high level analysis. Its focus is more on human behavior than system behavior since it represents different ways of attacking a system. In many cases it may for instance be valuable to represent both how the system reacts to a security breach caused by a human, and a non-human source.

3.4.8 OCTAVE Threat Tree

The OCTAVE method (Sect. 3.2.2) for security analysis has its own tree notation which has much in common with event trees, but also fault trees. The figure below is an *OCTAVE threat tree* from [3] showing the threat posed by a human actors using network access to harm "PIDS" (Patient Information Data System). The diagram illustrates the source of an incident, the method and the motive behind the incident which can be compared to fault trees. At the same time, it illustrates the outcome of the incident that is more like an event tree. The OCTAVE threat tree can be seen as taking the tree notation one step closer to security risk analysis, in particular information security. The use of deliberate and accidental threats (in OCTAVE called motive) is for instance in accordance with ISO/IEC13335 *Information security* (Sect. 3.3.2). Our concern is whether it is feasible to mix two well established techniques like FTA and ETA in this way. This makes it more difficult to exploit the benefits from computerized FTA or ETA tools, and also conflicts with many analysis methods that recommend using these analysis techniques. The intention of adapting the tree notations to more security focused analysis by integrating security related concepts is however good.

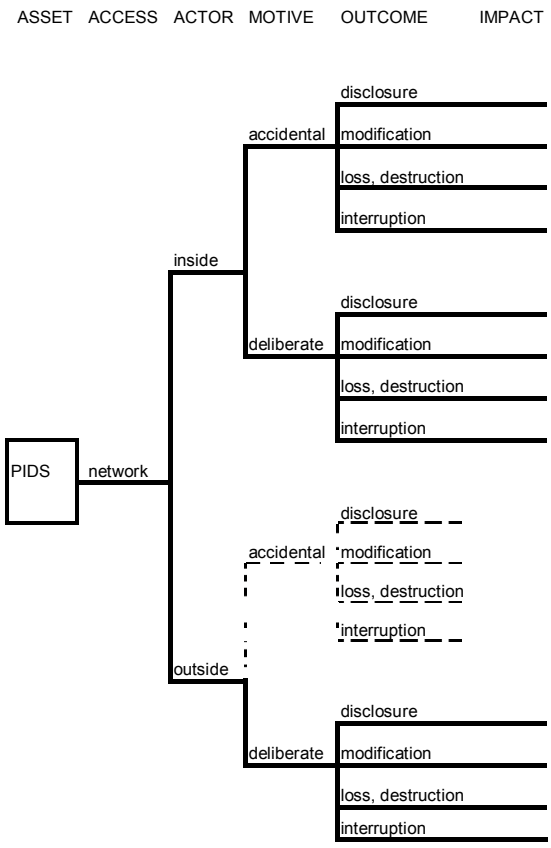


Figure 12 – OCTAVE threat tree

3.4.9 Bayesian Network

A *Bayesian network* [98, 107, 108, 129, 136, 150] is like a directed, acyclic graph. The intermediate nodes represent causes or contributing factors to the top node, which in Figure 13 is a "system failure" (taken from [129]). A Bayesian network is both a graphical and a probabilistic model that may be used to for instance predict the number of faults in a software component [41]. In Figure 13 the causes that contribute strongest to the event (A1-A3) are placed directly before the event. The causes are grouped into three categories: organizational factors, human factors and technical factors. When a Bayesian network is analyzed quantitatively, each node holds a table with a probability distribution reflecting its parent nodes. For any manipulation of the probabilities of the nodes, the effects both forwards (towards child nodes and the top node) and backwards (towards parent nodes) can be computed [42]. A Bayesian network can be utilized both quantitatively and qualitatively. If the Bayesian network is analyzed qualitatively, it provides relations between causes and effects. When analyzed quantitatively, one uses its powerful mathematical model for computing probabilities, which is not only based on the probabilities for the leaf nodes like in fault trees, but also on intermediate nodes.

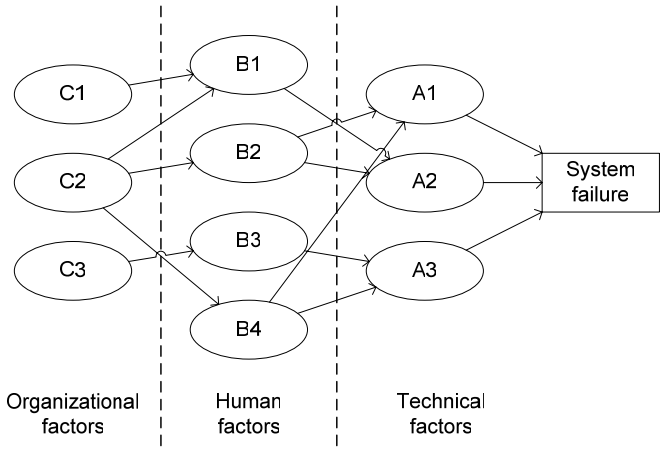


Figure 13 – Example of a BN from [129]

3.4.10 Markov Analysis

Markov analysis [61, 68, 94, 107] is a stochastic mathematical analysis method that looks at sequences of events and analyzes the tendency of which event that will be followed by another. Markov analysis may be used to analyze the reliability of systems that have a large degree of component dependencies. In contrast to FTA, Markov analysis does not assume complete component independence. It is also well suited to analyze systems that may partially fail or experience degraded states. A Markov analysis considers the system as a number of states, and transmissions between these states. The states are modeled graphically and statistical calculations are performed to determine the probability of each state transmission. Markov analysis is among others promoted by ISO/IEC61508. Markov models are more suitable for showing the operation modes of a system where one may transit forth and back between states, than a chain of events of a security attack which is more likely to be a one way chain. Nevertheless, describing the operation modes of a system also includes describing the different barriers that should prevent an attack or reduce the consequences of an attack and for this purpose Markov analysis may be a useful tool. Using Markov analysis requires a well-specified system and may not be as suitable for high-level analyses.

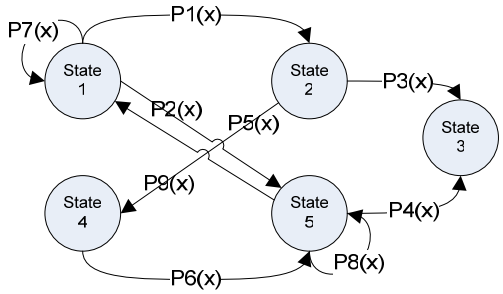


Figure 14 – Markov model

3.4.11 Block Diagram

Block diagrams [67, 129] are often used in reliability assessments and shows how the components of a system are parallel or serial and thereby identifies possible weak points. In the figure below, components 1-3 are parallel, while component 4 is not and therefore may be the weakest point in this system. While this notation is a suitable way of representing the components of the target system, it is not directly applicable to model security risks. The idea of showing alternative paths in a system can however be used in

security risk modeling to show where a threat may work its way through the target of analysis. The UML profile is already using this modeling principle, and it will be an important feature in a new security risk modeling language.

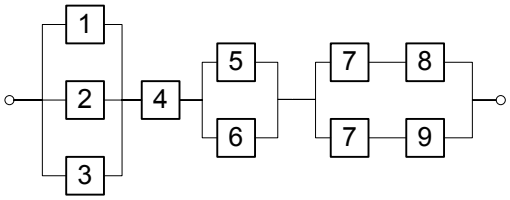


Figure 15 – Block diagram example

3.4.12 Other Techniques

There exist also risk analysis methods that assess UML models on a very detailed level. In [53] the authors propose a severity assessment methodology which integrates Functional Failure Analysis (FFA) [1], FMEA and FTA to analyze the severity of failures of the architectural elements. In [34] the authors describe a risk analysis method of performance failures based on annotated UML models and [50] describes a method for analyzing architecture level UML models to identify the parts of the system which may give rise to risks. The method applies complexity analysis of components and connectors, expert judgments for severity analysis and Markov models to estimate the risk factor of the different system scenarios. A method for estimating risk in the early phases of design is proposed in [5], where the analysis is conducted at the requirements-level based on UML design specifications. In [149] the authors propose a framework that models cluster computing systems' availability based on UML design notations, and evaluates system availability by transforming the UML availability modeling into corresponding analytical models (e.g. Markov models). Common for these techniques is that they deal with risk analysis on a very technical and detailed system level.

Preliminary Hazard Analysis (PHA) [93] is typically used early in the design phase of a plant where the process is new and one has little knowledge of its risks. The outcome is a set of hazards (what we refer to as unwanted incidents), their causes and major effects, in addition to a list of corrective/preventive measures. As the name indicates, the analysis is similar to a HazOp, except for being used in an earlier phase.

MORT (Management Oversight Risk Tree) [87] is a technique that can be used both to investigate the causes of incidents and to audit risk management systems. The technique uses a tree structure similar to fault trees (Sect. 3.4.4). A related approach is SMORT (Safety Management and Organization Review Technique) [96] which builds on MORT and targets internal investigations of incidents and near incidents.

"What if?-analysis" is one of the oldest risk identification methods, based on asking a series of questions often starting with "What if…?" or similar. The method may use checklists as a basis and is used in assessing system design. The analysis method is a typical brainstorming method, where consequences and recommendations are put forward by the people participating. This is an example of a "what if"-analysis result from an analysis of an reactor [93]:

What if?	Consequence / hazard	Recommendations
Cooling water pump breaks down	Runaway reaction explosion/fire	Stand-by pump / alarm systems

3.4.13 Summary

This section summarizes our judgments of existing risk analysis techniques and how they are relevant when developing a new, graphical risk modeling language inspired by the UML profile.

Technique	Relevant to a new risk modeling language?
SWOT	Yes, since it relies on a brainstorming technique which involves several participants it is a suitable situation to use graphical modeling language to facilitate communication and understanding.
FMEA/FMECA	Yes
HazOp	Yes, since it relies on a brainstorming technique which involves several participants it is a suitable situation to use graphical modeling language to facilitate communication and understanding.
Fault tree	Yes, there is no method for modeling logical and/or-gates in the UML profile which means that the reader cannot know which one to use when reading UML profile threat diagrams. The CORAS UML diagrams can probably express the chain of events that a fault tree represents, but not the logical order in terms of and/or dependencies, and not facilitate precise calculations
Event tree	Yes, there is no method for modeling or-gates in the UML profile, but some of the intention of an event tree can be modeled, although it will not look like a conventional event tree. Since event tree is a commonly used notation, the new language should include its idea and objective.
Cause-consequence diagram	Yes, since this is based on both fault tree and event tree, it is natural to include the intention behind cause-consequence diagrams into a new language.
Attack tree	Yes, since the notation is quite similar to fault trees it should be considered and sufficiently covered when developing the new language.
OCTAVE threat tree	Yes, both because it has strong similarities with event trees and because OCTAVE is an international well known security analysis method that a new language should aim to comply with if possible.
Bayesian network	Yes, the event modeling in Bayesian networks is similar to the threat scenario modeling in the UML profile, but the statistical model is unique to the Bayesian network. While it is not within our ambitions to develop this kind of a probability model, it would undoubtedly be very interesting and possibly future work.
Markov analysis	Partly, but only to simulate the different states since the analysis relies on it own, special modeling notation.
Block diagram	Yes, even though block diagrams are more suitable to model components of the target system and not the security risks itself, the idea of showing different paths through a system can be used in a new security risk modeling language.
UML based analysis techniques	These techniques already use UML modeling in analysis, and they also target a much more detailed level of system specification than our approach will address.
Preliminary Hazard Analysis (PHA)	Yes, with its similarities to HazOp we can conclude that also for this technique graphical modeling can be useful
MORT/SMORT	Partly, because MORT is based on using fault trees, which means that it already has a customized modeling notation. However, if the graphical approach is able to express all aspects of the MORT/SMORT analysis and shows superior with respect to understandability or usability, there is no reason for not substituting the fault tree with our approach.
What if? Analysis	Yes, with its basis in a kind of structured brainstorming we are certain that a graphical modeling approach may be useful.

3.5 Risk Documentation Techniques

There are many techniques that can be used to support risk identification, estimation and documentation, but the fact that CORAS already had a graphical modeling language made it a suitable place to start. We therefore give a rather detailed presentation of the UML profile, and then we describe other graphical notations that may be used in risk- and security analysis, and system development.

3.5.1 The UML Profile

The full name for the UML profile is the *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms* [124], and it is standardized by the OMG (Object Management Group). The language is based on the use case notation from the *Unified Modeling Language (UML)* [134]. It uses the official UML meta model, but with added richness in terms of domain specific symbols and concepts related to security risk modeling. In the following we describe the UML profile by means of examples of modeling taken from the profile [124]. The presentation will guide you through a complete security risk modeling process. The diagrams have not been given special names in the standard, but for simplicity we provide each diagram with a name in the figure caption. Stereotyping is a technique used in UML to add information to a model element by giving it special names or stereotyping labels.

The values and scales that will be used during the security analysis are defined in the value definition diagram (Figure 16). The stereotype <<ValueDefinition>> is used for defining each value type that is used. In this example all values are enumerations, i.e., values on an ordinal scale, except for "RiskReductionRef" which defines a mapping. Alternatively, assets could have been defined in terms of monetary values, frequency as probabilities and so on.

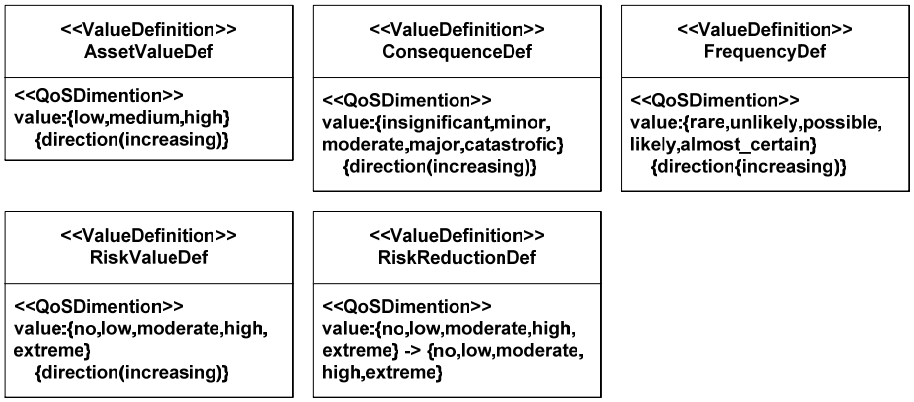


Figure 16 – Value definitions diagram in the UML profile

Figure 17 shows the specification of an asset, which is an important part of the CORAS method. The service is an entity that is associated with some quality characterizations. The asset is defined as the quality level of the service, related to the offered service quality. The asset is owned by the stakeholder "Service provider", and its value is assigned by instantiating the value definition for asset values (Figure 17). The diagram also shows that the asset has one vulnerability, which is "extensive computation".

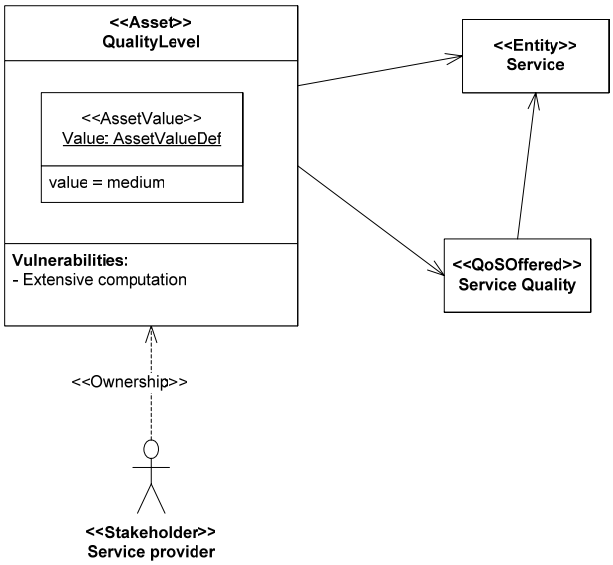


Figure 17 – Asset specification diagram in the UML profile

In Figure 18, the modeling of a threat is exemplified in a UML profile threat diagram. The threat "Malicious person" has the scenario (i.e. behavior) "Flooding". This threat scenario is related to the asset "QualityLevel". In this diagram, asset is shown using the UML actor stereotype only, while Figure 17 provided detailed information about the same asset.

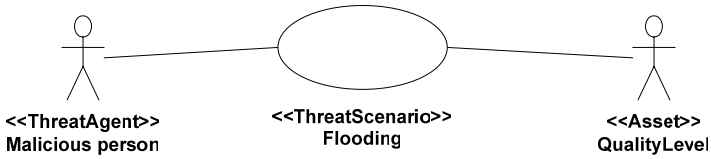


Figure 18 – Threat diagram in the UML profile

Figure 19 illustrates how unwanted incidents are modeled with the UML profile. The unwanted incident "Denial-of-Service" may harm the asset "QualityLevel", and includes the threat scenario from the threat diagram above. A scenario may lead to another scenario, and this is shown by use of the stereotype <<Initiate>>. In this case, "Denial-of-Service" initiates the unwanted incident "Loss of customer" which may affect the asset "Customers".

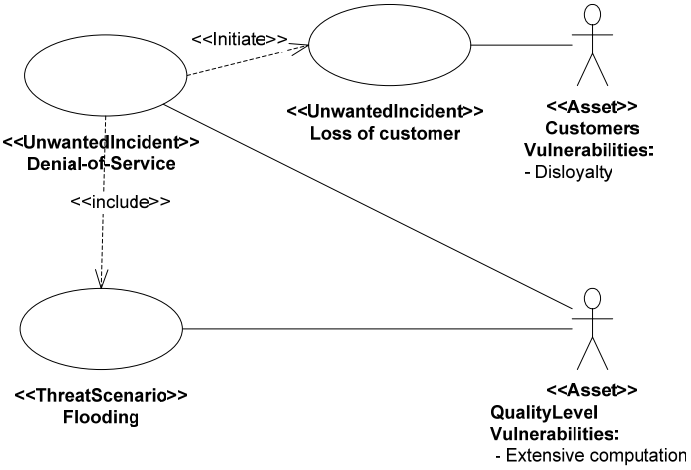


Figure 19 – Unwanted incident diagram in the UML profile

A risk is an assignment of consequence, frequency and risk value to an unwanted incident. Figure 20 illustrates how this is modeled. The values are instances of the corresponding value definitions (Figure 16). The risk of "Denial-of-service" is assigned to the unwanted incident "Denial-of-Service" using the stereotype <<RiskEvaluation>>. The diagram also shows that the risk is related to the asset "QualityLevel".

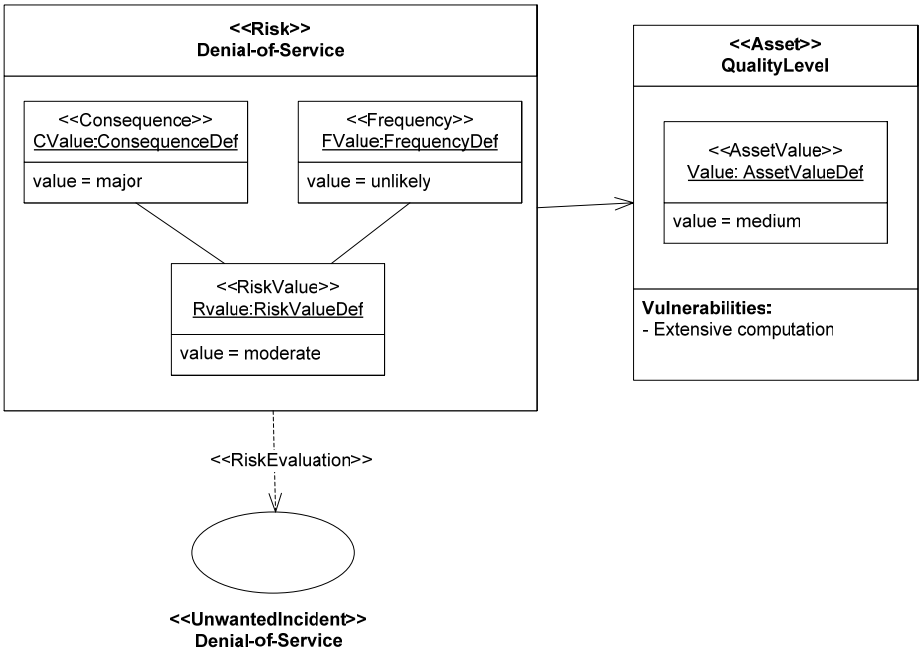


Figure 20 – Risk diagram in the UML profile

Similar risks may be grouped into risk themes. Figure 21 shows how the stereotype <<RiskTheme>> is used to define risk themes of instances of risks. This allows a risk to be a member of several risk themes. In this example, the risks "Denial-of-service" and "Loss of customer" are grouped to form the risk theme "DoSRelated". As seen in the example, a risk theme is also assigned an overall risk value.

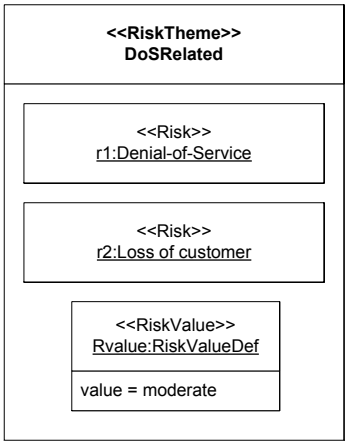


Figure 21 – Risk theme diagram in the UML profile

Figure 22 models "Authentication" as a treatment for the unwanted incident "Denial-of-Service". The stereotype <<Transfer>> (one of the predefined treatment options in AS/NZS4360) denotes that this treatment involves transferring the responsibility for the risk to the authentication mechanism solution.

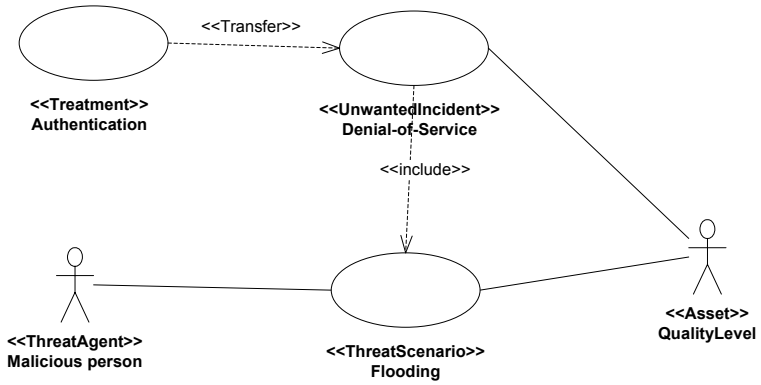


Figure 22 – Treatment diagram in the UML profile

Figure 23 shows an example of how a treatment effect is modeled. The treatment effect "DoSTransfer" is bound to the treatment "Authentication" by the use of the stereotype <<TreatmentEvaluation>>. The figure also shows that "DoSTransfer" relates to the risk "Denial-of-Service". The risk reduction, i.e. the value of the treatment effect, is a mapping from *moderate* to *low*, which means that implementing the treatment will reduce the risk value of "Denial-of-Service" from moderate to low.

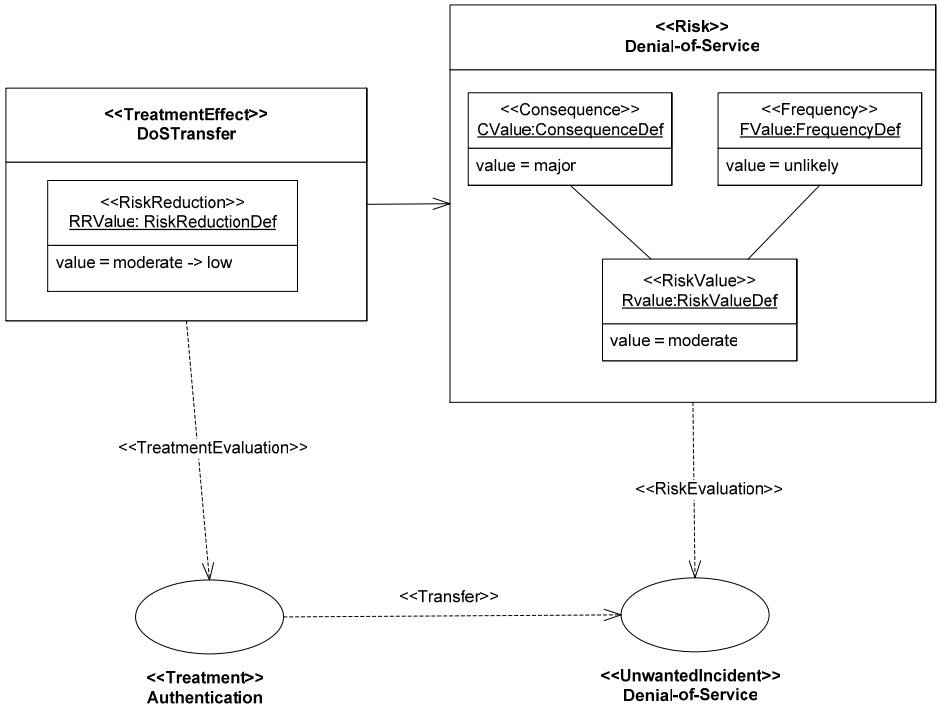


Figure 23 – Treatment effect diagram in the UML profile

The UML profile is as far as we know the only modeling notation that supports the entire security analysis process step-by-step. However, there exist related notations that can be used to model particular parts of the documentation in a security analysis, like reliability aspects of the system analyzed (e.g. "block diagram" in Sect. 3.4.11) or potential ways of attacking a system (e.g. "attack tree" in Sect. 3.4.7).

Comparing CORAS to the other techniques
The threat diagram in the UML profile may resemblance a fault tree, but may have more than one top node and lacks the notion of logical gates. The threat scenarios can be seen as leaf nodes/intermediate nodes are not given probabilities, something which makes fault tree computation impossible. The CORAS UML diagrams can express the chain of events that a fault tree represents, but not the logical order in terms of and/or dependencies. Neither can the CORAS UML diagrams provide a model for precise likelihood calculations.

The concept of barriers that one finds in event trees is underspecified in the CORAS UML diagrams, which only models the relation from an unwanted incident to each of the assets it potentially may harm. The success or failure of the barriers and the different types of consequences are not modeled explicitly. In addition, the consequence towards each asset must be modeled in separate CORAS UML diagrams

The UML profile provides insufficient support for the end-to-end view cause-consequence diagrams give. The complete chain of events are not modeled in one diagram, but must be specified using multiple diagram types.

The UML profile can be used in the same manner as an attack tree, with a threat assigned to the initial threat scenario as the leaf node, more threat scenarios as the intermediate nodes and the unwanted incident as the root node. The UML profile may on the other hand extend the attack tree notation by illustrating which assets that are affected by the unwanted incident (root node), and also the threats initiating the unwanted incidents.

The concepts modeled in OCTAVE threat trees are in many ways the similar to the concepts modeled in UML profile diagrams but they have different names. The concept "asset" is similar to asset in the CORAS method, while "access via network" would be considered a vulnerability. Both "inside actor" and "outside actor" would be called threats in the CORAS method, the "outcome" would be an unwanted incidents and their "impact" would be consequences in terms of damage to the asset. Also the information modelled is similar to the one gathered in the CORAS method, but the UML profile would normally not be used to model all possible outcomes of an event. This is however possible, but maybe not feasible when the events are numerous.

If the Bayesian network is analyzed qualitatively, it provides relations between causes and effects similar to the UML profile. The difference lies however in the Bayesian network's powerful mathematical model for computing probabilities, which is not only based on the probabilities for the leaf nodes like in fault trees, but also on intermediate nodes

The strength of the Markov model is the complex, underlying statistical model used for calculating probabilities for the different states of a system. The UML profile does not include a similar feature, and can at most be used to illustrate the different states of a system.

3.5.2 Misuse Case

The *misuse case* notation [145-147] is related to the UML use case notation (the example in Figure 24 is taken from [145]). As opposed to a use case which expresses allowed functionality in a system, a misuse case expresses the opposite, i.e. the functions that the system should **not** allow. A misuse case can be defined as "a completed sequence of

actions which results in loss for the organization or some specific stakeholder" [145]. Among others, misuse cases have been used in design of secure systems architectures [126]. A similar notation to use cases is *abuse cases* [115, 117] which by Sindre and Opdahl is considered to be complementary to misuse cases. Abuse cases target security requirements with respect to design and testing, whereas misuse cases are used to elicit security requirements in relation to other system requirements.

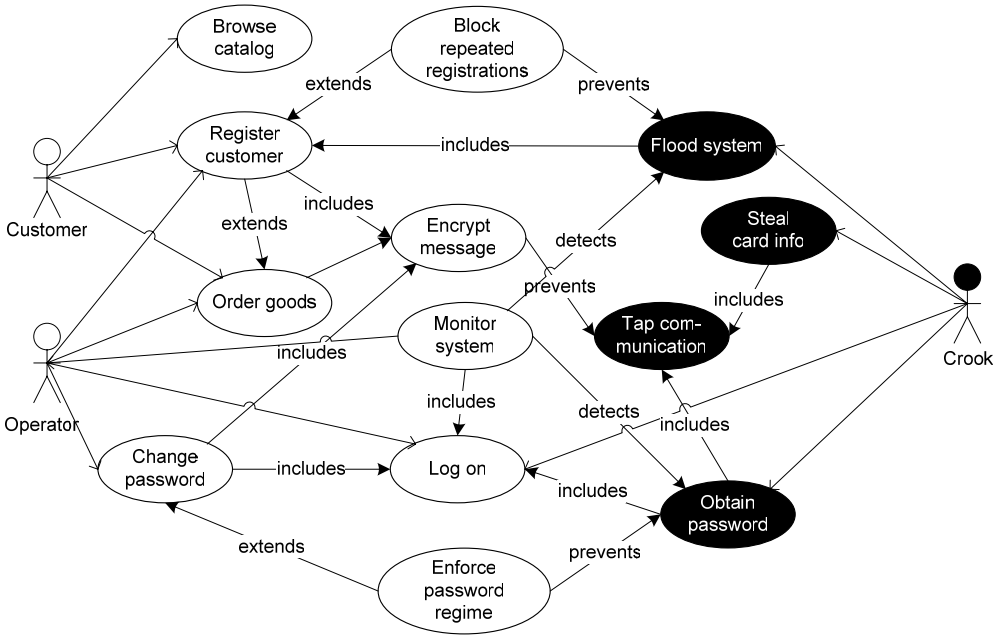


Figure 24 – Misuse case example

Misuse cases can be a useful tool in security analysis to direct focus towards functions in a system that may be exploited. The UML profile is inspired by the misuse case notation, but is richer and has a stronger relation to traditional risk analysis techniques. Misuse cases can therefore be expressed with the UML profile.

3.5.3 UML Variants for Security

There are a number of security oriented extensions and applications of UML, and in this section we introduce some of them. These and related approaches have however all been designed to capture and analyze security properties and security aspects at a more detailed level than what is required in the type of modeling aim for.

UMLsec [89-91] is an extension of UML (a UML profile) that provide means for specifying security requirements. The underlying basis is an "abstract state machine model" that formalizes UML elements (except for use cases) and extends stereotypes. The purpose is to be able to formally verify software specifications, which may reduce the number of security risks. A similar approach to UMLsec is described in [39], focusing on extending properties of essential UML elements (including use cases, actors, classes and methods) in order to apply security models directly (exemplified with their "mandatory access control" model).

Another language is *SecureUML* [11, 12, 110] which aims to extend UML with a meta model for role based access control (RBAC) [44] for use in model driven security engineering. Their "Model Driven Security" approach is based on first specifying systems models and their security requirements and then use tools to generate the system

architecture from these specifications. The approach combines system modeling and system security in a detailed level with particular focus on RBAC. RBAC is also targeted in [130] where the authors model the concept as reusable UML templates, more specifically by proposing a class diagram template for RBAC and use object diagram templates to specify RBAC constraints.

authUML [4] is a smaller framework in the same category as SecureUML and UMLsec which lets the modeler formally model and analyze access control requirements in UML use cases. Also within the area of aspect-oriented design there exists methods for analyzing the consequences of security concerns to other functional concerns [48], and in [116] the authors propose a method for visual security protocol modeling to be used within model-driven architecture.

3.5.4 Simple Technique for Illustrating Risk

The simple technique for illustrating risk (STIR) [114] modeling technique uses the UML use case diagrams to illustrate assets, threats and safeguards (existing countermeasures against risks). STIR has similarities with the UML profile, but the technique has fewer concepts and a different way of modeling. Figure 25 from [114] is an example of an ATS diagram (assets-threat-safeguard) that is used to get a complete overview of the threats for each asset and their protection by safeguards. The ATS diagrams are then examined to see whether the assets are sufficiently protected and the findings can be used as input to a more detailed risk analysis.

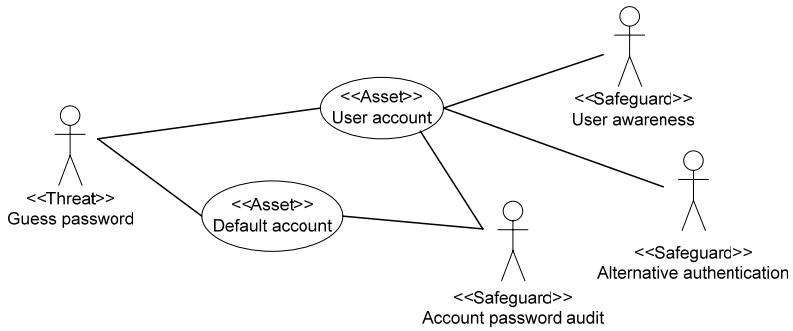


Figure 25 – Asset-threat-safeguard diagram

3.5.5 Riskit Graph

The *Riskit method* [97] includes a risk modeling technique based on a graph notation that makes it possible to specify factors that may influence a software development project (example in Figure 26 from [97]). The Riskit method deals with project risks and has main focus on supporting software development organizations in *developing* their products. The evaluation and management of risks that might occur during the operation of software has therefore been left out [97] (p. 12). Riskit uses its own definitions inspired by for instance organizational strategy research [45]. A *factor* may be compared to a threat scenario in UML profile, while the *event* is an unwanted incident. *Reaction* can be compared to consequence, and the *effect set* can be seen as a further detailing of the consequence. Riskit lacks "threat", possibly because it unusual to consider deliberate harmful actions towards a software development project when assessing the project risk. It also lacks the notion of "vulnerability".

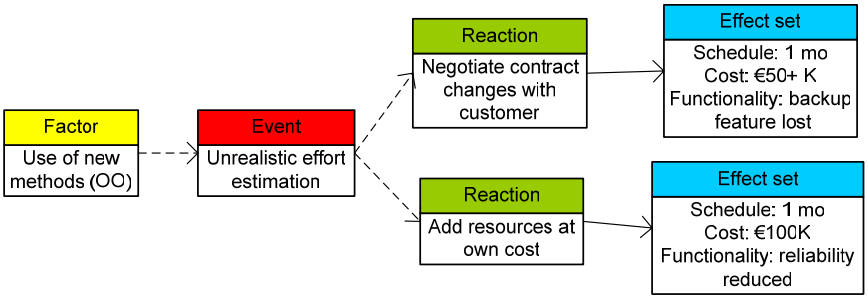


Figure 26 – Example of a Riskit graph

3.5.6 Microsoft's Threat Modeling and Related Approaches

In [59, 60, 153] Microsoft present what they call *threat modeling for software applications*. The process involves defining threats to a system, ranking them according to their risk level, and finally choosing between different techniques to mitigate them. By using their threat model STRIDE, the risk analysis will be focused towards particular threat scenarios (i.e. Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, and Elevation of privilege). To support the process they make use of data flow diagrams [38, 46] to describe the target, and a kind of tree notation to rank risks (quite similar to attack trees). The threat modeling takes place in the design phase to help reveal potential risks, but it is also claimed to be helpful in code review and testing.

In [121] another method called threat modeling is presented. The process resembles [153] but claims that *the sequence and description of steps is different and the execution of steps is extended to suit complex, networked systems*. The threat modeling is used as a basis for defining security requirements to a system and consists of three steps: (1) characterizing the system, (2) identifying assets and access points, and (3) identifying threats. Only step 1 seems to involve modeling, the other two assess the models from step 1 using check lists for common threats, vulnerabilities, attack goals etc. Attack trees [138] are mentioned during threat identification, but only as an additional mean that may be used to support the process. The outcome of the process is a threat profile for the system that is used for security requirement elicitation.

The process presented in [169] is claimed to be a lightweight formal complement to Microsoft's threat modeling approach. The process focuses on modeling functions, threats, and threat reducing efforts and then it checks the consistency between security threats and functions. Finally, it verifies the lack of threats in the refined model of indented functions and threats that have been mitigated. The process employs high level Petri nets (Predicate/Transition nets) [47], a formal method with both a graphical as well as a mathematical notation, often used to describe distributed systems.

3.5.7 Summary

This section summarizes the various modeling techniques that may be relevant to a new security risk modeling language.

Modeling technique	**Is the technique relevant to a new security risk modeling language?**
Misuse case	Yes, misuse cases were a source of inspiration for the UML profile and its ideas and notation will also be relevant to a new language.

UML variants for security	Partly, since these notations deal with risk analysis on much more detailed system specification level than our approach will do. They are however interesting as notations that can be used if there is a need for further specializations detailing of the threat scenarios.
STIR	Partly, because the UML profile already covers the modeling aspects of STIR.
Riskit graph	Partly, since the notation mainly addresses project risks and it becomes a bit awkward to "force" project related concepts like tools, techniques, resources and project schedules into a security analysis focused setting like the CORAS method.
Microsoft Threat Modeling	Yes, since the intention of dataflow diagrams and attack trees already may be illustrated in the UML profile as threat scenarios and unwanted incidents. It may also be interesting to investigate the possibility of complying to this approach since it represents a major, international threat modeling approach.

3.6 Information Visualization Techniques

The phrase "a picture is worth a thousand words" is claimed to bee a too simplistic view [164] since the usefulness of pictures highly depends on the situation at hand, and the skills of the reader. Nevertheless, software engineering studies have shown that applying graphical means to program- and component specifications increases the understanding of the system [18], and one has successfully applied graphical means to visualize the program interdependencies [109]. According to [106], the key activities for a reader of a diagram is searching and recognizing relevant information, and then using this to draw conclusions. Many graphical means and mechanisms may help the reader with these activities. We were inspired by research within the fields of information visualization in computer displays and statistical graphs/figures, related modeling languages, diagrammatic reasoning and model quality.

One of the key issues when documenting risks is the ability to show which methods a threat may use to initiate risks, and which of these are more likely than the other are. In many ways, this can be compared to showing different paths through a graph, and therefore we refer to it as graph navigation. It is also useful to be able to illustrate the weaknesses or deficiencies that make one vulnerable to threats, and finally one should be able to provide a summary of the most crucial risks. In the following we motivate and present the graphical means and mechanisms that may be used in security risk modeling, structured according to these three main issues: *representing graph navigation, vulnerabilities* and *risk*.

3.6.1 Representing Graph Navigation

We refer to the process of understanding and reading diagrams as "graph navigation". When describing the different paths via which a threat may choose to harm assets it may be useful to draw special attention to the paths that they are more likely to follow. It was also desired by the users of the original UML profile to have a way of describing the logical operators AND and OR when two paths are combined. An ambiguous threat scenario situation where logical operators could be useful is described in Figure 27.

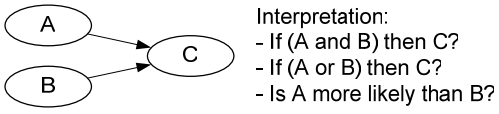


Figure 27 – Typical threat scenario situation

The original UML profile is based on a node-link diagram type, which in its simplest form consists of nodes connected via edges. A closed contour in a node-link diagram generally represents a concept of some kind, and a linking line between concepts represents some kind of relationship between them. Manipulating the appearance of these lines may give them different meanings, for instance the likelihood of the different paths (Figure 28).

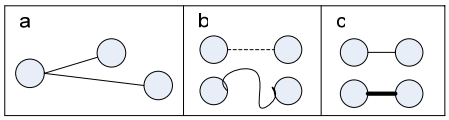


Figure 28 – Node-link configurations

Lines linking closed contours may have different colors, be straight or wavy, or have other graphical qualities that represent an attribute or type of relationship [160]. The thickness of a connecting line may be used to represent the magnitude of a relationship (a scalar attribute). Contrasts in the size or color of a line may make it "pop-out" perceptually and thereby call for the reader's attention [156]. This may be a useful feature to include in a security risk modeling language.

Network traffic maps that often use node-link diagrams to visualize network traffic, also inspired the use of line variations. Typically one illustrates the load or type of link data using different line colors or varying the thickness of the lines [14]. Variation of line appearance is also inspired by the gestalt principle *similarity*, meaning something that is different from its surroundings, will be easier identified by the reader (e.g. *italic style* vs. normal style in text). The gestalt principles [161] are some of the most recognized set of rules for good graphical displays. Implementing the gestalt principles may reduce the effort needed to understand illustrations, program interfaces, websites etc.

When looking for an AND-operator symbol, it was natural to turn to one of the most frequently used technique in risk analysis: Fault Tree Analysis (FTA) [66]. Fault trees let you specify dependencies between events in a tree structure using logical AND/OR operators called "gates" (Figure 29), similar to the ones used in electrical circuit design [140]. One of our goals is to make the security risk modeling language fault tree compliant. In the original UML profile, there was no standard interpretation of two threat scenarios pointing to the same threat scenario or unwanted incident (Figure 27). In order to express fault tree-logic we needed a way of saying both "If A and B occurs, C must occur" and "If either A or B occurs, C must occur". The challenge was to find the preferred way of illustrating this.

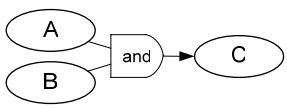


Figure 29 – The AND-gate symbol

An alternative to logical gates may be the UML inspired "information note". In UML one may attach notes to a diagram element to convey additional information about the specific element.

3.6.2 Representing Vulnerabilities

The original UML profile could only illustrate vulnerabilities as properties of their respective assets. There was no way of showing exactly where in the chain of events a

vulnerability was exploited, and there was no special vulnerability symbol. Both these aspects were requested by the users of a security risk modeling language.

The way vulnerabilities was modeled in the original UML profile made it simple to see all the vulnerabilities of an asset at a glance, but involved the duplication of vulnerabilities that were shared between two or more assets. Vulnerability is a stand-alone concept, not a binary relation as the ones discussed in the previous section on graph navigation. The descriptive text accompanying the vulnerability is obligatory. The question was whether it should have a symbol in addition to the textual description.

In research on the quality of modeling languages, pictures are claimed to have a much higher perceptibility, and information conveyed in pictures will be emphasized at the cost of textual information [99]. To achieve what [49] terms "syntactic disjointness" , the symbol must be unique and it must neither be too small, nor too large to cause misinterpretations (large elements often attracts attention at the sacrifice of the smaller elements). Syntactic disjointness makes it easier to separate the different elements in a model.

One may argue that the way vulnerabilities was modeled in the original CORAS language was an example of the gestalt principle *proximity*, stating that things that logically belong together should be placed close to each other (commonly used in program interfaces and websites). This view is very suitable if the focus is on the assets and their vulnerabilities, but when it comes to addressing *which* threats exploiting *which* vulnerabilities and especially *where* in the chain of events the exploitation takes place, the situation is different. This kind of information is especially valuable in relation to treatment identification.

Winn [165, 166] has found that people, whose languages are written from right to left, also process graphics from right to left. A related finding is that the item to the left always was interpreted as the cause while the item to the right was the effect [167]. This is in accordance with one of the node-link rules [160] stating that "placing closed contours spatially in an ordered sequence can represent conceptual ordering of some kind" (Figure 28, alt. b), and vulnerabilities may very well be placed within this ordering. The direction of the line should also be indicated with arrows along the line, since this have been found helpful in understanding similar notations like flow charts (used in schematic representations of processes) [29]. If the number of lines increases one may easily loose track of the path and repeated arrows help distinguishing between crossovers and connection points.

3.6.3 Representing Risks

Representing risk is a great challenge to its abstract nature. In the threat diagrams of the original UML profile risks were not illustrated explicitly, but represented as relations between unwanted incidents and assets. Illustrating the severity or magnitude of risks and unwanted incidents helps keeping focus on the most critical risks, something that is especially useful during the final risk treatment phase.

To visualize this issue, one may use means like line thickness, textual information labels, shapes that varied in size or color and more. The color of an enclosed region may represent a concept type (Figure 30) and the size may be used to represent the magnitude of a concept (Figure 31). This is also inspired by the gestalt principle of breaking the pattern to

attract attention. The reader of a risk diagram should quickly discover the most serious unwanted incidents, since they often represent major risks.



Figure 30 – Color

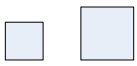


Figure 31 – Size

There are some aspects to be aware of when using color-coding. The number of different colors one may use is limited by the reader's ability to remember and distinguish the colors. Much research has been conducted to find the optimal number of colors, and the suggestions varies from 5 to 8 [27, 31, 32, 142, 163]. One should also have in mind how a model with colored symbols will look when printed in black-white and whether this may affect the interpretation of the symbols [49, 160].

Different types of coding in statistical graphs have been investigated with respect to *display search time*. The coding that gave the best performance was color, the second best coding was shape and then came letters/digits [30]. This top ranking of color has been confirmed by many other studies [88]. According to these results, one should benefit from using color to emphasize the most serious incidents, meaning that the reader should identify them more quickly compared to using other means. As in the case of colors, the number of different size-categories is not indefinite. The number of different size steps that can be distinguished from each others at a glance is as low as four (p. 182). An important aspect when using shapes is to avoid symbols of similar shapes. Shapes that are too similar have been found to increase search time and are therefore not recommended [148, 160].

Studies of *rapid processing of information* [28, 137] have found letters and digits to be some of the best coding forms. These investigations involved Sternberg tasks[1] which studies short-time memory and how graphical means are memorized. Digits, colors and letters were found to be the top three coding forms for rapid processing of information. Another investigation found digits, words and letters to be superior to colors and shapes when it comes to processing speed, and digits, letters and words were found to represent less subjective workload than colors and shapes [155]. These findings support the use of textual information in graphical modeling.

In some general rules regarding the use of shapes are described. For instance that the shape of a closed contour can be used to represent a concept type (Figure 32). Risk was one of the few relevant concepts for which the original UML profile did not have a special symbol.

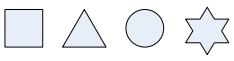


Figure 32 – Shapes

In [19, 160] the authors address evaluation of languages in terms of how easy they are to understand for users that are new to it. They look at it from the perspectives of philosophy, linguistics and cognitive psychology. They use a selection of the cognitive dimensions of [51, 52] to assess a languages ease of understandability. One of the dimensions is the number of different symbols in the language. If the number is high, the language will be

[1] For a limited time the subject is shown a set of objects which has to be memorized.

more difficult to learn, but the number of symbols must be high enough to ensure the language's expressive power. Other dimensions are the consistency and the discriminability of symbols in the language. Both inconsistent symbols and too similar symbols may cause misunderstandings. The degree of "motivation" of the symbols in the language plays an important role related to understanding. In a motivated language there exist a natural relation between the elements of the language and the concepts they represent [135]. The symbols in a motivated language will bear meaning in themselves and therefore make it easier to understand. As the authors point out, this is a challenge in software representations since it is often the effects and interfaces of the system that are interesting, not the "thing" itself. Motivation is also dependent on the culture the symbol is used in (e.g. a red cross vs. a red half moon).

3.6.4 Summary

This section summarizes the graphical mechanisms that may be used in a security risk modeling language, and also for what purpose they may be used.

Mechanism	Should be tested for the following purpose
Size	Size may be used to show likelihood of paths in a graph, the magnitude of threats, vulnerabilities, threat scenarios, unwanted incidents and risks.
Text labels	Text labels may be used to give additional information to the reader, possibly about magnitude or seriousness of risks, unwanted incidents and more. It may be used in combination with more graphical mechanisms.
Color	Color may be used to show the magnitude of threats, vulnerabilities, threat scenarios, unwanted incidents and risks
Shapes (symbols)	Intuitive shapes or symbols may be used to illustrate important concepts which contributes to the risks, e.g. vulnerability, different types of threats, assets or similar.

4 Research Method

This chapter describes the research method used in our thesis work. We first explain the concept of technology research, its methods and evaluation techniques. Then we present and motivate our instantiation of this, in particular how the results have been evaluated.

4.1 Technology Research

Within research on methods, tools, languages etc. for use in system development there are two main research paradigms: the behavioral-science paradigm and the design-science paradigm [56, 113]. The behavioral-science paradigm originates from research methods within natural science where it is used to develop and refine principles and laws. This is also considered to be the "classical" way of doing research [151] as we know it from mathematics, physics, chemistry and the social sciences. This type of research aims to explain phenomena in the world around us. When used in relation to system development it focuses on organizational or human aspects. The second research paradigm, design-science derives from engineering and the sciences of the artificial [144]. It aims to solve a problem by developing new artifacts which in relation to e.g. IT systems are defined to be *constructs, models, methods* and *instantiations* [113].

The research process used in behavioral-science can be summarized in three steps: (1) problem analysis, (2) innovation and (3) evaluation [151]. In the problem analysis phase the need for new explanations or adjusted theory is identified. The innovation involves coming up with new explanations or extend already existing theories. The evaluation should provide support to the new theory, i.e. show that it is in accordance with reality. This process may be repeated until the research goal is achieved. Design-science research, or technology research as it is called in [151], may use the same process as behavioral-science: the problem analysis focuses on identifying the needs with respect to a new artifact. The innovation consists in inventing and developing this artifact, and finally the artifact should be evaluated against relevant needs to show its usefulness.

Hevner et al. [56] have defined a set of guidelines for design-science in information systems. Our research domain, security risk analysis of computerized systems is also relevant to information systems and can therefore benefit from using the same set of guidelines. The guidelines are meant as assistance for conducting effective design-science research:

1. Design-science research requires the creation of an innovative, purposeful artifact.
2. The artifact must be for a specified problem domain.
3. Thorough evaluation of the artifact is crucial.
4. The artifact must be innovative, solving an, until now, unsolved problem or solving a known problem in a more effective or efficient manner.
5. The artifact itself must be rigorously defined, formally represented, coherent, and internally consistent.
6. The process by which it is created, and often the artifact itself, incorporates or enables a search process whereby a problem space is constructed and a mechanism posed or enacted to find an effective solution.
7. The results of the design-science research must be communicated effectively both to a technical audience and to a managerial audience.

As stated above, a thorough evaluation is crucial and in the section below we survey research methods for evaluation.

4.1.1 Methods for Evaluating the Results

As highlighted by McGrath [118], an ideal evaluation method should offer high precision, be realistic and facilitate generality. Unfortunately, these three aspects are fundamentally different from each other and cannot be reached using one evaluation method only. As Figure 33 shows, an evaluation method that gives high precision like a laboratory experiment is low on both realism and generality. The challenge for the researcher is therefore to select a set of evaluation methods that brings sufficiently support to the research results. McGrath classifies evaluation methods into eight categories as follows:

- *Laboratory experiment:* is an attempt to create a copy of an existing system where the researcher can control and possibly isolate (all) the variables to be examined. The method can give precise measurements, but lacks realism and its findings are not easily generalized.
- *Experimental simulation*: is a laboratory test aiming to simulate a real situation where the subjects participate for research purposes. When it comes to strengths and weaknesses, the method lies between a field study and a laboratory experiment.
- *Field experiment*: is a field study carried out in a natural environment, but in which the researcher intervenes and manipulates a certain factor. The method lies between a field study and a laboratory experiment when it comes to strengths and weaknesses.
- *Field study*: consists of direct observations of a "natural" (existing) system, with little or no interference from the researcher. The method is realistic, but lacks precision and cannot easily be generalized.
- *Computer simulation*: is to create a model of the real world. The results are low on precision, but score on realism and can to a certain degree be generalized.
- *Non-empirical evidence (formal theory)*: is argumentation based on logical reasoning rather than empirical evidence. The results from using the method are suitable for generalization, but scores low on realism and precision.
- *Survey*: is a collection of information from a broad and carefully selected group of informants. The results from a survey may be generalized, but there is less focus on realism and precision.
- *Qualitative interview (judgment study)*: means collecting information from a few selected individuals who provide more precise and detailed answers than obtained through a survey. The method gives results that are more precise than those of a survey, but cannot be generalized to the same degree and is low on realism.

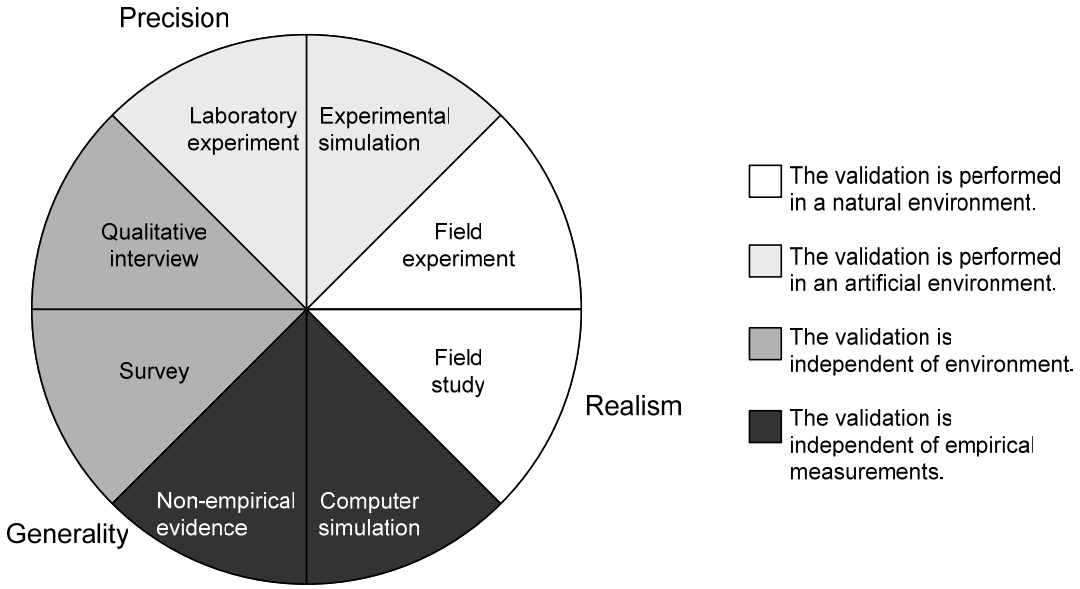


Figure 33 – Evaluation methods [118]

4.2 Our Research Method

As already explained in *Chapter 3*, our thesis work has been carried out in the context of the research project SECURIS funded by the Research Council of Norway. The research activities in SECURIS have been driven by industrial filed trials. Each year from 2003 until 2007 there was 1-3 such trials. Methods and tools developed within the project were thereby tested out in real industrial settings to ensure that they met the needs of the industrial partners. Within the SECURIS project, and also in this thesis we refer to these trials as field trials although they were more like experimental simulations in the classification of McGrath.

The overall goals of our thesis work is characterized in the problem description in *Chapter 2* as a set of success criteria. The problem description was initially very general and unspecified, but as the research progressed it was refined into concrete criteria. This iterative and incremental research approach has been used throughout our thesis work. More specifically, for each research challenge that occurred we have typically proposed one or more solutions and evaluated these to select the most promising solution for further work. This iterative process has resulted in four new artifacts, but also new knowledge that have general interest beyond our specific approach to security analysis.

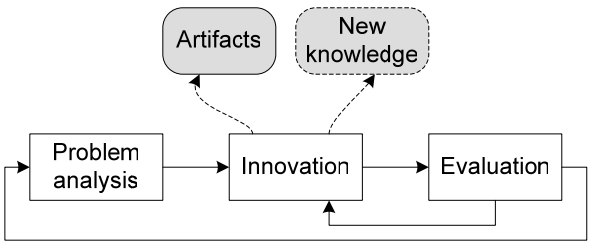


Figure 34 – Our research method

In the following we present the research methods used for each of the artifacts in the order they were presented in *Chapter 2*:

1. A common basis for security analysis terminology
2. A language for specifying and documenting security analysis findings

3. Modeling instructions for the user
4. A way of evaluating the quality of security risk modeling approaches

It is important to note that even though these artifacts are treated separately below, they were in fact developed in parallel, and one artifact may build on another (e.g. the language builds on the conceptual foundation).

4.2.1 Developing a Common Basis for Security Analysis Terminology

The CORAS security analysis method already had a conceptual foundation for security analysis terminology. However, this conceptual model had not been subject to empirical investigations addressing understandability.

In order to improve the original model we investigated it in two iterations, first using students as subjects (Empirical investigation of terminology I), and then professionals within system engineering and risk analysis (Empirical investigation of terminology II). *Chapter 5* summarizes the two studies, while the full details are available in *Appendices A* and *B*.

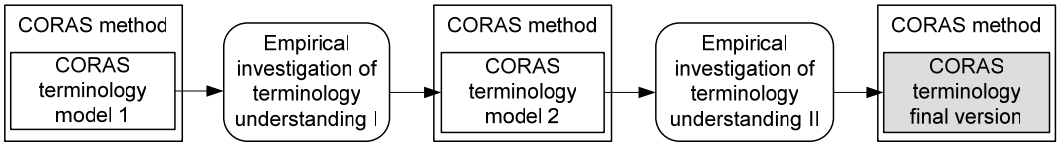


Figure 35 – Method used for developing the foundation

By investigating alternative solutions of the conceptual foundation, the evaluation became a part of its development. Both investigations were surveys based on questionnaires, where the knowledge of risk analysis terminology was tested. According to McGrath, surveys give results that may be generalized, but they will be less precise and low on realism. In this case we argue that the realism of the evaluation was higher than normal since we investigated the intuitive understanding of the concepts, something that may be relevant in any context. The precision of the results is however low since surveys written in natural language may give room for misunderstandings. Questions that seemed to have caused trouble for the subjects were discarded from the analysis. By doing this we reduced the possibility of misinterpreting the findings, which can be seen as an attempt to increase precision.

In the following tables, we present the set-up for the two investigations, including a description of subjects, material, analysis method, and how the results have been used.

Table 1 – Set-up for terminology investigation 1

Subjects:	31 master students in systems engineering from the University of Oslo.
Material:	A questionnaire with 20 short statements. Each covered the relationship between 2-4 security analysis concepts (in the conceptual model), and was either correct or incorrect.
Measuring:	The score for each statement, looking for particular easy or difficult concept-relationships.
Results used for:	Redesign of the conceptual model.

Table 2 – Set-up for terminology investigation 2

Subjects:	57 subjects: • 12 professional researchers, including 2 industrial psychologists, from the Institute for Energy Technology (http://www.ife.no), who had competence on developing safety critical systems for nuclear power plants. • 22 professional researchers, including 3 industrial psychologists, from SINTEF ICT (http://www.sintef.no) with competence on software development, modeling, and usability studies. • 23 master students in systems engineering from the University of Oslo and the University College of Østfold.
Material:	A multiple choice questionnaire with 16 questions or statements about risk analysis terminology based on the conceptual model that had been redesigned since terminology investigation 1.
Measuring and analyzing:	The data was analyzed with respect to three aspects: • to what extent do the subjects answer correctly without being properly trained? • is there any difference between the score of the students and the professionals? • which concepts and relations are particular difficult or easy to understand? We used the definitions of the conceptual model to decide whether an answer was correct or incorrect, and the results were given as percentage of all the subjects. The difference between the students and the professionals was investigated using a one tailed t-test, with a significance level α=0.05. 57 subjects is considered to be sufficient for assuming normal distribution and therefore a parametric t-test may be used (the limit is said to be about 30 subjects [159]). To identify particularly difficult or easily understandable parts of the terminology similar questions were grouped into a total of six groups and the differences between these six question groups were investigated using the Wilcoxon signed rank test [159] (p. 614), also with a significance level α=0.05. In this case a more conservative, non-parametric test was used since the number of questions for each category was different and therefore the data was not expected to be normally distributed. A non-parametric test is more conservative than its parametric alternative in the sense that it needs a larger sample set to discover differences than a parametric, but can on the other hand be used for smaller sets and does not assume anything about the population of the sample set. SPSS v13.0 on Windows XP was used for all the statistical tests.
Results used for:	Confirming the design of our conceptual foundation which forms the basis for the graphical security risk modeling language, in addition to pointing out possible final adjustments.

4.2.2 Developing a Language for Specifying and Documenting Security Analysis Findings

With the UML profile (Sect. 3.5.1) as the starting point, an improved language was developed. The process has been iterative and incremental (Figure 36) in the sense that the language has been tested in several field trials, each time in a improved version, based on the previous field trials and also findings from other investigations. After completing the final version of the graphical language, a structured semantics has been defined (Sect.

8.1.4) capturing the intended meaning of the CORAS diagrams in terms of paragraphs in English.

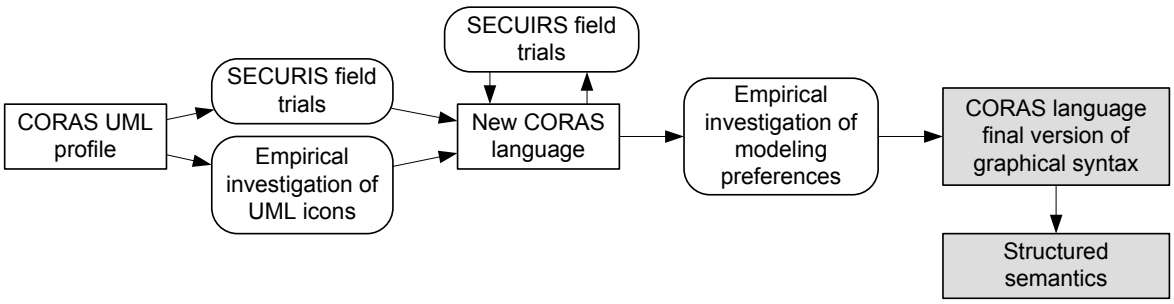


Figure 36 – Method used for developing the security risk modeling language

Different evaluation methods have been used during the development of the CORAS language. The evaluations have given valuable input for further research and redesign. The evaluation methods used were: SECURIS field trials (what McGrath refers to as experimental simulation), "classroom experiment" (close to laboratory experiment), survey and a "modeling benchmarking test" (a kind of experimental simulation).

SECURIS field trials
The field trials in SECURIS also falls within the category of action research [13, 37] since the researchers played the role of analysts and therefore to some extent were within the scope of the object studied. The researchers facilitated and led the analysis, while the client decided the scope of the analysis and provided people and resources. Each security analysis required about 250 person hours from the analysis team, and 50-100 hours from the client. The security analyses produced two reports, one confidential report presenting the findings regarding security risks, and one presenting lessons learned from applying the CORAS method and the current version of the security risk modeling approach. The results were used to improve both method and modeling approach before the next SECURIS trial.

The first application of the UML profile in real security analyses provided experience with practical use, and formed the basic requirements to such a language. In the subsequent field trials, improved versions of the modeling language were tested and a modeling practice was gradually established. The field trials were a unique opportunity to test the method and modeling on real security issues since it was integrated in the client organizations regular work on security. The client and its representatives had a genuine interest in the findings from the analysis, and the results influenced decisions beyond the scope of the actual analysis, in some cases it had effects throughout the whole organization. To get the participants personal view on the method and modeling, we collected written feedback via evaluation forms.

Classroom experiment
One of the innovations of the UML profile was the use of specially designed icons related to security risks. Our first investigation aimed to gather support for this design decision by comparing models with and without special icons to identify differences. The special icons should help inexperienced users to understand the language better and faster, and therefore master students in system engineering were used as subjects. (See also *Chapter 5* (summary) or *Appendix A*.)

Survey
After several field trials, the language had "stabilized" and it was time to make the final design decisions regarding the modeling style and its graphical representation in the form of concrete syntax. To identify the preferences of the users, we presented several ways of modeling the same scenarios to a group of subjects and asked them to select the ones they preferred. We also tried to learn from related research within other domains; in particular, we made use of recommendations from the field of from graphical information visualization. (See also *Chapter 5* (summary) or *Appendix G*.)

Modeling benchmarking test
The modeling benchmarking test consists of a set of core security risk scenarios, defined on the basis of field trials and more general security risk related information (*Chapter 6*). The collection of scenarios is an example of information gathered during a typical security analysis. Modeling the core security scenarios with the CORAS language may be seen as kind of experimental simulation in McGrath's terminology.

With respect to Figure 33, we note that the icon investigation scores high on precision, the investigation of modeling preferences scores on generality, and the SECURIS field trials and the modeling-benchmarking test are fairly strong on realism. However, these evaluations have focused on slightly different aspects of the same modeling approach. The CORAS security risk modeling approach consists of both a language (syntax + semantics) as well as a modeling guideline, all strongly connected to the CORAS security analysis method. It is therefore unfeasible to continue focusing on separate evaluations of the different parts, and instead consentrate on experiences from practical application of the full CORAS approach.

The validity of the structured semantics has been ensured by using diagrams from real field trials as input for the pattern-matching rules. Since the structured semantics covers every legal combination of constructs, the developers had to precisely define de permitted modeling options and styles. This also provided valuable input to the modeling guideline. The research has so far completed the set of rules, and the next step will be to evaluate the diagram translation with respect to understandability and more.

In the following tables, we present the set-up for the classroom experiment (the icon investigation) and the survey (investigating modeling preferences), including a description of subjects, material, analysis method, and how the results have been used.

Table 3 – Set-up for the icon investigation

Subjects:	25 master students in system engineering from the University of Oslo. The experiment took place as an exercise in the master course "Unassessable IT systems".
Material:	The icon experiment material consisted of two sets of security risk scenarios modeled with the UML profile, one using standard UML icons and one using CORAS icons, and a related questionnaire. 13 of the 25 subjects received models with CORAS icons, 12 received standard UML icons. Both sets were stereotyped[2] with stereotype names (see Figure 37). The material had three parts: • Part 1 and 2 consisted of questions related to whether the subjects

[2] In UML one uses stereotyping to label elements of different kinds. This can be either textual, with icons or both. "<<Actor>>" is an example of textual stereotyping that often is added to the pin-man icon in UML.

	were able to quickly identify specific elements like treatment, risk, threat agent etc. in a model (i.e. whether they are able to navigating the models). The questions were of the type "How many assets are explicitly modeled in this diagram?" • Part 3 focused on how the models were understood and interpreted. An example of the type of question used here is "Which unwanted incidents affect the asset "Web server" in this model?" For part 1 and 2 the subjects had only a limited time to answer the questionnaire (3 minutes and 1,5 minutes) to stress the point that they should quickly identify the different diagram elements. For part 3 they had more time available (15 minutes).
Measuring and analyzing:	The two groups of subjects were compared to see whether the one with CORAS icons received a higher score than the group without. In the same manner we analyzed whether one group managed to complete more questions than the other group, or used less time pr. model-question-set. Since the number of subjects that answered within each part of the questionnaire decreased for each part, in addition to an overall limited sample set, we used a non-parametric test to ensure that our statistical analysis was conservative. The non-parametric Mann-Whitney test was used with alpha level = 0.05.
Results used for:	To decide whether one should continue to use special graphical icons in the security risk modeling language because it had positive effects on the interpretation of the models.

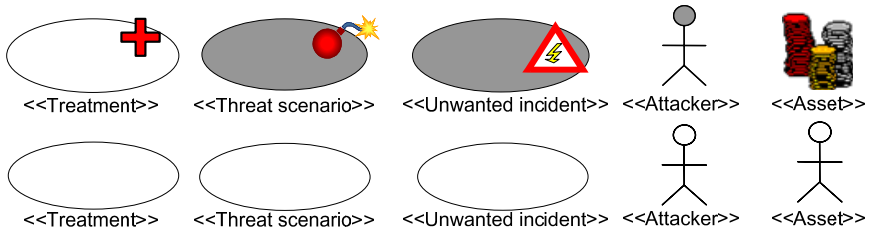


Figure 37 – Icons used in the icon investigation

Table 4 – Set-up for the modeling preferences investigation

Subjects:	33 professionals within system engineering: The majority were males between the age of 26 and 35 with 1-3 years of work experience.
Material:	The material was in the form of a questionnaire with alternative representations of a CORAS threat diagram. The diagram was a simplified version of a real threat diagram. The subjects were to prioritize between different modeling alternatives using the set-up shown in Figure 38. The survey was distributed via e-mail to people in various parts of the software industry.

Measuring and analyzing:	The investigation targeted three unsolved issues: • How should we visualize frequency measures in threat diagrams to ease graph navigation, especially the likelihood of single relations and combined relations? • What is the preferred way of visualizing vulnerabilities? • How can we visualize risks in the models to improve the understanding of this abstract concept? The null hypothesis and alternative hypothesis for all tasks were: H_0: the modeling alternatives are equally preferred by the subjects. H_1: the modeling alternatives are not equally preferred. The data was recoded into "0" and "1" to reflect which diagram that was preferred: a mark in one of the three fields to the left (see Figure 38) was transformed into "0" and marks in the three fields to the right were recoded as "1". The middle field was interpreted as missing data to obtain a cut-off point between the two sides. The frequency of 1's and 0's could thereafter be compared. The data was in the form of frequencies and not distributed normally; therefore the non-parametric Chi-Square test [143] was used to identify possible statistical differences between the representations with a significance level of 0,05. The Chi-Square test uses the assumption that the frequency of 1's and 0's are the same. SPSS v13.0 on Windows XP was used for the statistical tests.
Results used for:	Deciding how the final version of the security modeling language should look like with respect to design and layout for particular modeling tasks.

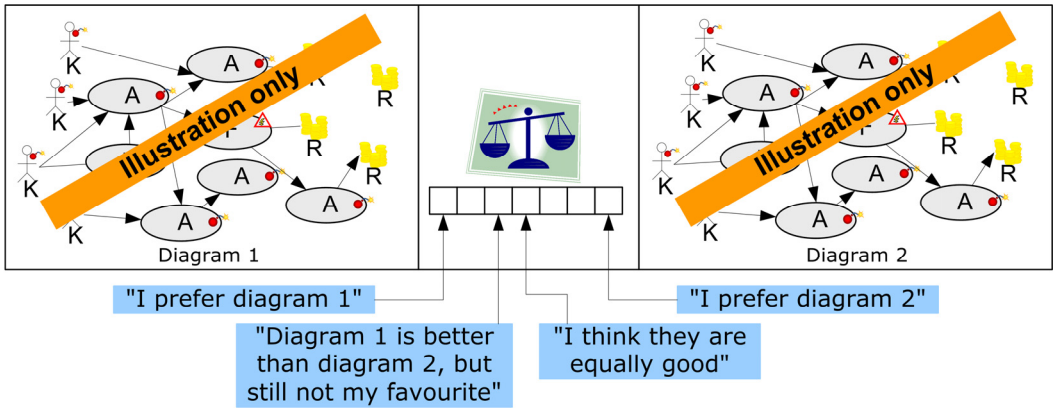


Figure 38 – Instructions for filling out the modeling preference tasks

4.2.3 Developing Modeling Instructions for the User

In order to use the security risk modeling language to its full potential there is a need for a modeling guideline, particularly targeting inexperienced users. The modeling guideline can be seen as the link between the language and the method, and changes to either one of them affects the guideline. The early use of the UML profile in the CORAS project had established a kind of modeling practice. This practice was applied in the SECURIS field trials, each time in a revised version based on experiences from the previous field trial.

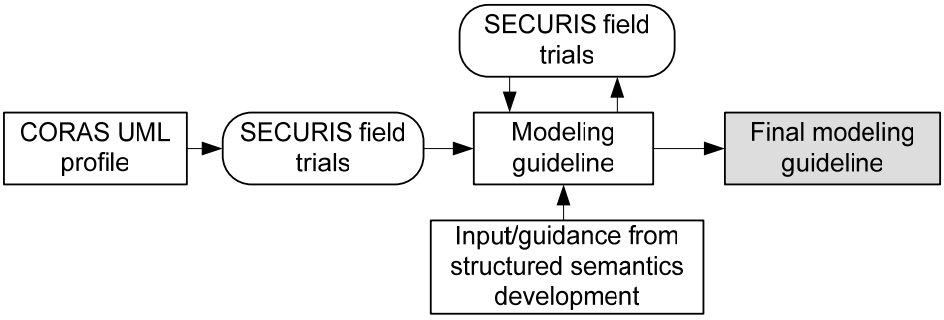


Figure 39 – Method used for developing the guideline

The evaluations of the modeling guideline are based on lessons learned from field trials, experimental simulation in terms of consistency checking against the structured semantics and modeling the benchmarking test case.

The SECURIS field trials differed a lot with respect to target, technology, scope and complexity. It was therefore unfeasible, and also impossible to define a strict guideline at an early stage. As the field trials progressed, the guideline was adapted to handle the various challenges that arose.

The development of the structured semantics provided valuable input to the modeling guideline. Before the structured semantics, the modeling style for each diagram type was a bit underspecified. The formal style of the structured semantics forced the developers to decide which combinations of constructs that should be allowed in a diagram. The guideline had to ensure that it did not conflict with the permitted and unacceptable constructs in the language. In this way, the structured semantics worked as a consistency, or quality check of the guideline. This work made the modeling guideline more complete and also helped define the exact content and purpose of each of diagram type.

In the same manner as the language itself must handle the core security risk scenarios (benchmarking test), it should be easy to arrive at these models by following the modeling guideline. This can be seen as a kind of experimental simulation where the use of the guideline is simulated. This simulation made it easier to identify inconsistencies, lack of guidance or similar, in the guideline.

It would have been extremely valuable to apply the guideline in a field study *without* the involvement of the researchers. After the field study, there should have been a qualitative interview of the users to obtain in-depth information about their experience with the guideline. However, this was not practical within the setting of the SECURIS project.

4.2.4 Developing a Way of Evaluating the Quality of Security Risk Modeling Approaches

The quality framework for security risk analysis modeling approaches has been developed to evaluate the new CORAS language, an also to make it easier for others to evaluate the quality and expressiveness of security risk modeling approaches.

Three main sources of information were used when developing the quality framework:

- SEQUAL [100-104] a quality framework for modeling languages within information modeling.

- Modeling needs within security analysis in general, particularly with focus on weaknesses in existing risk modeling notations.
- The use of the UML profile during field trials.

Evaluation of the framework has been made in two iterations; first integrated as part of its development, and second by applying it to the UML profile and the new CORAS language.

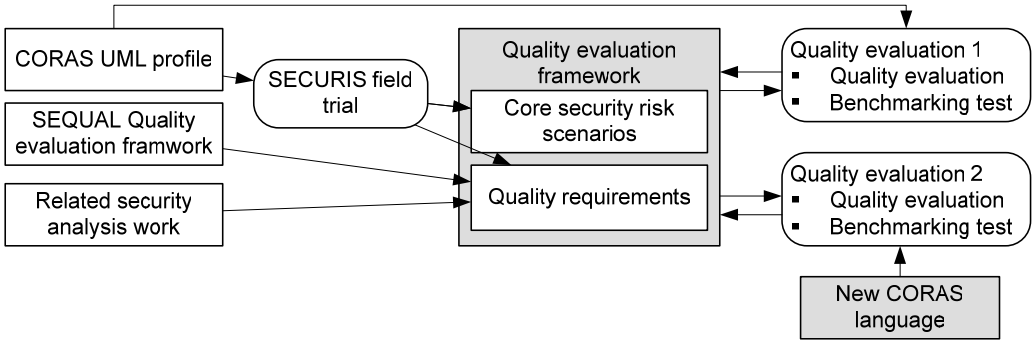


Figure 40 – Method used for developing the quality framework

When specifying the language requirements we used SEQUAL as basis. SEQUAL has been developed for comparing modeling languages, identify areas where a language needs improvements etc. In addition to the general quality requirements, more security analysis specific requirements were included into the SEQUAL structure. These were based on relevant modeling needs of existing security risk- and risk analysis methods. A security risk modeling language must be able to express realistic and relevant security risk scenarios. In the SECURIS field trials we have gained experience that we have used to construct a complete example of informal security analysis documentation called "the core security risk scenarios". (See also *Chapter 6*). The evaluation of the quality framework during its development is based on non-empirical evidence. Each requirement is followed by a characterization of the rationale for its inclusion into the framework (see *Chapter 6*). This makes it possible for a potential user of the framework to understand the background for the requirement and judge the relevance of a given requirement.

To evaluate the final quality framework it was first applied to the UML profile (See summary in *Chapter 7*, [57]). The results were important to the problem analysis (See *Chapter 2*) since it gave valuable insight into the weaknesses and strengths of the UML profile, and consequently which parts that we had to improve in the new CORAS language. The second evaluation of the framework was to apply it to the new CORAS language (*Chapter 9* and *Appendix H*). These two evaluations may be classified as experimental simulations where the framework is tested against real modeling approaches. After two evaluations, there were indications that some of the requirements originating from information modeling and some of the early requirements to security risk analysis modeling, were not as relevant as they had seemed[3]. An example is the general requirement that all concepts in a language must have a special symbol. Experience with the CORAS method in practice, does not support the request for a special symbol for e.g. "consequence", - but we cannot exclude that other approaches may need it. Because of this, we chose to keep these requirements in the framework since they may be relevant to other kinds of security risk modeling approaches than CORAS.

[3] The requirements that have been characterized as less relevant are listed in the framework description in *Chapter 6*, and also in *Appendix H*.

5 Initial Investigations

This chapter presents the initial investigations the results of which heavily influenced the design of the CORAS language. We first conducted a study of the interpretation of the conceptual model already developed in the CORAS project (5.2.1) prior to the start-up of our thesis work. On the basis of this investigation we came up with a revised conceptual foundation. We later conducted a second investigation focusing on improving finding ways to improve the understandability of certain important concepts (5.2.2). The last of the initial investigations was an experiment testing whether to use specialized graphical icons (5.3).

We start by explaining the original conceptual model developed in the CORAS project (5.1).

5.1 The Original Conceptual Model of the CORAS Project

Security analysis uses a terminology that includes general risk analysis concepts as well as more specialized security analysis terminology. A conceptual model may be used to explain how a concept like risk relates to for example threat or vulnerability. In the CORAS project, the conceptual model also played an important role as a kind of abstract syntax for the CORAS UML profile (presented in Sect. 3.5.1).

The original conceptual model of CORAS in Figure 41 using the class diagram notation from the Unified Modeling Language (UML) [125]. The associations between the elements have cardinalities that say how many instances of one element that can be related to one instance of the other (example: "a stakeholder has at least one and maximum an infinite number of assets; and an asset belongs to only one stakeholder").

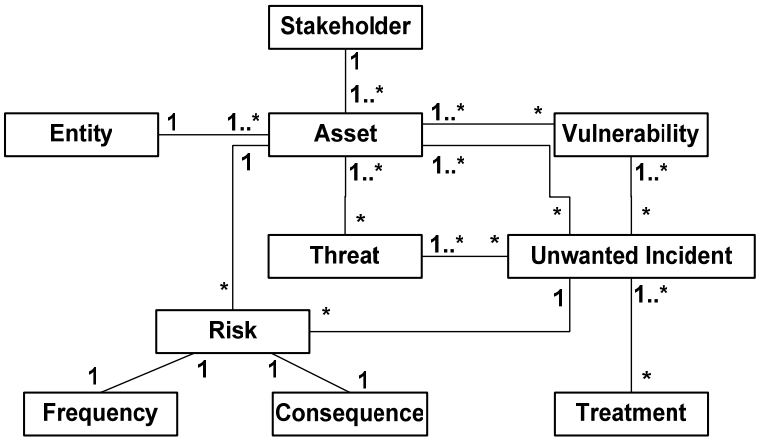


Figure 41 – The original conceptual model of CORAS

The concepts used in the model are defined as follows:

- **Stakeholders** are those people and organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity regarding the target system [7].
- An **asset** is something to which a stakeholder directly assigns value and, hence, for which the stakeholder requires protection [54]. A stakeholder wants to protect his/her assets from being compromised.

- An **entity** is a physical or abstract part or feature of the target of evaluation that becomes an asset when assigned value by a stakeholder, for example a service provided by the target [112].
- A **vulnerability** is a weakness of an asset or group of assets which can be exploited by one or more threats [76].
- An **unwanted incident** [54] is an event that may harm one or more assets and something one wants to prevent. An unwanted incident is the result of a threat exploiting a vulnerability.
- A **threat** is a potential cause of an unwanted incident [76] i.e. someone or something that wants to harm, remove or interfere with the asset.
- A **risk** is the chance of something happening that will have an impact upon objectives (assets) [7]. A risk is the combination of an unwanted incident, the chance of it happening and its consequence.
- **Consequence** is the outcome of an event expressed qualitatively or quantitatively, being a loss, injury, disadvantage or gain [7].
- **Frequency** a measure of the rate of occurrence of an event expressed as the number of occurrences of an event in a given time [7].
- A **treatment** is the selection and implementation of appropriate options for dealing with risk [7].

5.2 Investigating Concepts and Terminology

5.2.1 The First Terminology Investigation

As described in the previous section, the CORAS method and UML profile had a conceptual foundation defined in a UML class diagram. Several of its concepts and definitions are used in daily language, but often in a different meaning and with no link to its original context. For instance, many of us have probably used the phrase "I won't take the risk" without giving the definition of risk any thought.

The purpose of the first terminology investigation was to identify the parts of the CORAS conceptual model that are particularly difficult (or straightforward) to understand. The assumption for our investigation was that a precisely defined conceptual model would reduce the misunderstanding related to ambiguous terminology in a security analysis method. The questionnaire that was used tested definitions of concepts, as well as how the different concepts relate to each other (for more about the set up we refer to Sect. 4.2.1*)*.

The results showed that few subjects had problems with the relation between a stakeholder and an asset, or between a threat and a vulnerability. These concepts are part of the daily language and most people have an intuitive understanding of them. Concepts like frequency, consequence and probability created more confusion and a large majority was not aware that probability is kind of frequency. The results showed the importance of using terms that are intuitively understood, not conflicting with the daily language.

On the basis of these results we propose a set of changes to the original model (Figure 41) which are implemented in Figure 42:

- To avoid confusion about frequency, probability and consequence we have included probability as a specialization of frequency.

- In order to emphasize that a risk consists of a frequency value, consequence value and an unwanted incident, these terms have been grouped together in a logical manner that illustrates how they are components of a risk. The black diamond symbolizes that if a risk is eliminated this will also remove the frequency and consequence values (composition). The white diamond means that the unwanted incident is a part of the risk, but if the risk is eliminated the unwanted incident may still exist in other risks (aggregation). The direct relation between treatment and unwanted incident has been removed and instead we have connected treatment to risk. By doing this we specify that a treatment is directed towards a risk, but not whether it targets a vulnerability, a threat, an unwanted incident or a combination of these.
- The association between asset and risk was a major source for misunderstandings. Nevertheless, we have decided to keep the relation because we believe it is important. The subjects seem to think of an unwanted incident in the way we use the term risk, and by removing the direct association between asset and unwanted incident we hope that this misunderstanding will be less common.
- We chose to remove "entity" from the model. The term was often misunderstood (48.9% incorrect or uncertain answers). In addition, it is not natural to speak of general entities in a structured brainstorming, only entities assigned value which then are called assets.
- The association between asset and vulnerability has been changed from a regular relation to become a part of the asset in the sense that an asset can have a vulnerability.
- In the original conceptual model it is not explicitly modeled that "a threat exploits a vulnerability to initiate an unwanted incident". This is an intuitive relation that achieved a high percentage of correct answers and we feel it is correct to make it clearer than it was in the original model.
- The new model tries to emphasize that an unwanted incident is a part of a risk and therefore one or more threats are always associated with the risk through its unwanted incident.
- We have also chosen to highlight concepts grouped together in the form of compositions and aggregations, i.e. vulnerability is tightly connected to asset, and risk is a concept that includes three other concepts.

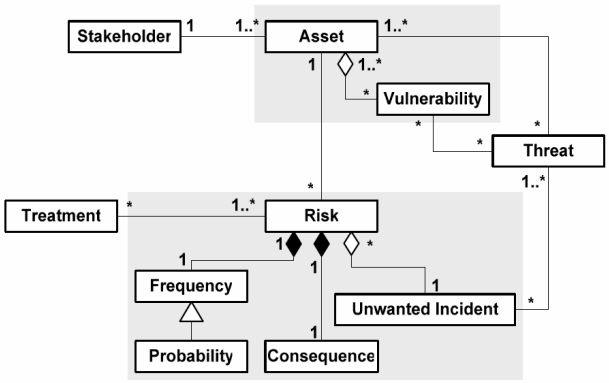


Figure 42 – The first revision of the conceptual foundation

5.2.2 The Second Terminology Investigation

The second terminology investigation considered how the revised conceptual foundation was understood among professionals and students within software engineering and risk analysis. The subjects had various competences in one or more of the following domains:

(1) system modeling and design,
(2) risk analysis and system development in the safety domain and
(3) system development and security.

The investigation used a similar set up as the first terminology investigation. The model in Figure 42 had after the previous investigation been extended with the concept **likelihood**. Likelihood was in [7] defined as a qualitative description of probability or frequency, and we decided to include it as a specialization of frequency.

The findings showed major differences between the various risk-related concepts. Asset and vulnerability related questions received the highest number of correct answers, while the frequency related obtained the lowest number. Many of the terms we have in our conceptual model (Figure 42) are well understood, even by people without training in risk analysis. Being more experienced in risk analysis, the professionals obtained a higher number of correct answers than the students.

The results from this study show that the revised model successfully reflects the subjects' common understanding of the risk analysis terminology. Nevertheless, we made some adjustments based on the findings from this investigation:

- To increase the awareness of non-human threats we have specified threat into human threat and non-human threat.
- The specialization of frequency into probability and likelihood was not satisfactory. According to [6] likelihood is a general descriptions of frequency or probability and can be expressed *both qualitatively and quantitatively*. Even though likelihood covers the other terms, it has turned out as one of the most difficult concepts to understand. The conceptual foundation will therefore not distinguish between the frequency measures (probability/frequency/likelihood), but leave it to the security analysis team to decide which one to use.

The UML 2.0 class diagram in Figure 43 shows the final model (the shaded areas are not part of the UML notation and merely highlight concepts that are closely related to each other).

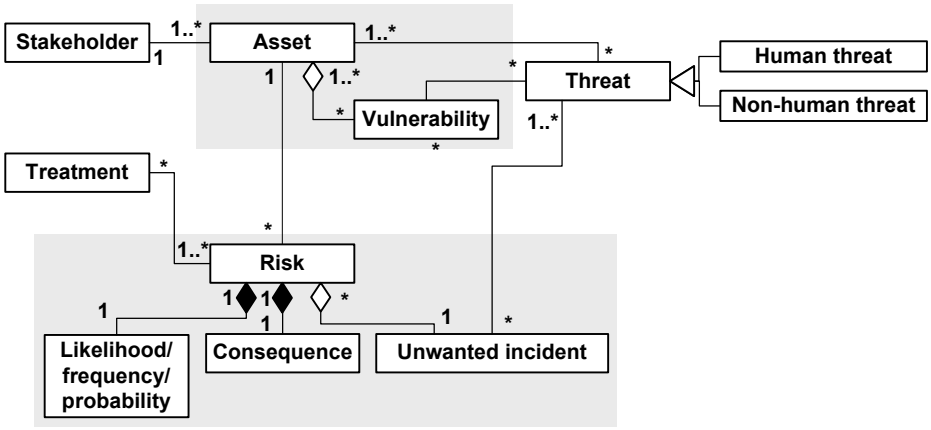


Figure 43 – The final revision of the conceptual foundation

5.3 Investigating the Use of Specialized Graphical Symbols

In addition to the terminology investigations, we explored the effect of using graphical, risk related symbols in risk modeling. This investigation is called the icon experiment.

The UML use case notation has its own set of standardized icons, but, the UML profile is recognized by its specialized graphical icons that symbolize the different terms in the conceptual model. The icons were believed to make the models easier to read and understand. Often graphical icons are seen as merely "decoration" just to make the models look nicer, but it is often believed that they may also improve the understandability of the models.

In this experiment we investigated whether the use of specialized icons make the models easier to understand. The difference between the standard UML icons and the specialized CORAS icons are shown in Figure 44.

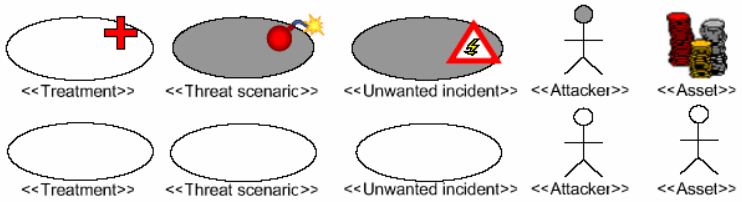


Figure 44 – Standard UML icons and the UML profile symbols

The results from the icon experiment showed that stereotyping with CORAS icons vs. UML icons improves the speed of navigation, i.e. identification of specific model elements. The results showed a statistical significant difference in the number of questions completed in favor of the group using CORAS icons. The icons did not affect the correctness of interpretation of models. Hence, the number of misinterpretations did not increase.

The conclusion from our icon experiment is that using the specialized graphical icons helps the participants in a structured brainstorming session to identify the same model elements faster than if standard icons are used without increasing the number of misinterpretations.

6 Language Requirements Defined within a Quality Framework

In this section we present our requirements to an ideal security risk modeling language in a framework that focuses on the quality of the language. This quality framework consists of two parts, first we describe the core modeling needs related to the different phases in the security analysis process. Then we use a quality evaluation framework for modeling languages called SEQUAL to structure and identify the detailed requirements.

In the following terms like *model, conceptual model, modeling language* and *diagram* are used. A model is a collection of diagrams that describes a system. The diagrams can show different views of the system's architecture, functions, processes etc. and are made with a modeling language. Figure 45 illustrates the relationships between these terms. A conceptual model is a description of important concepts, or terms, and how they relate to each other.

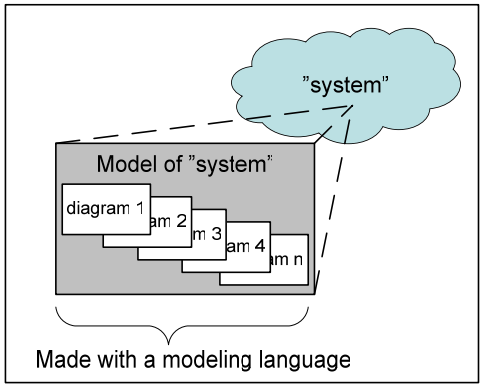


Figure 45 – Relationships between system, model and diagrams

To our knowledge it does not exist any predefined "benchmarking" scenarios we can use to evaluate a security risk modeling languages, therefore we have defined a set of test scenarios representing typical modeling needs identified through experience from several industrial security risk analyses. The phases in CORAS have different purposes and therefore different modeling needs. In the following we give an overview of the general modeling needs for each of the phases and a description of the core modeling needs using a security risk case as a benchmarking test. This test case inspired by, but not similar to, the field trials and should therefore represent realistic security analysis information. The information is structured according to which phase it is gathered in, using the CORAS method. The idea is that a good security risk modeling language should be able to model all aspects of this information, which we call the core security risk scenarios.

6.1 The Core Security Risk Scenarios

In order to evaluate a security risk modeling language we need a benchmarking case representing core security risk scenarios to test the notation against. Through experience from several major security analyses in the SECURIS project we have gathered typical security analysis information into a complete set of scenarios from (1)-context establishment through (2)-risk identification, (3)-risk estimation, (4)-risk evaluation and (5) treatment identification. For each of the five phases we explain what the purpose of the phase is and what tasks it includes. Then we present relevant security analysis information

that may be modeled graphically using a security modeling language. The characteristics of the core security risk scenarios are among others:

- There are two stakeholders of the system (one client, one other interested party. The distinction lies in whether the stakeholder is economically responsible for the analysis (e.g. a client like a company) or not (e.g. other interested party like the authorities). Multiple stakeholders in a security analysis makes it more complicated when it comes to judging: *what are the assets of the analysis?* And *what risk level can be accepted*?
- Assets may depend on each others, meaning damage to one asset may indirectly damage another asset. This has also been addressed in the core security risk scenarios.
- Often the statistical material on which to make judgments about how often an incident may occur is limited. In the example case we have used an approach that does not require precise, numerical input for the likelihood of unwanted incidents.
- When a risk is found acceptable, it should be documented but kept out of the treatment identification phase to ensure enough focus on the unacceptable risks.

6.1.1 Phase I: Establishing the Context

In this phase it is important to obtain an understanding of the target of analysis and its assets. The context establishment also focuses on the stakeholder's main concerns regarding target vulnerabilities and threats. The following aspects needs to be modeled (in one or more diagrams)

- Target overview diagram: an overview of the target (system or part of a system) that will be analyzed, annotated with the stakeholder(s), the assets, the main vulnerabilities and the main threats

An overview diagram of this type will help scoping the analysis during the preliminary security analysis and ensure that the correct level of details is established at an early point in time.

The assets are important in the CORAS method and therefore it can be useful to model them explicitly in a separate diagram:

- Asset diagram: describing the assets, how they relate and a ranking of their perceived importance. Assets that are affected by risks often have relations to other assets which indirectly may be affected. Even if some assets are defined to be out of the scope of the analysis, it is useful to model them to see the overall asset-picture.

6.1.1.1 Core Security Risk Scenarios in Phase I

Target description: The target of analysis is a web portal that serves as a communication medium between ordinary citizens and various public entities. The information provided is confidential personal information. The company that develops the portal will gather information from several databases within the public entities. The users authenticate themselves to the web portal using a password and username, while the authentication mechanism is simpler for the developers. Inside the developers network one uses a simple "remember-this-computer" mechanism to access the portal without being prompted for username and password each time. The users are free to set their own passwords without restrictions. Due to the importance of the information provided to its users, the service must be available 24/7. The web portal will gradually put into service and the security analysis will look at the security during development, testing and maintenance.

Stakeholders, clients and other interested parties: There are two stakeholders in this case where the company management is the *client* of the analysis, i.e. the one paying for the analysis. The other stakeholder is the central authority who is considered to be one of the "other parties". Other parties are not paying for the analysis itself, but have the authority to set requirements to the risk acceptance levels. In this case the analysis has been initiated as a consequence of the security requirements from the central authorities and therefore they are included in the analysis:

- **Company management (CM)** that develops and delivers the service, and therefore bears all the costs related to the development and maintenance.
- **Central authorities (CA)** that regulate what information the service should provide and how it should be protected. They have the authority to close down the service if it fails to fulfill their regulations.

Assets: The assets related to the target of analysis are ranged according to its values to the client and other parties. The potential damage a threat may cause to the asset (lost asset value) is not specified in details but described shortly in the parentheses:

The company management's assets:

- **CM1 – Users personal information** (damage is measured by the type of information disclosure, for example major = the information is available to the public for one day, medium = the information is available to the public for one hour, minor = the information is available to an unauthorized employee in the company for a week)
- **CM2 – Company reputatio**n (damage is measured by the type of negative media publicity in, TV, radio, large newspaper or small newspaper, negative rumors etc.)
- **CM3 – Availability of service** (damage is measured as down time of the service)
- **CM4 – User efficiency** (damage is measured as the increased effort needed to use the functions provided by the service)

The central authorities' assets:

- **CA1 – Users personal information** (measured as above)
- **CA2 – Availability of the service** (measured as above)
- **CA3 – User efficiency** (measured as above)

The assets are not independent, meaning if one asset is harmed it may affect other assets. An in some cases an asset has to be harmed first before another asset can be harmed. In this case the first asset is called a *direct asset* and the second an *indirect asset*. The identified relations between assets are:

- Company reputation (CM2) can only be harmed if one of the other assets is harmed first. This means that CM2 is an indirect asset.
- Within the scope of the analysis, only damage to the availability of service may affect user efficiency (CM4, CA3), making User efficiency (CA3) an indirect asset.

Measures: One needs to define several different measures like asset values, likelihood of risk and consequences of risks:

- **Assets**: may be ranged according to their perceived importance, monetary values etc. but is not required.
- **Likelihood** is measured qualitatively in three categories: *seldom* (= 1 time per 5 years or less), *sometimes* (= more than 1 time per 5 years and less than 1 time per year) and *often* (= 1 time per year or more). The categories may be mapped to intervals of probabilities if such data is available and appropriate to use.

- **Consequence** is measured qualitatively in three categories: *minor, moderate* and *major*. The consequence categories should to be mapped to actual damage for each asset. A "minor" damage could in some cases mean "system down in 2 minutes" while in other cases a minor consequence could be "system down in 4 hours". This should be specified in accordance with the client and other parties. In this generic case we will only use the category names since substituting them with numbers or text would have no impact on how it is modeled graphically.
- **Risk value** is measured in three categories: *low, medium* and *high.* The risk value is based on a combination of likelihood and consequence, either as a risk matrix (like in this case) or a risk function that computes a value based on probability and consequence (this requires the consequence to be measured quantitatively).

	Likelihood		
Consequence	Seldom	Sometimes	Often
Minor	*Low*	*Low*	*Medium*
Moderate	*Low*	*Medium*	*High*
Major	*Medium*	*High*	*High*

Figure 46 – Risk value matrix

- **Risk reduction** is measured in terms of decreased risk value (based on reduction in consequence and/or likelihood).

Risk evaluation criteria: The tolerance levels, or acceptance levels, for risks against specific assets are decided already at this stage in the analysis. In this case the two stakeholders, company management and the central authority value the assets differently, and consequently have different risk acceptance levels. These levels are later used during risk evaluation to decide which risks that can be accepted and which that need to be treated.

Table 5 – Risk evaluation criteria

Asset	**Max accepted risk level**	
	CM (Client):	**CA (Other parties):**
User's personal information	low risk	low risk
Company's reputation	medium risk	any risk
Availability of service	medium risk	medium risk
User efficiency	high risk	medium risk

6.1.2 Phase II: Identifying Risks

This phase needs models of two main types: system models (UML etc.) and risk models. The first type depends on the target type and is not part of the core modeling scenarios. The risk models must include one or more diagrams that include:

- the threats and threat scenarios related to assets.
- the vulnerabilities and which threats that can exploit them (i.e. the threat's way "into" the target).
- the unwanted incidents the threats may cause

6.1.2.1 Core Security Risk Scenarios in Phase II

Threats: There are both human and non-human threats, that either can be threats by an accident or have more deliberate motives (these distinctions are in accordance with the security standard ISO/IEC13335 [76]):

- **Company employee (human, accidental)**: an employee may make a mistake causing an unwanted incident or unintentionally infect the server with malicious code during an update.
- **Company employee (human, deliberate)**: an employee may use his or her access rights to intentionally cause an unwanted incident.
- **Hacker (human, deliberate)**: a hacker may want to harm the users or the company for fun or for economical reasons (e.g. blackmailing).
- **Internal infrastructure (non-human)**: hardware or software, part of the service, may fail and initiate unwanted incidents.
- **External resources (non-human)**: resources that deliver data to the service.
- **Virus attack (non-human)**: an environmental circumstance outside the company's control.
- **Web portal service user (human, accidental)**: a user may for example use the service incorrectly

Vulnerabilities:

- **Insufficient authentication mechanisms**: within the company development team the authentication mechanism only requires a username and a password, no secure ID or similar identification. Inside the company network the user is not prompted for username or password when applying the "remember-me" function.
- **System design weakness**: the development environment used by the company has very few restrictions on what an employee may modify and does not provide warnings related to critical updates.
- **Unsecured WLAN**: the company has an open WLAN in their development environment which is possible to detect outside the company building.
- **Too simple password**: there is no control on whether the user of the portal changes his initial password, and if he does there are no rules for how the new password should look like (length and combination of letters, numbers)
- **Shared infrastructure resources**: the service runs on hardware or software that is shared with other less critical services. This means that if one of the other services encounter a problem it may affect the service.
- **Low robustness**: in cases of high traffic to the portal, the server tends to degrade in performance and response time increases.
- **External resource failure:** a resource that provides data to the web portal service may fail, and the service is dependent on the availability of these databases.
- **Internal hardware or software failure:** the internal infrastructure may fail due to hardware or software errors.
- **Insufficient logging**: access and modification of user's personal information is insufficiently logged, meaning that one cannot be sure who has made the changes (i.e. the user or one of the company's employees).
- **Unclear security update routines**: security updates are communicated via e-mail or the intranet and the individual employee is responsible for keeping his or her computer updated.

Unwanted incidents: The unwanted incidents that may happen are listed below followed by a description of the threat scenarios that may lead to the incident (the threat is marked with bold fonts):

U1: Disclosure of users' personal information:

1. An employee may unintentionally modify the system making it disclose personal information of one or more users to all the other users.
2. An employee may use her or his job privileges to access users' personal information and use it for blackmailing (no internal logging in the company).
3. An employee in the company may use another employee's computer that has the "remember-me" function enabled and thereby get access to users' personal information.
4. A hacker may attack the service via the WLAN and eavesdrop to the data transmission.
5. A hacker may exploit the simple password policy to access users' personal information.

U2: Unauthorized modification of users' personal information:

1. An employee may use her or his job privileges to modify users' personal information without being logged.
2. An employee may unintentionally update the system causing it to modify or delete users' personal information.
3. A user may enter information repeatedly if the service's response time is too long, accidentally making the information incorrect.
4. A hacker may attack the data transmission via the WLAN and tamper with users' personal information.
5. Virus attack on the service may cause the server to crash, deleting all active user sessions and their previous data modifications leaving the information partly incorrect.
6. A user may unintentionally enter wrong or incomplete information to the service, affecting already stored data. Without any logging it is impossible to prove who is responsible for the changes.

U3: Unavailability of service due to hackers:

1. A hacker may cause a denial-of-service-attack to the service making it unavailable to both customers and employees.
2. A hacker may use the WLAN to obtain a password and a username and then use this to log in as an employee with authorized access to servers, and therefore be able to tamper with the server or databases.

U4: Unavailability of service due to infrastructure failure:

1. Hardware or software, part of the service infrastructure, may fail or malfunction and make the service fully or partly unavailable.
2. External sources of information may fail or malfunction making the service unavailable.

U5: Unavailability of service due to malicious code:

1. An employee may unintentionally infect the server with malicious code using a false security or operative system patch.
2. A virus attack may cause extensive traffic and thereby make the service unavailable.

U6: Damage to company reputation:

1. If users' personal information is disclosed to media it may harm the company's reputation.
2. If the possibility to modify users' personal information is disclosed to the press it may harm the company's reputation.
3. If the service is unavailable it may harm the company's reputation.

U7: Reduced user efficiency:

1. If the service is unavailable it may reduce the users' efficiency.

6.1.3 Phase III: Estimating Risks

Estimating risks is to provide likelihood and potential consequence estimates for each risk. The modeling needs in this phase are:

- A description of the risks that includes both the threat's method(s) and which assets that are harmed. (giving a possibility to annotate threat scenarios and unwanted incidents with likelihood estimates.)
- A description of the associations between an unwanted incident and an asset, representing risks. This can be annotated with the most likely consequence value (e.g. "loss of 1-10K €", "10-20% reduced user efficiency") reflecting the scale the asset is measured in.

If the proper data is available one can apply statistical methods and conventional modeling notations like fault trees [66] and event trees [64].

6.1.3.1 Core Security Risk Scenarios in Phase III

Each risk is given a consequence and likelihood estimate as shown in the following table.

Table 6 – Risks with consequence and likelihood estimates

Risks	Asset harmed	Consequence estimate	Likelihood estimate
R1CM) Disclosure of users' personal information	CM1	Major	Sometimes
R1CA) Disclosure of users' personal information	CA1	Major	Sometimes
R2CM) Unauthorized modification of user's personal information	CM1	Major	Seldom
R2CA) Unauthorized modification of user's personal information	CA1	Major	Seldom
R3CM) Unavailability of service due to hackers	CM3	Major	Seldom
R3CA) Unavailability of service due to hackers	CA2	Major	Seldom
R4CM) Unavailability of service due to infrastructure failure	CM3	Moderate	Sometimes
R4CA) Unavailability of service due to infrastructure failure	CA2	Moderate	Sometimes
R5CM) Unavailability of service due to malicious code	CM3	Moderate	Seldom
R5CA) Unavailability of service due to malicious code	CA2	Moderate	Seldom
R6CM) Damage to company reputation	CM2	Moderate	Seldom
R7CM) Reduced user efficiency	CM4	Minor	Sometimes
R7CA) Reduced user efficiency	CA3	Moderate	Sometimes

6.1.4 Phase IV: Evaluating Risks

When evaluating the risks one decides which ones that are most serious. The risks are prioritized according to their gravity and the ones that cannot be tolerated are subject to treatment identification.

In this phase we need to model an overview of the acceptable and unacceptable risks.

6.1.4.1 Core Security Risk Scenarios in Phase IV

Table 7 – Risks with risk values

Risks	Asset harmed	Computed risk value*
R1CM) Disclosure of users' personal information	CM1	High risk
R1CA) Disclosure of users' personal information	CA1	High risk
R2CM) Unauthorized modification of user's personal information	CM1	Medium risk
R2CA) Unauthorized modification of user's personal information	CA1	Medium risk
R3CM) Unavailability of service due to hackers	CM3	Medium risk
R3CA) Unavailability of service due to hackers	CA2	Medium risk
R4CM) Unavailability of service due to infrastructure failure	CM3	Medium risk
R4CA) Unavailability of service due to infrastructure failure	CA2	Medium risk
R5CM) Unavailability of service due to malicious code	CM3	Low risk
R5CA) Unavailability of service due to malicious code	CA2	Low risk
R6CM) Damage to company reputation	CM2	Low risk
R7CM) Reduced user efficiency	CM4	Low risk
R7CA) Reduced user efficiency	CA3	Medium risk

*The risk value is set using the risk matrix in Figure 46.

Comparing the risk values with the risk tolerance levels from phase 1 (Table 5) gives the following risk evaluation (shown in Figure 47):

- Company management: R1CM and R2CM are higher than the accepted risk level.
- Central authorities: R1CA and R2CA are higher than the accepted risk level.

	Likelihood		
Consequence	Seldom	Sometimes	Often
Minor		R7CM	
Moderate	R5CM, R5CA, R6CM	R4CM, R4CA, R7CA	
Major	***R2CM, R2CA***, R3CM, R3CA	***R1CM, R1CA***	

Figure 47 – Risks placed in the risk evaluation matrix

6.1.5 Phase V: Identifying Treatments

The purpose of this phase is to decide which risks that need treatments, i.e. are too serious to be left unattended and what kind of treatments. In this phase it is useful to have an overview diagram of the risks (including the vulnerabilities, threats and unwanted incidents involved) with risk values as input and extend it with various treatments options.

One uses the risk value to decide which risks that needs to be treated. The client (the person or organization that initiated the analysis in the beginning, often identical to the stakeholder, but not always) decides the risk tolerance level.

6.1.5.1 Core Security Risk Scenarios in Phase V

Treatment options:

- **TO1: Upgrade to more robust infrastructure solution** that have lower failure rate.
- **TO2: Install redundant system** that will take over in case of infrastructure failure or attack.
- **TO3: Install improved firewall** that will make it more difficult for a hacker to find vulnerabilities.
- **TO4: Install intrusion detection system** that will detect the attack rapidly and make it possibly to switch to manual routines.
- **TO5**: Remove the possibility employees have to **access other users' personal information**.
- **TO6: Remove the unsecured WLAN**.
- **TO7: Remove the "remember me"-function** for employees.
- **TO8: Involve users** in the development of an improved system.
- **TO9**: **Implement logging facilities**.
- **TO10**: The user and the service provider should share the responsibility for modification of data due to user errors in combination with slow response from the service. This should be stated in a **legal contract**.

Treatment effects: Estimated effects on likelihood and/or consequence are shown in the table below. These are only for example purposes and have not been estimated on basis of any expert judgments or other sources of information.

Table 8 – Treatment effects

	R1CM	R1CA	R2CM	R2CA
TO1	-	-	Reduce likelihood	Reduce likelihood
TO2	-	-	Reduce consequence	Reduce consequence
TO3	Reduce likelihood	Reduce likelihood	Reduce likelihood	Reduce likelihood
TO4	Reduce consequence	Reduce consequence	Reduce consequence Reduce likelihood	Reduce consequence Reduce likelihood
TO5	Reduce likelihood	Reduce likelihood	Reduce likelihood	Reduce likelihood
TO6	Reduce likelihood	Reduce likelihood	Reduce likelihood	Reduce likelihood
TO7	Reduce likelihood	Reduce likelihood	Reduce consequence	Reduce consequence
TO8	Reduce likelihood	Reduce likelihood	Reduce likelihood	Reduce likelihood
TO9	Reduce likelihood	Reduce likelihood	Reduce consequence Reduce likelihood	Reduce consequence Reduce likelihood
TO10	-	-	Reduce consequence	Reduce consequence

The individual treatment options' effects on risk values are shown in the table below (unc hanged risk values means that the single treatment is not sufficient to reduce the risk value, "-" means the treatment is not applied for the risk). Since these estimates are included in the core security risk scenarios with the purpose of showing how they are dealt with in the models, they are only example estimates without a thorough rationale.

Table 9 – Treatment effects on risk values

	R1CM	R1CA	R2CM	R2CA
TO1	-	-	medium → low	medium → low
TO2	-	-	medium → low	medium → low
TO3	high → high	high → high	medium → medium	medium → medium
TO4	high → high	high → high	medium → medium	medium → medium
TO5	high → high	high → high	medium → medium	medium → medium
TO6	high → low	high → low	medium → low	medium → low
TO7	high → high	high → high	medium → medium	medium → medium

TO8	high → medium	high → medium	medium → medium	medium → medium
TO9	high → medium	high → medium	medium → low	medium → low
TO10	-	-	medium → medium	medium → medium

The final treatments selected for implementation are typically decided upon after a cost-benefit assessment of each of the treatment options. However, such an assessment is outside the scope of this core security risk scenario example.

6.2 Quality Requirements

Modeling is a commonly used technique within system development and there exist a number of modeling languages which are more or less suitable for this purpose. Graphical models are often used to ease the understanding of complex systems. In security analyses one deals with critical systems that may contain confidential data, provide critical services or have a high demand for availability. A correct understanding of the system and its risks is highly important, and this is where graphical models are found useful. The participants in the analysis normally have much competence on different parts of the target and may therefore view it differently. This may cause problems in understanding each other or problems in agreeing on the appropriate level of precision or scope of the analysis. By graphically modeling the target with its threats, assets and risks one can easily reduce the number of misunderstandings since this clarifies these aspects. The challenge is to find a modeling language that people understand, preferably suitable for use in a computerized modeling tool.

The quality of a modeling language depends on several factors, a language which is excellent for one task may be inappropriate for a different task. Through our experience with modeling in industrial security risk analyses, we have developed a set of detailed requirements to this kind of modeling language. To structure these requirements we have implemented them in the quality framework for modeling languages called SEQUAL developed by Krogstie, Sindre, Lindland and Sølvberg [100-104]. The framework can be used when selecting between different languages, investigating possible improvement areas for a language, or as the basis requirements to a new modeling language. The framework deals with both the quality of a particular model, and with the quality of modeling languages. In this work we will only use the part related to modeling languages, or what we call the language's "appropriateness factors".

The appropriateness factors of a modeling language are related to the modeling task definition, i.e. the goal of the modeling task (***G***), its domain (***D***), the knowledge of the people involved in the modeling task (***Ks, Km***), the interpretation of the models (***I***), the language that is used (***L***) and the tools (***T***) (illustrated in Figure 48). Figure 48 also shows the graphical model (model externalization, ***M***), but as mentioned above, this work will not go into the quality aspects of a concrete model. In this evaluation, the purpose is to evaluate the UML profile's appropriateness factors for security risk modeling in structured brainstorming sessions.

The six appropriateness factors are:

- Domain appropriateness: to be appropriate for the domain, the language should include all concepts necessary to express anything within the domain it is meant for.

- Participant language knowledge appropriateness: to be appropriate for the participants' language knowledge, the concepts and constructs in the languages should be as close as possible to the participants' understanding of the "real world".
- Knowledge externalizability appropriateness: to be appropriate for the knowledge externalizability the language should be able to express all aspects of the domain that the users are interested in.
- Comprehensibility appropriateness: to have an appropriate comprehensibility the language should be understandable for the users.
- Technical actor interpretation appropriateness: to be considered appropriate for the technical actors the language should have a syntax and semantics that a computerized tool can understand.
- Organizational appropriateness: to be appropriate for the organization the language should fit into existing technology, work processes and modeling methods that are used by the organization.

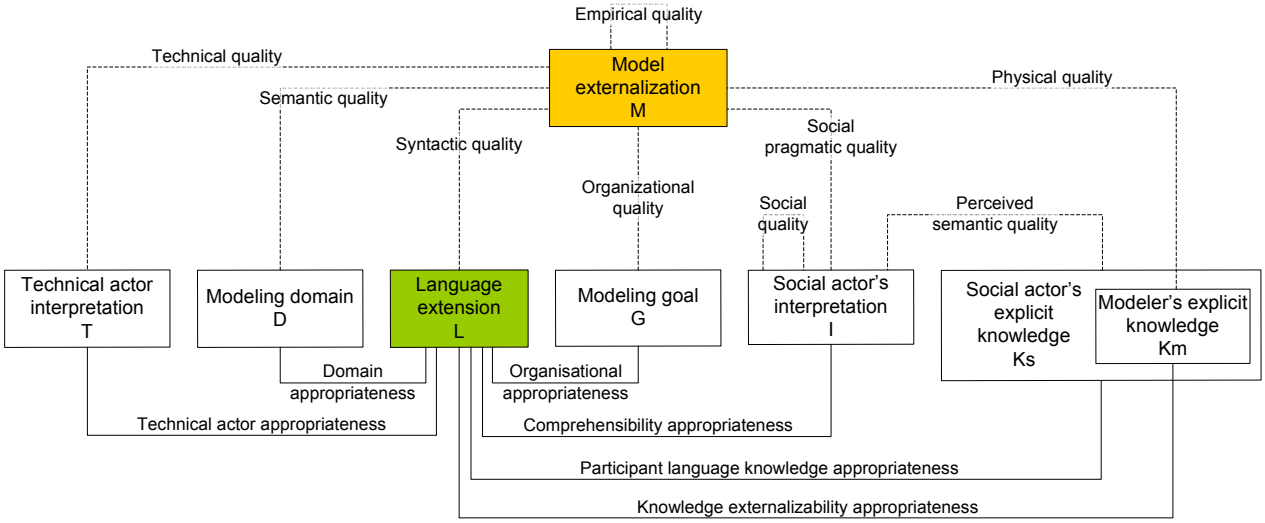


Figure 48 – The quality framework

6.3 Adapting SEQUAL to the Security Analysis Setting

Before assessing the appropriateness of a language, it is necessary to define the modeling task for which the modeling language should be used.

Table 10 – SEQUAL aspects

The goal of modeling (***G***)	the goals of the modeling task (normally organizational)
The modeling domain (***D***)	the domain, i.e., the set of all statements which can be stated about the situation at hand.
The relevant explicit knowledge (K_s) K_m (a subset of K_s)	the relevant explicit knowledge of the set of stakeholders being involved in modeling (the audience). A subset of the audience is those actively involved in modeling, and their knowledge is denoted Km.
The social actor interpretation (***I***)	the social actor interpretation, i.e., the set of all statements which the audience thinks an externalized model consists of.
The model externalization (***M***)	the externalized model, i.e., the set of all statements in someone's model of part of the perceived reality, written in a language.
The language extension (***L***)	the language extension, i.e., the set of all statements that are possible to make according to the graphemes, vocabulary, and syntax of the modeling languages used

The technical actor interpretation (***T***)	the technical actor interpretation, i.e., the statements in the model as 'interpreted' by different model activators (for example modeling tools).

In order to reflect the security analysis setting, or more specifically the structured brainstorming setting, the definition in Table 10 is translated from its general form into a more specialized form (Table 11). A structured brainstorming is a methodical "walk-through" of a target system with purpose of identifying as much information about the target as possible for each step. The brainstorming is a group exercise involving highly skilled people with competence of relevant parts of the target of analysis.

Table 11 – Modeling task definition in a security analysis setting

G: to reduce the effort needed to identify and understand the overall risk picture for the system assessed in a structured brainstorming, and thereby contribute to increased control of the risks for the organization
D: the security risks towards the system assessed.
K_s: the knowledge of the people who contributes to the models in the structured brainstorming. This will typically be experts on various parts of the system assessed, in addition to the analysis leader and the secretary. K_m: the knowledge of the security analysis leader (or analysis secretary) which gathers information from the participants and models it during the structured brainstorming.
I: the interpretation of models seen from the participant's point of view (e.g. system owner's, developer's and user's).
M: the model of the system's risks.
L: all the statements that is possible to make in the UML profile according to its definition.
T: the statements in the model that can be interpreted by other modeling tools or specialized security analysis tools.

In the following we discuss our requirements to a security risk modeling language for each of the appropriateness factor categories. Throughout the evaluation we will refer to the modeling task definition using the letters from Table 11 (modeling domain = *D*, modeling goal = *G* etc.). Within each category we have included a number of extra requirements based on experiences with the CORAS security analysis method in industrial field trials.

The requirements are not assigned explicit weights to show their importance since they all represent desired language features, instead we use the terms "must" and "should" to indicate their importance (requirements described with "must" are more required than those explained with "should"). The requirements are numbered sequential and "S-x" means that the requirement originates directly from SEQUAL, while "C-x" means that the requirement comes from user-experiences with the CORAS method.

6.3.1 Domain Appropriateness

The domain appropriateness of a modeling language relates to how much of a domain one is capable of modeling (defined in the language's internal representation) and how it is expressed in a diagram (defined in the language's external representation).

In information modeling there are several types of modeling perspectives to choose between. According to [104] there are 7 general modeling perspectives: *structural, functional, behavioral, rule-oriented, object-oriented, language-action-oriented* [168] and *role & actor-oriented modeling perspective*. Security risk modeling has many similarities

with information modeling. Information modeling often describes the behavior of a workflow or process, whereas security risk modeling deals with describing the "workflow" of a threat that initiates an unwanted incident. For security risk modeling we have identified the need of describing *how* incidents can happen, *who is* initiating them and *what* will be affected. There is also a need for showing more static relations that can specify dependencies between assets, treatments, risks and more. Based on this we require our modeling language to provide the following modeling perspectives:

Table 12 – Domain appropriateness: modeling perspectives

Requirements	Explanation
C-1 Structural modeling	***L*** must provide a structural modeling perspective. Structural modeling is used to illustrate for example how assets relate to each other, how one can group similar risks or treatments, how vulnerabilities relate to assets and more.
C-2 Behavioral modeling	***L*** must provide a behavioral modeling perspective. The behavior modeling perspective shows how a threat can initiate one or more unwanted incidents to harm assets, or how various treatments can be applied to risks and more.

SEQUAL distinguishes between *symbol* and *concept*: a concept is a phenomena/something one wants to express, while a symbol is the graphical notation used to model the concept.

In terms of requirements to domain appropriateness, ***L*** must support the concepts and relations according to the revised CORAS conceptual model for security analysis terminology (specified in *Chapter 5*). By enforcing this requirement we will insure that ***L*** covers all terms and relations that we find relevant for security analysis. To fulfill this requirement ***L*** must be able to express what is described in the explanation of the conceptual model, i.e. which concepts relate to each other and in what way.

Table 13 – Domain appropriateness: concepts

Requirements	Explanation
C-3 Asset	***L*** must include the concept "asset". *An **asset** is something to which an organization directly assigns value and, hence, for which the organization requires protection* [55]. The concept is very central in a security analysis. It should be up to the modeler to decide how much details the specification of an asset should include. Often an asset can be affected by other assets, for example a company's reputation is affected by the quality of the product they deliver, the service they provide, the employee's satisfaction etc. The modeler should have the option to create groups of assets that are similar or affect each other.
C-4 Vulnerability	***L*** must include the concept "vulnerability". *A **vulnerability** is a weakness with respect to an asset or group of assets that can be exploited by one or more threats* [76]. Vulnerabilities are critical parts of the system and important to establish early in the security analysis. Vulnerability can be composed into composite vulnerabilities according to the chosen level of detail. It would be useful to group similar vulnerabilities since they may be treated in the same way.
C-5 Risk	***L*** must include the concept "risk". *A **risk** is the chance of something happening that will have an impact upon objectives* [6]. Similar risks may be grouped and risk can be composed into composite risks.

C-6 Stakeholder	***L*** must include the concept "stakeholder". ***Stakeholders*** *are those people and organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity* [6]. By identifying a system's stakeholders one gets an overview of for example which people or systems that use it or depend on its functions.
C-7 Threat	***L*** must include the concept "threat". *A* ***threat*** *is a potential cause of an (unwanted) incident which may result in harm to a system or organization* [76]. It should be possible to specify whether a threat is human or non-human. Other more specified threats can be specified within these categories, e.g. deliberate threats and non-deliberate (accidental), but we do not find it necessary to define these as separate concepts in ***L***.
C-8 Unwanted incident	***L*** must include the concept "unwanted incident". An **unwanted incident** may result in harm to a system or organization [55]. In [76] called information security incident. An unwanted incident can vary in the level of details and ***L*** should provide the possibility to model both very detailed and more general. An unwanted incident may very well consist of more incidents.
C-9 Treatment	***L*** must include the concept "treatment". *A* ***treatment*** *is the selection and implementation of appropriate options for dealing with risk* [55, 76]. It should be possible to specify the treatment strategy. According to [76] treatment strategies can be categorized into four groups: 1) reduce likelihood of risk, 2) reduce consequence of risk, 3) transfer risk in full or part, 4) avoid risk or 5) retain risk. We do not require ***L*** to support exactly these categories, but there should be an option to specify a treatment in more detail if the user finds it necessary. In our opinion treatment strategies are of two types, either their purpose is to reduce the likelihood of a risk, or reduce its consequence(s). Transferring a risk partly or full can for example mean outsourcing the risky part to someone more qualified (reduce the likelihood of risk) or buying insurance against the risk (reduce the consequence of the risk). The strategy of avoiding a risk is really reducing its likelihood or consequence to zero. Based on this a treatment can either be regarded as *preventive* (i.e. reduce the likelihood) or *repairing* (i.e. reduce the consequence) and these interpretations should be included in the treatment concept in ***L***. The term ***safeguard*** defined as *a practice, procedure or mechanism that reduces risk* [55] are sometimes used instead of treatment. These terms are two names for the same concept. One could possibly argue that *safeguard* is more representative for a protection mechanism against threats that already exist in the system than treatment, which sounds like something applied after an unwanted incident. Establishing existing safeguards is relevant both in context- and treatment identification.
C-10 Likelihood/ frequency/ probability	***L*** must include the concepts "likelihood, frequency, probability". Likelihood, frequency and probability are all measures for variants of "how likely is it that this will happen". ***L*** must support all these measure-types. Likelihood is a qualitative or quantitative description of frequency or probability [6]. Frequency and probability is quantitative measures with a higher degree of precision. Frequency is an exact number of occurrences, while probability is a number between 0-1 where 0 = unlikely and 1 = will happen (in practice one normally says 50% instead of 0.5). Depending on the statistical data available for the security analysis the user should decide which measure to use.

C-11 Consequence	***L*** must include the concept "consequence". A consequence *is 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* [6]. Sometimes consequence is called "Impact" but have the same meaning: *the result of an unwanted incident* [76].
C-12 Threat scenario	***L*** must include the concept "threat scenario". A **threat scenario** is a description of how a threat may lead to an unwanted incident by exploiting vulnerabilities. A threat scenario must be able to express all from very detailed scenarios to vaguer scenario descriptions. A threat scenario can be a chain of threat scenarios, meaning that a threat scenario can lead to one or more scenarios depending on the level of details required. A threat scenario may be given a likelihood estimate. This is used in cases where the likelihood of an unwanted incident cannot be decided precisely but is best estimated from the likelihoods of each threat scenario that may lead to the incident.
C-13 The relations in our conceptual model (Chapter 5)	***L*** must support the explicit relations in the conceptual model, but it must also be possible to specify relations between concepts that are not part of the model (e.g. threat scenario). The symbols used to represent these relations are discussed in the symbol section. The following relations must be supported: 1. A stakeholder can be associated with one or more assets, but an asset can only be associated with one stakeholder 2. An asset is associated with at least one vulnerability, a vulnerability can be associated with more than one asset. 3. A threat must exploit (go via) at least one vulnerability in order to harm an asset. 4. A threat is associated with at least one unwanted incident via one or more threat scenarios. An unwanted incident is associated with at least one threat. 5. A threat is associated with at least one asset via threat scenario(s) and unwanted incident. An asset is associated with at least one threat via unwanted incident and threat scenario(s). 6. A threat is indirectly associated with a risk through its relation to unwanted incident (not explicitly shown in the model). This means that all threats are associated with at least one risk and every risk is associated with at least one threat. 7. An unwanted incident is indirectly associated with at least one asset through risk. In special cases one may experience that an asset is not related to any unwanted incidents, meaning that it has not been identified any risks for this asset. 8. A treatment is always related to one or more risks. 9. A treatment is indirectly related to an unwanted incident, threat, or vulnerability (or a combination of these) via risk (not explicitly shown in the model). Still it should be possible to model a treatment towards one of these concepts. 10. A risk is always associated with one asset, while an asset may be affected by many risks. 11. A risk always includes estimates of likelihood and consequence. 12. A risk always includes one unwanted incident, while an unwanted incident may participate in several risks, meaning that there must be a relation between risk and unwanted incident.

C-14 And/or operators	There must exist and/or operators in ***L***. In practice an unwanted incident often requires two or more threat scenarios to occur simultaneously before it occurs. To handle these cases we require ***L*** to provide "and" and "or" operators. If the outcomes of two or more events are joined in an and-operator they all are required to initiate a new event. If the operator is an or-operator it is sufficiently that only one event occurs to initiate a new event. Operators will provide additional information about the behavior in the threat scenarios, specifying the alternative ways a threat can behave. Operators will also increase the possibilities of using conventional fault tree techniques to compute the likelihood of risks.
C-15 ***L*** must be independent of the target that is assessed	This requirement means that ***L*** must be useful for describing any type of security critical system. It should not have any concepts that depend on a specific system type or technology. Since the CORAS method can be applied to any security critical target, the modeling language should have the same level of flexibility.
C-16 Region	There must exist a concept similar to region. A region is a logical or physical part of the target that can be used as the link between the risk specific documentation and the target documentation. This is useful for structuring the documentation and helps the reader understand the risk models.

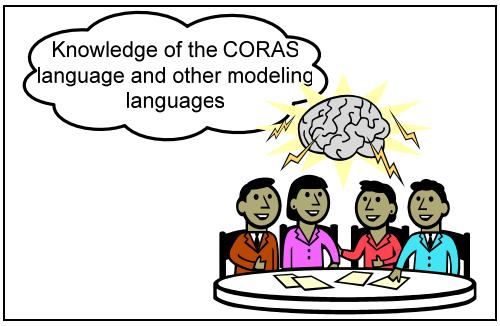
As already mentioned, SEQUAL distinguishes between concepts and symbols. To be a language with high domain appropriateness ***L*** must provide symbols for the set of the concepts identified in the discussion above:

Table 14 – Domain appropriateness: symbols

Requirements	Explanation
C-17 Asset	There must be a symbol for asset.
C-18 Vulnerability	There must be a symbol for vulnerability.
C-19 Aggregated vulnerability	There should be symbols for aggregated vulnerabilities
C-20 Risk	There must be a risk symbol that can show a risk value.
C-21 Risk theme	There should be symbols for risk themes (similar types of risks).
C-22 Aggregated risk	There should be symbols for aggregated risk.
C-23 Stakeholder	There must be a stakeholder symbol.
C-24 Threat	There must be a general threat symbol.
C-25 Human threat	***L*** should provide symbols for human- and non-human threat. Any other type of threat can be specified within one of these categories and extra symbols can be added if desired.
C-26 Non-human threat	
C-27 Threat scenario	There must be a symbol for threat scenario. The symbol should have the option to illustrate its estimated likelihood.
C-28 Unwanted incident	There must be a symbol for unwanted incident.
C-29 Treatment	There must be a symbol for treatment.
C-30 Treatment type	It should be possible to specify the type of treatment (e.g. preventive or repairing)
C-31 Likelihood, frequency, probability	There should be a symbol for likelihood/frequency/probability that could be used if particular attention to this concept is required.
C-32 Consequence	There should be a symbol for consequence that could be used if particular attention to this concept is required.

C-33 Association type	We must have both directed (arrow) and undirected associations (line only). A concept may initiate another, a concept may affect another, or one concept has some relation to another without specifying direction. It should also be possible to assign these relations with a description. We need: a) a undirected association between two symbols (a line with no arrow ends): b) a directed association pointing from X to Y, with the option to annotate it with a description (e.g. "initiates", "affects", "reduces frequency" etc.):
C-34 Operators	a) There must be an and-symbol for fault tree modeling where two or more relations are joined together and initiate a new event(s) in the meaning "if both A and B, then C". b) There must be an or-symbol for fault tree modeling where two or more relations are joined and initiate to a new event(s) in the meaning "if either A or B, then C".
C-35 Region	There should be a symbol for logical or physical regions that can be used to specify target.

6.3.2 Participant Language Knowledge Appropriateness



Participant language knowledge appropriateness is related to ***L*** and ***Ks***. ***Ks*** is the participant's knowledge about ***D*** and ***L*** (including all other modeling languages). ***M*** (the external representation) is made on the basis of ***Ks***. In this setting "participants" means those who are involved in modeling, but *without* doing the actual modeling (for example expert participants in the brainstorming). In order for ***L*** to be appropriate for the participant's language knowledge, ***L***'s internal representation should not conflict with the participants understanding of ***D***, and ***M*** should relate to ***D*** in an intuitive manner (this is also relevant for comprehensibility appropriateness).

With respect to participant knowledge appropriateness we have identified the following requirements:

Table 15 – Participant language knowledge appropriateness

Requirements	Explanation
S-1 ***L***'s external representation must imitate the real world	If the interpretation of ***M*** corresponds to the participants' understanding of the real world (***Ks)*** there will be less confusion and misunderstandings. An example is: "an unwanted incident affects three assets simultaneously" this can be modeled several ways but to avoid conflicts with the modelers understanding of the real world it should clearly illustrate the "simultaneously" aspect.
S-2 The symbols used in ***L*** must be based on most common interpretation of concept-symbol	This means that the symbols used in ***L*** represent ***D*** better or are more intuitive than other symbols that could have been used.
C-36 ***L*** must be understandable for people unfamiliar with modeling and without specific training	***M*** made with ***L*** should be easy to understand (read), even for people without modeling experience. To find how much effort is needed to learn ***L***, one should base oneself upon experiences with similar modeling languages like UML use cases, activity diagrams or flow charts.

6.3.3 Knowledge Externalizability Appropriateness

This appropriateness factor focuses on how ***Km*** (relevant modeler knowledge) may be articulated in ***L*** (the modeling language). Is it possible for the modeler to express his or her knowledge about for instance the target threats with ***L***? To achieve high score on knowledge externalizability appropriateness the modeler (the one who creates the actual model ***M***) should be able to use ***Km*** to learn ***L*** faster, be able to express all ***Km*** with ***L***, and design better ***M***.

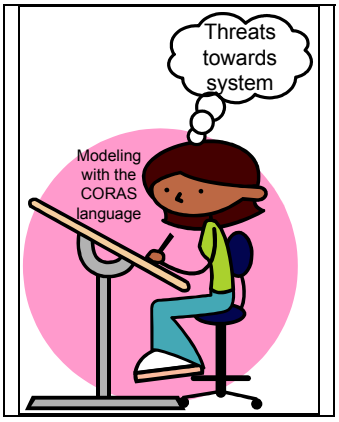


Table 16 – Knowledge externalizability appropriateness

Requirements	Explanation
S-3 ***L*** must help externalize tacit knowledge	This means that ***L*** should use well-known metaphors/analogies to explain/model more complicated relations to lower the effort needed to understand the models
S-4 It must be easy to model as part of actual work	The modeling should not require extensive training and heavy tool-support. The models should be easy and quick to create as part of the security analysis, and update during maintenance of the system. This means that modeling should not only be done before (planning) or after (post-hoc rationalization), but support interactive modeling.
C-37 It must be possible to model fault trees with ***L***	Fault tree analysis (FTA) (Sect. 3.4.4) is a well known risk analysis technique and it must be possible to draw fault tree diagrams using ***L.*** A conventional fault tree diagram has a restricted expressiveness which means that the notation is a subset of ***L***.
C-38 It should be possible to model event trees with ***L***	Event tree analysis (ETA) (Sect. 3.4.5) is a well known risk analysis technique and it should be possible to draw event tree diagrams using L (see **Figure 9 – Event** tree example).

6.3.4 Comprehensibility Appropriateness

Comprehensibility appropriateness relates ***L*** (the modeling language) and ***I*** (the social actor's interpretation) and focuses on how easy it is to interpret ***M*** (models) made with ***L***. There are requirements both to the internal and external representation of ***L***.



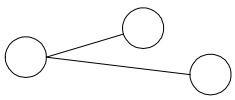
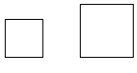
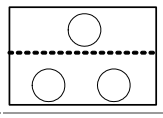
Table 17 – Comprehensibility appropriateness: internal representation

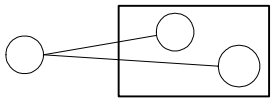
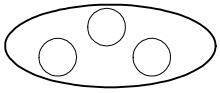
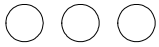
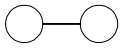
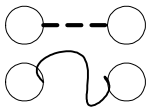
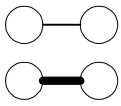
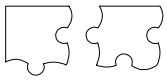
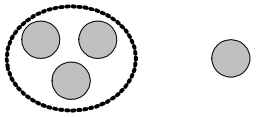
Requirements	Explanation
C-39 ***L*** must build on most-common ***I*** of risk specific concepts	If the interpretation of the terms corresponds to the participants' intuitive understanding of security analysis specific terms, there will be less confusion and misunderstandings.

S-5	The concepts in ***L*** must be general rather than specialized	The more delimited *D* is the more specialized concepts can be used, but the use of specialization should correspond to the specialization level in the security analysis, i.e. if the analysis has a high-level scope, *M* should also be high-level. The security analysis domain is in itself specialized, but within this domain we want *L* to be general (i.e. independent of target type, analysis scope and technology).
S-6	The concepts in ***L*** must be composable	*L* should make it possible to group related statements in a natural way. This means that we use compose in the sense "aggregate", meaning a form for grouping, e.g. vulnerabilities may be aggregated into a group of vulnerabilities etc.
S-7	***L*** must be flexible in precision	*L* must be able to express both precise knowledge and more vague information, meaning it must include both precise and vague constructs. Initially in a security analysis one often deals with incomplete or incorrect information about threats or assets, but they still must be modeled to provide an overview (even though they may be changed as more information is gathered). In some cases the natural language description of the modeling element will decide the model's precision level.
S-8	If the number of concepts has to be large, the phenomena must be organized hierarchically	In *D* there exists many types of threats, vulnerabilities, assets and so on, but it should be possible to see these concepts as a hierarchy where the top concepts are more general than the ones on the lower levels.
S-9	The concepts of ***L*** must be easily distinguishable from each other.	By using easily distinguishable concepts the models will be easier to understand.
S-10	The use of concepts must be uniform throughout the whole set of statements that can be expressed within ***L***	Meaning that a concept like 'risk' is to mean the same thing every time it is used
S-11	***L*** must be flexible in the level of detail	This is especially relevant early in the security analysis, for example it must be possible to specify some threat scenarios very detailed, while others will be more high level. In architectural descriptions one has different viewpoints of the architecture that shows specific parts or aspects of the architecture. The same idea can be used in security risk modeling, some diagrams may show the overall risk picture, some are dedicated to show details about threats etc. Modeling tools often have functions for showing/hiding diagram details, and the idea can also be used even without computerized tool. E.g. *L* should be able to provide an overall risk picture for a system, describing risks and threats that can harm the various assets in the system without details about "how" (threat scenarios + unwanted incidents)

Table 18 – Comprehension appropriateness: external representation

Requirements	Explanation
C-40 It should be possible to associate the overall threat picture to target descriptions	It must be possible to specify a general target with boundaries, its threats, assets and vulnerabilities in an abstract manner to facilitate integration with target documentation. This will typically be useful in the preliminary security analysis where it can help guiding the scope of the analysis. The target specification should initially contain as little as possible technology related aspects (only a logical or physical region can sometimes be sufficient), but be extended with more details as the analysis progresses. If the threat diagrams can be mapped to target descriptions they will increase their value and not be a "stand-alone" documentation type.
S-12 ***L*** must contain constructs that can represent the intention of the underlying conceptual model	This means that there should be possible to externally represent the concepts and relations in the underlying conceptual model. This requirement is covered by the requirements in the domain appropriateness section.
S-13 Symbol discrimination in ***L*** must be easy.	To avoid confusion, misunderstandings, irritation and frustration the symbols must be easy to distinguish.
S-14 It must be easy to distinguish which of the symbols in ***L*** any graphical mark in ***M*** is part of	These are means to achieve what Goodman [49] terms syntactic disjointness: • The use of symbols should be uniform, i.e. a symbol should not represent one concept in one context and another one in a different context. Neither should different symbols be used for the same concept in different contexts. • One should strive for symbolic simplicity. • One should use a uniform writing system for concepts at a comparable level. All symbols (at least within each sub-language) should be within the same writing system (e.g. non-phonological such as pictographic, ideographic, logographic, or phonological such as alphabetic). Obviously a modeling language contains both graphics and text, but then the text is labels to concepts, not 1.order concepts. • If using colors to mark semantics, one should not use more than 5-6 different colors in a given view [142]. The color of the label-text will depend on the color of the symbol. One also should have in mind how a model with colored symbols will look when printed (i.e. if the semantic differentiation meant to be carried by the coloring is retained)

S-15 The use of emphasis in ***L*** must be in accordance with the relative importance of the statements in the given ***M***	• Size (the big is more easily noticed than the small) • Solidity (e.g. bold letters vs. ordinary letters, full lines vs. dotted lines, thick lines vs. thin lines, filled boxes vs. non-filled boxes) • Difference from ordinary pattern (e.g. slanted letters, a rare symbol will attract attention among a large number of ordinary ones) • Foreground/background differences (if the background is white, things will be easier noticed the darker they are) • Color (red attracts the eye more than other colors). • Change (blinking or moving symbols attract attention) • Pictures vs. text (pictures usually having a much higher perceptibility, information conveyed in pictures will be emphasized at the cost of information conveyed textually) • Position (Westerners tend to read it from left to right) • Connectivity (objects able to connect to many others (having a high degree) will attract attention compared to objects making few connections)
S-16 Composition of symbols should be made in an aesthetically pleasing way	These are means for achieving an aesthetically pleasing ***M*** (according to [154]): • Angles between edges should not be too small • Minimize the area occupied by the drawing • Balance the diagram with respect to the axis • Minimize the number of bends along edges • Maximize the number of faces drawn as convex polygons • Minimize the number of crossings between edges • Place nodes with high degree in the centre of the drawing • Minimize differences among nodes' dimensions • Minimize the global length of edges • Minimize the length of the longest edge • Have symmetry of sons in hierarchies • Have uniform density of nodes in the drawing • Have verticality of hierarchical structures
S-17 ***L*** should not have empty symbols.	Symbols that are not related to a specific concept should not be used.
S-18 ***M*** must adhere to the most common principles from gestalt psychology	This does not mean that ***L*** should incorporate all the principles listed below, but rather avoid direct violation of them. Within the area of gestalt psychology, a number of principles for how to convey meaning through perceptual means is provided [160]:
	• A closed contour in a node-link diagram generally represents a concept of some kind. 
	• The shape of a closed contour is frequently used to represent a concept type. 
	• The color of an enclosed region represent a concept type 
	• The size of an enclosed region can be used to represent the magnitude of a concept. 
	• Lines that partition a region within a closed contour can delineate subparts of a concept 

	• Closed-contour regions may be aggregated by overlapping them. The result is readily seen as a composite concept	
	• A number of closed-contour regions within a larger closed contour can represent conceptual containment	
	• Placing closed contours spatially in an ordered sequence can represent conceptual ordering of some kind	
	• A linking line between concepts represents some kind of relationship between them	
	• A lined linking closed contours can have different colors, or other graphical qualities such as waviness, and this effectively represents an attribute or type of relationship	
	• The thickness of a connecting line can be used to represent the magnitude of a relationship (a scalar attribute)	
	• A contour can be shaped with tabs and sockets that can indicate which components have particular relationships	
	• Proximity of components can represent groups	
S-19 The most common modeling tasks should be as efficient as possible	Meaning that the most common modeling tasks should take less effort to model than more unusual tasks. "Common" in a CORAS setting must be interpreted as the most high-level modeling tasks, or the minimal modeling effort required by the method. The modeling test case in 6.1 provides an example of the minimum amount of information that needs to be modeled using the CORAS method.	
C-41 There must be a reasonable number of diagram types	***L*** must not include numerous different diagram types, ideally it should concentrate on one main diagram type with the possibility of specify details of this diagram in separate diagrams.	
C-42 ***L*** must have a precise semantics	To ensure that the language is understood in the same manner by all readers, *L* should have a precise textual semantics.	

6.3.5 Technical Actor Interpretation Appropriateness

Technical actor interpretation appropriateness relates to how well ***L*** can be used in computerized tools. With respect to this, we have identified the following requirements:

Table 19 – Technical actor interpretation appropriateness

Requirements	Explanation
S-20 ***L*** must have a formal syntax.	Formal syntax defines what a modeler is permitted and prohibited from drawing. Having a formal syntax insures that ***M*** made by different modelers have consistent notation.

S-21 ***L*** should have a formal semantics.	A formal semantics defines what the elements in the syntax means, described in a mathematical way. Having a formal semantics will make ***T*** consistent in different modeling tools, making automatic reasoning like consistency checks, translation of models to tables (and vice versa) and more.

6.3.6 Organizational Appropriateness

According to SEQUAL, this section should be based upon requirements from a specific organization that evaluates a language. In this report we do not have one particular organization in mind, but we present some general requirements drawn from our experience with industrial risk analyses. With respect to organizational appropriateness we have identified the following requirements:

Table 20 – Organizational appropriateness

Requirements	Explanation
S-22 ***L*** must contribute to reach ***G***	*M* must contribute to *G,* meaning it must reduce the effort needed to identify and understand the overall risk picture for the system assessed in a security analysis, and thereby contribute to increased control of the risks for the organization
C-43 ***M*** must ease the explanation of risks	*M* must ease the explanation of security risks related to the system assessed
C-44 ***L*** must be usable without investing in expensive software	There should be a version of *L* that can be used in general drawing or modeling tool (i.e. a version providing the symbols only, a plug-in or *L* must come with its own, free modeling environment). The simple version must not require the organization to purchase a new and different modeling tool if it already has one.

7 Evaluation of the UML Profile

This section summarizes the results evaluating the UML profile using the quality framework in *Chapter 6*. We first summarizes the findings regarding modeling the core security risk scenarios, and then the results from evaluating against the SEQUAL-based quality requirements. For the full version of the evaluation we refer to the technical report [57].

7.1 Modeling the Core Security Risk Scenarios (summary)

In order to evaluate a security risk modeling language we needed a benchmarking case representing core security risk scenarios to test the notation against. Through experience from several major security analyses in the SECURIS project we have gathered typical security analysis information into a complete set of scenarios covering all the phases in a security analysis. In the following sections we summarizes whether the UML profile was able to model the information.

Phase 1 - Context establishment:
It was unclear how to model the he distinction between direct and indirect assets within the asset diagram. This means that one cannot specify how damage to one asset may cause damage to other assets. However, the UML profile provides an option to model relationships between concepts like assets by using other UML notations, e.g. class diagrams. The standard does not provide any examples of this, but it means that one can create a hierarchy of assets and show the relationships between them. The UML profile does not let you model the risk acceptance level set by each client for each asset.

Phase 2 - Risk identification
It was no obvious way of modeling the indirect assets in threat and unwanted incident diagrams. If modeled as ordinary direct assets, we would loose the extra information about their status as indirect. If left out of the diagrams they have to be remembered or dealt with in some other way. Modeling this fairly small example of security analysis information resulted in five diagrams that are partly overlapping. To follow the path from the threat "Hacker" to the asset "CM2-Company reputation", it is necessary to look at as many as three diagrams.

Phase 3 - Risk estimation & Phase 4 - Risk evaluation
All the information about the risks and their consequence and likelihood estimates was modeled using the UML profile's risk diagrams. The thirteen risks resulted in as many as 7 figures. Also in these diagrams, information that already has been modeled is repeated (vulnerabilities). To find out which threats that may cause a risk one has trace the path backwards from unwanted incident via each of its threat scenarios to find the initiating threats. This makes it difficult to get a complete overview of the risk picture. The UML profile also lacks the option for differentiating between acceptable and non-acceptable risks.

Phase 5 - Treatment identification
On the positive side, the UML profile was able to express all the information in this phase, but on the negative side it was necessary to use 14 figures (approx. 9 pages) for the

information that is presented on 1,5 A4 page in *Chapter 6*. If we were to complete the diagrams (i.e. include R2CA) it would have required another five figures.

7.2 Evaluation of the Language Quality (summary)

The table below summarizes the average scores for each appropriateness factor discussed in *Chapter 6*. The scores provides an *indication* of weak and strong aspects of the UML profile, but since the level of details in the requirements can vary and no weighting has been included to adjust the score for this, we only use this as an indication of how well the language meets our requirements.

Table 21 – Evaluation scores

Category	Must-requirements			Should-requirements		
	Score	# req.	Avg.	Score	# req.	Avg.:
Domain appropriateness:						
• modeling perspectives	5	2	2,5		-	-
• concepts	34,7	14	2,5	4	-	-
• symbols	21,5	10	2		3*	1,3
Participant language knowledge appropriateness	5	3	1,7		-	-
Knowledge externalizability appropriateness	4	3	1,3	0	1	0
Comprehensibility appropriateness:						
• internal representation	18	7*	2,6		-	-
• external representation	8	6*	1,3	3	2	1,5
Technical actor interpretation appropriateness	3	1	3	0	1	0
Organizational appropriateness	5	3	1,7		-	-

* one or more requirements were omitted, see Sect. 6.7.1 in [57]

As we can se from Table 21, the UML profile has both weak and strong aspects.

The knowledge externalizability appropriateness receives a low score due to the illogical representation of the threats' paths via threat scenarios to the unwanted incidents where they cause harm to assets. The natural way of describing this is a logical sequence of events starting with the threat that initiates a threat scenario which leads to an unwanted incident that harms an asset. This idea is part of the language, but unfortunately the external representation fails to show its logical order. The UML profile makes use of the UML construct "include" which intend to show how a use case is included in another use case, but at first glance they look like they are on the same "level". In the unwanted incident diagram (Figure 20) in Sect.3.5.1, the threat scenario is included in the incident scenario "Denial-of-service", but for an untrained reader they seem like two equal entities, a clear violation of common gestalt principles. The language also fails to show which vulnerabilities a threat may exploit since these are only modeled as properties of the asset. The fact that more than one threat potentially can harm the same asset using different vulnerabilities is not possible to model.

Often a risk picture includes several dependencies, i.e. events that simultaneously may lead to an unwanted incident. This is traditionally modeled using fault trees, but the lack of AND/OR-operators in the UML profile makes fault tree modeling impossible.

Comprehensibility appropriateness of the external representation receives a low score, mainly related to the lack of adherence to common gestalt principles. The use of colors and size are not well considered and the include arrow in threat diagrams can easily be misunderstood. When it comes to judging the modeling effort required to make diagrams

as a natural part of the analysis process there are differences between the various diagram types. While threat and unwanted incident diagrams are fairly straightforward and does not take much effort, the rest of the diagrams are unnecessarily time- and space consuming from a usability perspective. To ensure a common understanding of the diagrams, a precise mapping from diagrams to text should be used. This mapping is not defined for the UML profile.

The rather low score for *participant language knowledge appropriateness* is caused by the symbols (icons) used. The UML profile symbols we have evaluated are based on standard UML use case symbols, but they are neither uniform, intuitive nor well designed and give an amateurish impression. Since the symbols are not finalized and just sketches of possible representations, the low score for this factor should not be emphasized as much as the other appropriateness factors of the language.

The UML profiles *organizational appropriateness* could have been better. It will probably provide a consistent and complete way of documenting the analysis information, but used as defined in [124] would not help reducing the effort needed to understand the organization's overall risk picture. Creating and understanding the models requires thorough knowledge of the target and the modeling notation itself and is not suitable for bridging the gap between system modelers and more ordinary system users. The ideas behind the various diagrams are on the other hand good, but they need to become simpler and more manageable in terms of complexity.

7.3 Conclusion

As we summarize in Sect. 7.1 it was possible to model almost all the information in the core security risk scenarios with the UML profile. However, being able to express the core security risk scenarios is not sufficient. The models should also present the information in the core security scenarios in a better way than the textual description alone. The diagrams should be understandable and manageable in terms of complexity, and provide a good overview of the information. With respect to this, many of the diagram types are not suitable for presentation of security risk analysis information. They are often characterized by duplication of information, and information that is spread out over several diagrams which makes it difficult to get an overview. The diagrams tend to be extremely space consuming (particularly the treatment effect and risk evaluation diagrams), and require the modeler to repeatedly model almost identical diagrams.

Also the quality evaluation points at the language's external representation as its main weakness. Its internal representation and underlying concepts related to expressing the domain are appropriate for the security analysis setting. Also its technical actor appropriateness receives a high score. The language should be a tool that helps visualizing, explaining and documenting the security risk analysis and this highly depends on its external representation. Unfortunately, some of the weaknesses are related to the underlying UML notation itself. This makes it difficult to redesign the language without violating fundamental UML constructs. There seems to be an inevitable choice to make, either use a non-optimal language with a notation that conforms to UML, or develop a new language without the restrictions from UML. A new language will on the one side not have the support and strength from UML, but on the other side it can have its main focus on understandability and usability. The two languages could be seen as two different versions based on the same underlying basis and with different strengths and application areas.

8 The CORAS Security Risk Modeling Language and Guideline

This section presents the syntax and semantics of the CORAS language and the guideline that explains how to use the CORAS language during the security analysis.

8.1 The Syntax and Semantics

For the CORAS language, we have the following syntaxes and semantics: the conceptual foundation (abstract syntax), the graphical syntax, and the textual syntax with its precise semantics in the form of translation rules into paragraphs in English.

8.1.1 The Conceptual Foundation

The conceptual foundation of the CORAS language can be seen as an abstract syntax where the concepts in the language match this syntax. The development of this foundation is described in *Chapter 5*, but the model (Figure 49) and its definitions are briefly repeated here. The class diagram notation is read as follows: "a threat is associated with at least 1 asset and maximum infinite assets, while an asset may have from 0 to an infinite number of associated threats".

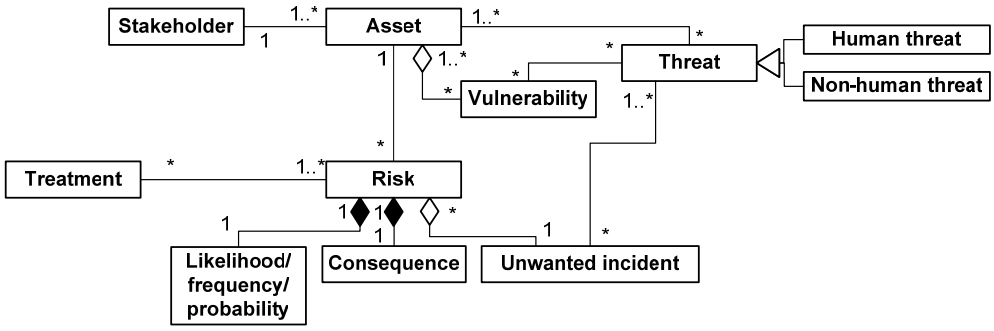


Figure 49 – The conceptual foundation

Figure 49 can be explained as follows: **Stakeholders** are those people and organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity regarding the target of analysis [6]. In practice we often use the term "client" for the organization that is paying for the actual analysis, while other stakeholders are called "other interested parties" for instance authorities, interest groups etc. An **asset** is something to which a stakeholder directly assigns value and, hence, for which the stakeholder requires protection [55]. Assets are subject to **vulnerabilities,** which are weaknesses which can be exploited by one or more threats [76]. A **threat** is a potential cause of an unwanted incident [76]. According to [6, 76], threats can be categorized as either human or non-human (environmental). Human threats can also be said to have accidental or deliberate origin. An **unwanted incident** is an event that may harm or reduce the value of assets and is something we want to prevent [55]. A **risk** is the chance of something happening that will have an impact upon objectives (assets) [6]. Our model captures this interpretation by defining a risk to consist of an unwanted incident, a likelihood measure and a consequence. The abstract concept "risk", the more concrete "unwanted incident", and their respective relationships to "asset" require some explanation. In our definition, an unwanted incident that harms more than one asset gives rise to one unique risk for each asset it harms. This enables us to keep the consequences for different stakeholders separate, since an asset is

always defined with respect to a single stakeholder. The level of risk is measured by a **risk value** (e.g. low, medium, high or other scales) which is based upon the estimated **likelihood** (a general description of frequency or probability), **frequency** or **probability** for the unwanted incident to happen and its **consequence** in terms of damage to an asset [6]. A **treatment** is the selection and implementation of appropriate options for dealing with risk [55, 76].

8.1.2 The Graphical Syntax

Figure 50 presents the syntactical representation of the conceptual foundation that is used in the CORAS diagrams, i.e. the graphical syntax. The stapled arrow is used for treatments, while the solid arrow is used between threats, vulnerabilities, threat scenarios, unwanted incidents and assets. The logical and/or gates are used when drawing fault trees [66] with the CORAS language.

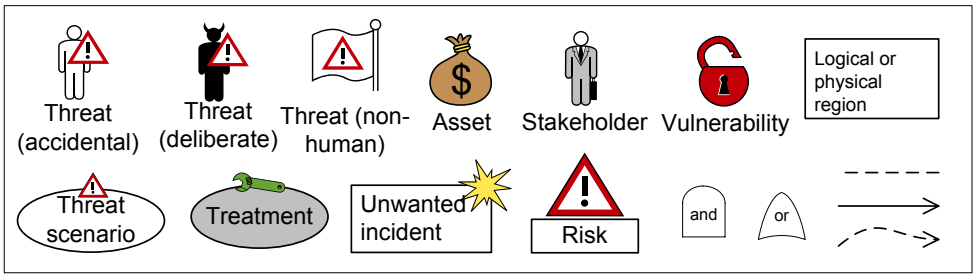


Figure 50 – The graphical syntax

The graphical syntax is further described in the example driven introduction in the modeling guideline in Sect. 8.2.

8.1.3 The Textual Syntax

The textual syntax of the CORAS language [36] (full details in *Appendix D*) has been defined for the elements in Figure 50, except for the AND/OR gates and the region. Ongoing research [21] is investigating the concept of region and this work may be included into the textual syntax in the future. Defining a textual syntax for the logical gates has so far been out of the scope of our work, but it may be a subject for future work.

To characterize the textual syntax for the CORAS language, we have used the ISO standardized Extended BNF notation [81]. The vertical bar _|_ represents options, braces {_} (respectively {_}$^{-}$) means an ordered sequence of zero (respectively one) or more repetitions of the enclosed element, and square brackets [_] denotes optional features. The terminal operators are surrounded by quotes: '_'.

To improve readability, we do not resolve all the EBNF definitions completely into terminal operators, but stop at the level of *identifier, linguistic term* and *numerical value*. An identifier is a name or a natural language description of an element, i.e. a finite sequence of upper- and lower-case letters, numbers, spaces and punctuation marks. Its precise definition in EBNF is:

$$identifier = \{upper\text{-}case\ letter | lower\text{-}case\ letter | digit | \text{`}\ \text{'} | punctuation\ mark\}^{-};$$
$$uppercase\ letter = \text{`A'}|\text{`B'}|\text{`C'}|\ldots$$
$$lowercase\ letter = \text{`a'}|\text{`b'}|\text{`c'}|\ldots$$
$$punctuation\ mark = \text{`.'}|\text{`,'}|\text{`:'}|\ldots$$
$$\ldots$$

Linguistic terms are identifiers that are members of a totally ordered set, for example likelihoods given as 'high', 'medium' and 'low'. Numerical values are just that: numbers allowing precise calculations of likelihoods, consequences and risk values. In the translation rules, we use ⇛ to denote the translation from graphical to textual syntax. The CORAS diagrams consist of two kinds of elements: vertices and relations between these. The elements may be annotated with additional information. The translation rules for the vertices and annotations of the diagrams are given by the naming conventions in Table 22.

Table 22 – Naming conventions

Vertex	Instance
party	p
asset	a
deliberate threat	dt
accidental threat	at
non-human threat	nht
threat scenario	ts
unwanted incident	ui
risk	r
treatment scenario	trs

Annotation	Instance
vulnerability	$v = \{v\} = V_1$
vulnerability set	$V_n = \{v_1, \ldots, v_n\}$
likelihood	l
consequence	c
risk value	rv
risk function	rf

When there is more than one instance of a type, we use subscripts, for example dt_1, dt_2 if there are two deliberate threats. The translation from the textual syntax into English is defined by a function [[_]] on textual expressions. We use _ := _ to denote "defined as".

Each diagram type has its own textual syntax described in *Appendix D*. In Figure 51 we provide an example of an asset diagram and its graphical syntax (note that stakeholder in *Appendix D* is named party, and therefore is denoted with a *p* in the diagram).

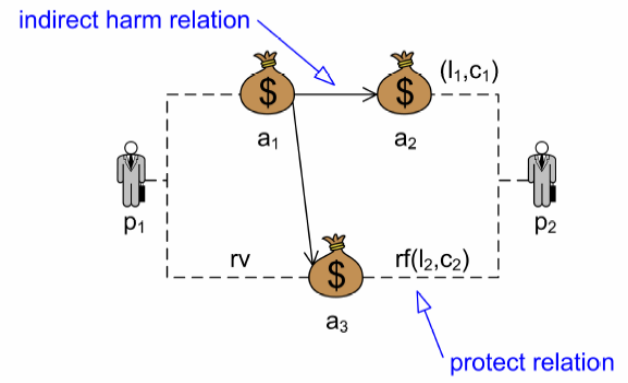


Figure 51 – The graphical syntax of asset diagrams

Below we describe the textual syntax of the asset diagram in Figure 51:

$$
\begin{aligned}
diagram &= (\{vertex\}^{-}, \{relation\});\\
vertex &= party \mid asset;\\
relation &= protect \mid indirect\ harm;\\
protect &= party \overset{[risk\ level]}{\cdots} asset;\\
indirect\ harm &= asset \rightarrow asset;\\
party &= identifier;\\
asset &= identifier;\\
risk\ level &= risk\ value \mid risk\ function(likelihood, consequence) \mid\\
&\quad (likelihood, consequence);\\
risk\ value &= linguistic\ term \mid numerical\ value;\\
risk\ function &= identifier;\\
likelihood &= linguistic\ term \mid numerical\ value;\\
consequence &= linguistic\ term \mid numerical\ value;
\end{aligned}
$$

8.1.4 The Structured Semantics

The translation of each diagram type from its graphical syntax to its textual syntax is described in full details in *Appendix D*. In this section we provide an example based on the asset diagram above. The translation is done schematically, vertex by vertex and relation by relation. The set of textual expressions we obtain is the translation of the complete diagram.

The translation of a vertex in the textual syntax is the label of the icon representing it in the graphical syntax. For example, the translation of the set of vertices in Figure 51 is the set:

$$\{p_1, p_2, a_1, a_2, a_3\}.$$

Note that in this translation, the type of each node is preserved through the naming conventions from Table 22. When the naming convention is not used we need to type each vertex, for example in a type table, or decompose the diagram into three sets: sets of parties, sets of assets and sets of relations.

There are two kinds of relations in the asset diagram in Figure 51:

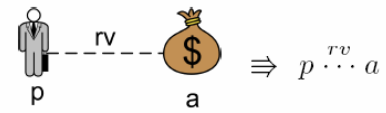


Figure 52 – Protect relation

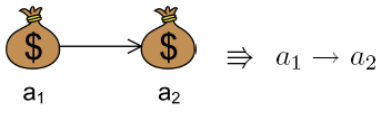


Figure 53 – Indirect harm relation

The translation of the protect relation similar when the relation is annotated with a risk function or a likelihood and consequence pair instead of a risk value. In either case, replace *rv* with the appropriate annotation. If the protect relation is not annotated, then remove *rv* altogether.

Translation from the Textual Syntax to English:
A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0$, $m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation:

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

The vertices:

$$[\![p]\!] := \boldsymbol{p} \text{ is a party.}$$
$$[\![a]\!] := \boldsymbol{a} \text{ is an asset.}$$

The protect relation:

$$[\![p \cdots a]\!] := \boldsymbol{p} \text{ wants to protect the value of } \boldsymbol{a}.$$
$$[\![p \overset{rl}{\cdots} a]\!] := \boldsymbol{p} \text{ wants to protect the value of } \boldsymbol{a}, \text{ but accepts } [\![rl]\!] \text{ or less.}$$

The second expression is the translation of a protect relation annotated with a maximum acceptable risk level. The *rl* for risk level is replaced by either *rv, rf(l, c)* or *(l, c)* depending on how the risk level is formulated.

The indirect harm relation:

$$[\![a_1 \rightarrow a_2]\!] := \boldsymbol{a_2} \text{ is harmed indirectly via } \boldsymbol{a_1}.$$

The annotations:

$$[\![rv]\!] := \text{risk value } \boldsymbol{rv}$$
$$[\![rf(l, c)]\!] := \text{risk function } \boldsymbol{rf} \text{ of } [\![(l, c)]\!]$$
$$[\![(l, c)]\!] := [\![l]\!] \text{ and } [\![c]\!]$$
$$[\![l]\!] := \text{likelihood } \boldsymbol{l}$$
$$[\![c]\!] := \text{consequence } \boldsymbol{c}$$

We illustrate the translation of asset diagrams with the following small example:

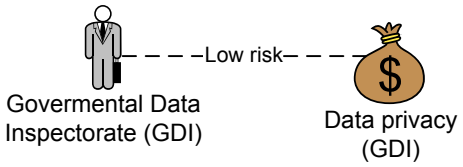


Figure 54 – Example of an asset diagram

A translation of the diagram above consists of two steps: first, we translate the vertices of the diagram:

- **Governmental Data Inspectorate (GDI)** is a party.
- **Data privacy (GDI)** is an asset.

Then we translate the relation using the vertices above:

- **Governmental Data Inspectorate (GDI)** wants to protect the value of **Data privacy (GDI)**, but accepts risk value **Low risk** or less.

8.2 The Guideline (an Example Driven Introduction)

This guideline follows the CORAS security analysis approach from the context establishment, via the risk identification, estimation and evaluation, and ending with the risk treatment. Within these phases there are a total of seven steps, summarized in Table 23:

Table 23 – Steps within the different phases in the CORAS approach

Phases	Steps
1. Context establishment:	**Step 1**: The first step involves an introductory meeting. The main item on the agenda for this meeting is to get the representatives of the client to present their overall goals of the analysis and the target they wish to have analyzed. Hence, during the initial step the analysts will gather information based on the client's presentations and discussions. **Step 2**: The second step also involves a separate meeting with representatives of the client. However, this time the analysts will present *their* understanding of what they learned at the first meeting and from studying documentation that has been made available to them by the client. The second step also involves a rough, high-level security analysis. During this analysis the first threats, vulnerabilities, threat scenarios and unwanted incidents are identified. They will be used to help directing and scoping the more detailed analysis still to come. **Step 3**: The third step involves a more refined description of the target to be analyzed, and also all assumptions and other preconditions being made. Step three is terminated once all this documentation has been approved by the client.
2. Risk identification:	**Step 4**: This step is organized as a workshop gathering people with expertise on the target of analysis. The goal is to identify as many potential unwanted incidents as possible, as well as threats, vulnerabilities and threat scenarios.
3. Risk estimation:	**Step 5**: The fifth step is also organized as a workshop. This time with the focus on estimating consequences and likelihood values for each of the identified unwanted incidents.
4. Risk evaluation:	**Step 6**: This step involves giving the client the first overall risk picture. This will typically trigger some adjustments and corrections.
5. Risk treatment:	**Step 7**: The last step is devoted to treatment identification, and one may address cost/benefit issues of the treatments. This step is also best organized as a workshop.

Throughout this section we give an example of how information from a security analysis may be modeled with the CORAS language, and guidelines for how and where in the process the diagrams are made. The guideline also provides instructions for modeling the target of analysis, and it suggests which roles that should be present at the different steps of the analysis. In the end of the guideline we present a selecting of useful recommendations, tips and hints from our experiences with CORAS in industrial field trials.

The target of analysis in our example is a web-based application that communicates confidential information between an insurance company and its customers. The development project is expensive and prestigious to the company, but the governmental data inspectorate is concerned whether the level of privacy is sufficient.

8.2.1 Context Establishment

The purpose of the context establishment is to characterize the target of the analysis and its environment. In our example the web application is represented as a logical region

(inspired by [141]) with two independent stakeholders: the company management, which is the "client", and the governmental data inspectorate which represents "other interested parties". The governmental data inspectorate has "Data privacy" as its only asset, while the company management identifies three additional assets: "Application availability", "Application interface usability" and "Company brand & reputation". "Company brand & reputation" is the only "indirect" asset which means that the asset can only be harmed if one of the direct assets is harmed first (Figure 55). The separation between indirect and direct assets implies that in risk identification should be at least one threat diagram for the direct assets, and one for the indirect assets.

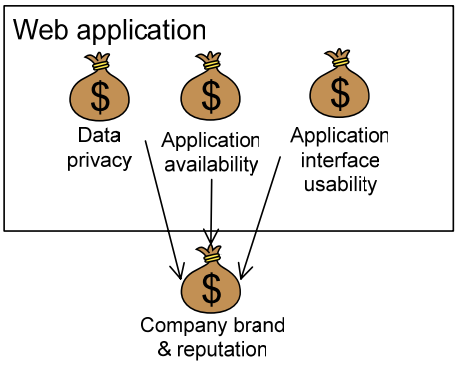


Figure 55 – Asset diagram: indirect and direct assets

The company management and the governmental data inspectorate value the assets differently. The company management wants to protect all assets, while the governmental data inspectorate is mainly interested in "Data privacy". The difference is described in the asset diagram in Figure 56, showing which assets each of the stakeholders are interested in and the risk value they are willingly to accept. If one has specified more detailed risk acceptance levels like maximum acceptable likelihood- and consequence values, they may be used instead of the risk values.

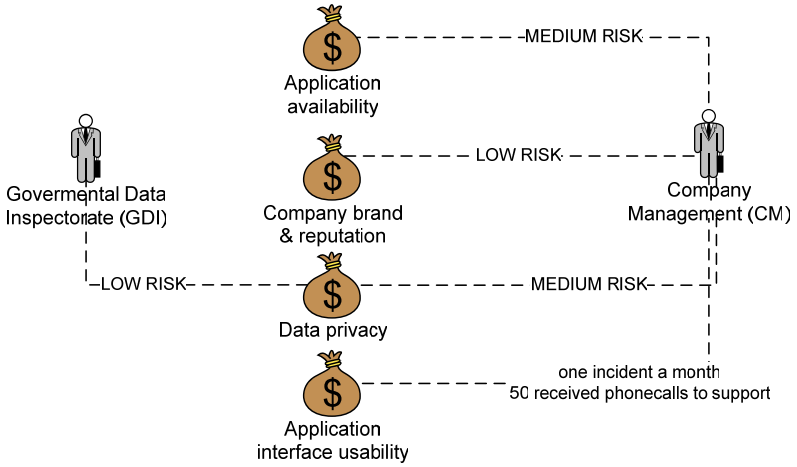


Figure 56 – Asset diagram: risk acceptance values

Since the two stakeholders value "Data privacy" differently, the asset is represented as two separate assets "GDI1. Data privacy" and "CM2. Data privacy" (Figure 57). The rest of the assets are also numbered to reflect their stakeholder: "CM1. Company brand & reputation", "CM3. Application availability" and "CM4. Application interface usability".

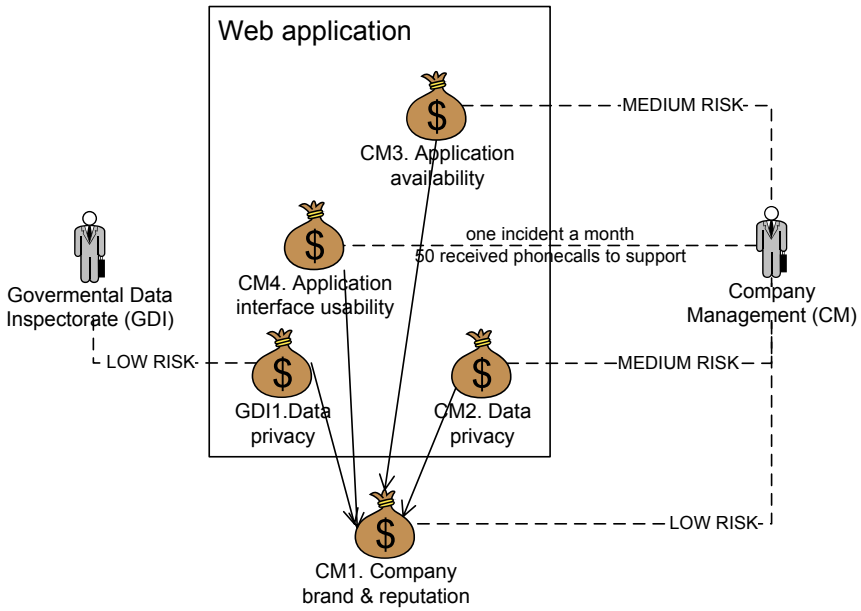


Figure 57 – Asset diagram: complete with all details

If the analysis has several assets and stakeholders with conflicting concerns, it may be a good idea to keep the asset related information in two separate diagrams. One diagram may show the assets (direct & indirect) and the relations between these, and the other diagram may specify the stakeholders' risk acceptance values for each asset (without relations between the assets). When combining these two views, one faces the risk of having a potential confusing and cluttered diagram.

Each phase in this guideline summarizes the instructions in a table like the one shown below. Since the context establishment includes three steps, there are also three sections in this table:

The introductory meeting step	
Analysis tasks: • The security analysis method is introduced. • The client presents the goals and the target of the analysis. • The focus and scope of the analysis is set. • The meetings and workshops are planned.	People that should participate: • Analysis leader (required) • Analysis secretary (required) • Representatives of the client: - Decision makers (required) - Technical expertise (optional) - Users (optional)
Modeling guideline: 1. At this early stage of the analysis it can be useful to describe the target with informal like drawings, pictures or sketches on a blackboard. 2. The presentation can later be supplemented with more formal modeling techniques such as UML or data flow-diagram.	
The high level analysis step	
Analysis tasks: • The target as understood by the analysts is presented. • The assets are identified. • A high-level analysis is conducted.	People that should participate: • Security analysis leader (required) • Security analysis secretary (required) • Representatives of the client: - Decision makers (required) - Technical expertise (required) - Users (optional)

<table>
<tr><td colspan="2">Modeling guideline:</td></tr>
<tr><td colspan="2">Asset diagrams:
1. Draw a region that logically or physically represents the target of analysis.
2. Place the assets within the region.
3. Indicate with arrows which assets may affect other assets.
4. Move assets, that are found to be indirect, outside the region. Indirect assets are only harmed as a consequence of a direct asset being harmed first.
5. Associate stakeholders (clients and other interested parties) with their assets.
6. Annotate each stakeholder-asset-relation with the maximum acceptable level of risk the stakeholder is willingly to accept.
Target descriptions:
1. Use a formal or standardized notation such as UML, but ensure that the notation is explained thoroughly so that the participants understand it.
2. Create models of both the static and the dynamic features of the target. Static may be hardware configurations, network design etc., while dynamic may be work processes, information flow etc.
3. For the static parts of the description, UML class diagrams and UML collaboration diagrams (or similar notations) are recommended.
4. For the dynamic parts we recommend UML activity diagrams and UML sequence diagrams (or similar notations)</td></tr>
<tr><td colspan="2">The approval step</td></tr>
<tr><td>Analysis tasks:
• The client approves target descriptions and asset descriptions.
• The assets may be ranked according to importance (optional).
• Consequence scales must be set for each asset within the scope of the analysis.
• A likelihood scale must be defined.
• The client must decide risk evaluation criteria for each asset within the scope of the analysis.</td><td>People that should participate:
• Security analysis leader (required)
• Security analysis secretary (required)
• Representatives of the client:
- Decision makers (required)
- Technical expertise (required)
- Users (optional)</td></tr>
<tr><td colspan="2">Modeling guidelines:
- Modeling is not part of this step.</td></tr>
</table>

8.2.2 Risk Identification

In the risk identification phase the security analysis leader and the brainstorming participants must find answers to questions like: *what are you most concerned about with respect to your assets* (modeled as threat scenarios and unwanted incidents), *who/what initiates these* (threats), and *what makes this possible* (vulnerabilities). This information is modeled in *CORAS threat diagrams*.

Consider the threat diagram in Figure 58 (and the extracted threat diagram for indirect assets in Figure 59). The diagram in Figure 58 focuses on network related threat scenarios towards the direct assets. The asset "CM4. Application interface usability" is not affected by these kinds of incidents and therefore left out. We have two threats: employees who may cause incidents by accident (human, accidental) and IT-infrastructure (non-human threat). Since this is a small example, we model both human and non-human threats in the same diagram, but often it can be useful to keep these in separate diagrams. The company management and the governmental data inspectorate are concerned about the following unwanted incidents: "disclosure of data", "corruption of data", and "unavailability of application".

We now explain one chain of events from the initiation caused by a threat to the left, to its impact on an asset to the right. The threat "IT-infrastructure" may first exploit the vulnerability "hardware failure" to make the server crash. Then it uses the vulnerability "poor backup solution" to initiate the threat scenario "application database fails to switch to backup solution". This threat scenario may again lead to the unwanted incident "unavailability of application", which impacts the availability of the application.

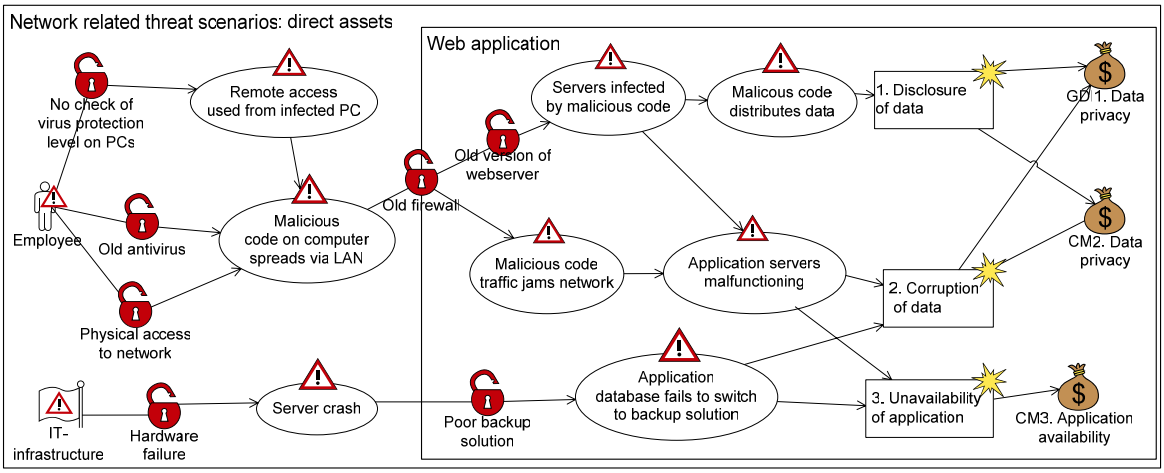


Figure 58 – Threat diagram: direct assets

To assess the indirect assets, the unwanted incidents that indirectly harm assets are specified separately (Figure 59). If they were to be modeled in the threat diagram above they would have been placed to the utmost right, but by extracting them into a separate diagram we can also include relevant incidents from other threat diagrams. Often indirect assets are of a more organizational or enterprise related kind, and therefore likely to be affected by several different types of incidents described multiple, different threat diagrams. To keep the unwanted incidents relation to their initiating threat, the threats are included in the diagram, while information about the vulnerabilities and threat scenarios are left out.

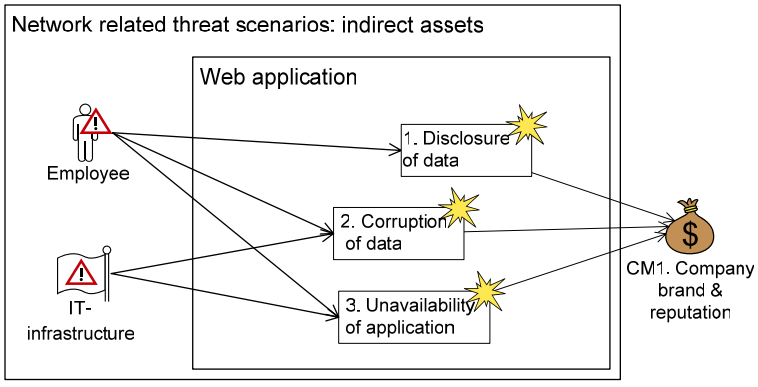


Figure 59 – Threat diagram: indirect assets

<table>
<tr><th colspan="2">The risk identification step</th></tr>
<tr><td>Analysis tasks:
The initial threat diagrams should be completed with identified threats, vulnerabilities, threat scenarios and unwanted incidents.</td><td>People that should participate:
• Security analysis leader (required)
• Security analysis secretary (required)
• Representatives of the client:
- Decision makers (optional*)
- Technical expertise (required)
- Users (required)
*because this workshop often has a technical focus and the decision makers' competence is more relevant in the next step.</td></tr>
<tr><td colspan="2">Modeling guideline:</td></tr>
<tr><td colspan="2">Threat diagrams:
1. Use the region from the asset diagram. You may add more regions that group elements to increase the readability of the diagram. See also Sect.8.2.6 for further information about regions.
2. Decide how to structure the threat diagrams. A diagram may either focus on one asset at the time, a particular aspect of the target, or different kinds of threats. For instance, deliberate sabotage in one diagram, mistakes in an other, environmental in a third etc. ([76] contains a useful classification). This makes it easier to generalize over the risks, for example "these risks all harm asset X", "these risks are caused by human errors" or "these risks are related to the network".
3. Threats are placed to the left in the region; threats that can be classified as external to the target (hackers, intruders, third party services etc.) are placed outside the region.
4. Assets are listed on the right.
5. Unwanted incidents are placed within the region with relations to the assets they impact.
6. Assets that are not harmed by any incidents can be removed from the diagram.
7. Add threat scenarios between the threats and the unwanted incidents in the same order as they occur in real time (i.e. in a logical sequence). See Sect. 8.2.6 for tips and hints.
8. Insert the vulnerabilities before the threat scenario or unwanted incident they lead to. E.g.: a vulnerability called "poor backup solution" is typically placed before the threat scenario "the backup solution fails to run the application database correctly".
9. If specifying indirect assets at the outmost right of the threat diagram makes it too complex, draw separate threat diagrams for the indirect assets that shows the threats, unwanted incidents and the indirect assets only.</td></tr>
</table>

8.2.3 Risk Estimation

The threat diagrams are input to the risk estimation where threat scenarios and unwanted incidents are assigned likelihood values and consequences. Likelihood values may be either qualitative (e.g. often, seldom etc.) or quantitative (e.g. 2 times per year, 50% probability etc.) depending on the type of analysis. If information on likelihood of threat scenarios and unwanted incidents is available, it may be added to the threat diagrams. For threat scenarios and unwanted incidents that are difficult to estimate, the analysis leader may give suggestions based on historical data like security incident statistics or personal experience. The likelihood of the threat scenarios can be used to extract a combined likelihood for unwanted incidents. In Figure 60 the final likelihood for "corruption of data" is based on the likelihood of the two threat scenarios "application servers malfunctioning" and "application database fails to switch to backup solution". Consequences are estimated for each "unwanted incident-asset" relation. The consequence value is taken from the asset's consequence scale that was defined during the context establishment.

There are different ways of computing the likelihood of an incident that may be caused by more than one threat scenario. If the estimates are suitable for mathematical calculations a computerized tool may be used. Since the likelihood scale in our case is in the form of intervals, the analysis leader decides to use an informal method that is quite straight forward and transparent, suitable for the brainstorming setting. For more precise calculation of probabilities fault tree analysis (FTA) [66] may be used. It is of course important that the combined estimates reflect reality, meaning that the combined estimates must be validated by the participants in the analysis.

Table 24 – Example of how one may combine likelihood estimates

Threat scenario	Likelihood	Unwanted incident	Combined likelihood
Application servers malfunctioning	1 occurrence per 5 years (1:5y)	Corruption of data	(1:5y)+(1:1y) = 6:5y which is so close to 1:1y that this value is used as a combined likelihood estimate.
Application database fails to switch to backup solution	1 occurrence per year (1:1y)		

We here use the consequence scale: large (3), medium (2) and small (1), but in a real security analysis more precise scales for each asset should be used.

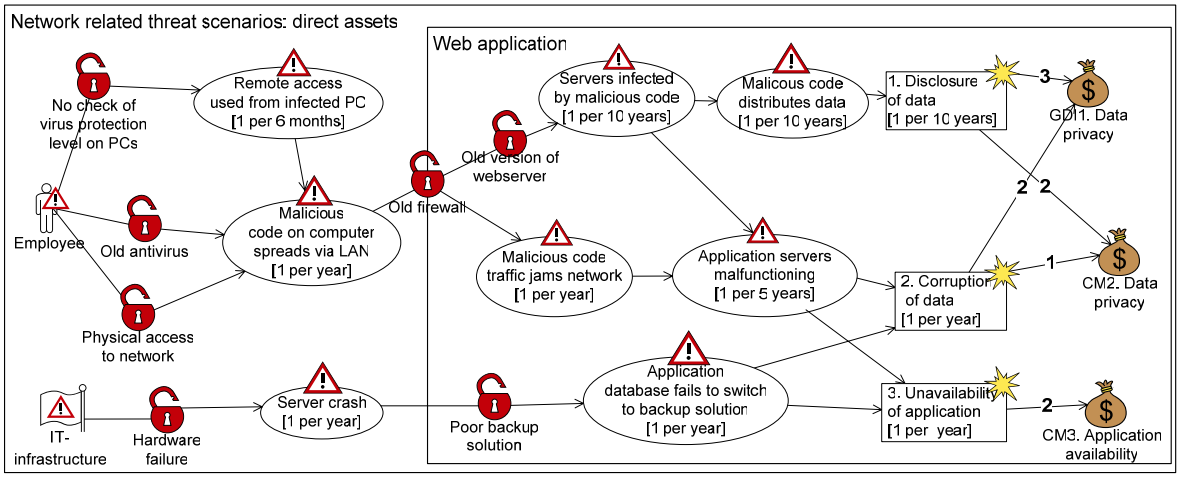


Figure 60 – Threat diagram (direct assets): likelihood and consequence estimation

In the threat diagram showing indirect assets (Figure 61), there are no threat scenarios that can be used as input for estimating the likelihood of the unwanted incidents. Instead one uses the estimates from the equivalent unwanted incidents in the direct asset threat diagrams and adjusts them either up or down according to the participants judgments. With respect to estimating consequences, we also use the 1-3 scale in the example, but could very well use a scale that specify in more details exact consequences for "CM1. Company brand & reputation". It is also permitted to split an incident into two incidents if the initiating threats give rise to potentially different consequences. In Figure 61 "Unavailability of application" is estimated to have a more serious consequence if it is caused by an employee, than if it is caused by IT-infrastructure.

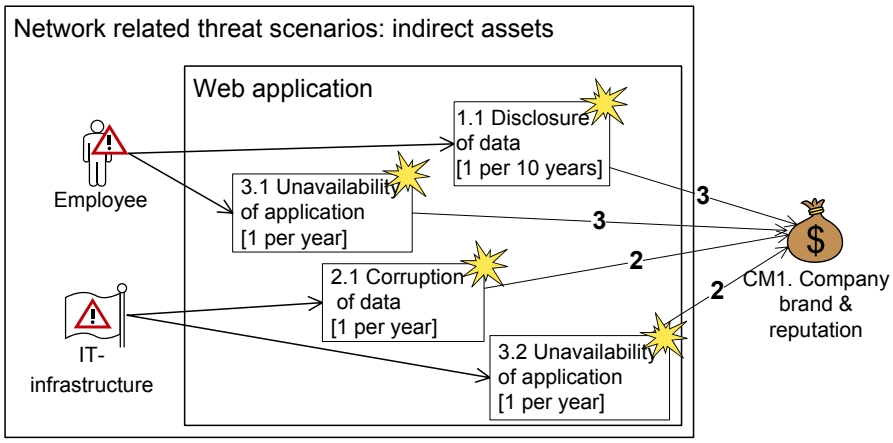


Figure 61 – Threat diagram (only for the indirect asset): likelihood and consequence estimation

The risk estimation step	
Analysis tasks: • Provide likelihood estimates for the unwanted incidents • Estimate consequences of the unwanted incidents. • Estimate risks on basis of the likelihood and consequence estimates.	People that should participate: • Security analysis leader (required) • Security analysis secretary (required) • Representatives of the client: - Decision makers (required) - Technical expertise regarding the target (required) - Users (required)
Modeling guideline: *Risk estimation on threat diagrams:* 1. Add likelihood estimates to the threat scenarios if this kind of information is available. 2. Add likelihood estimates to each unwanted incident, either directly, or based on the threat scenarios or unwanted incidents that lead up to it. 3. Annotate each unwanted incident-asset relation with a consequence taken from the respective asset's consequence scale. 4. If there are extracted threat diagrams for the indirect assets, do step 2 and 3 also for these. If the consequences are different according to which threat that gives rise to the unwanted incident, one may split the incident.	

8.2.4 Risk Evaluation

On the basis of the risk estimation we model the risks with their resulting risk values in risk diagrams, and optionally also in risk evaluation diagrams. Risk diagrams are used to give an overview of the magnitude of each risk, including the threats that are involved and the assets that are harmed. The risk evaluation diagram is similar, but shows whether the risks are acceptable or unacceptable according to the risk acceptance criteria defined during context establishment (Figure 57).

The risk diagram for direct assets (left diagram in Figure 62) illustrates that there is a low risk for disclosure of data caused by an employee, towards the asset "CM2. Data privacy", but a medium risk for the same incident towards "GDI1. Data privacy". For the indirect assets (right diagram in Figure 62) we see that the risk value of "Unavailability of application" is different according to which threat that is initiating it.

The preferred numbering convention is up to the modeler, we recommend that risks referring to the same unwanted incident start at the same number (e.g. "1.1 Disclosure of

data" and "1.2 Disclosure of data"). We also recommend that the risks in the diagram for indirect assets refer to their "parent incident" or origin, starting with the same number as its parent (e.g. "1.1.1 Disclosure of data" in Figure 62).

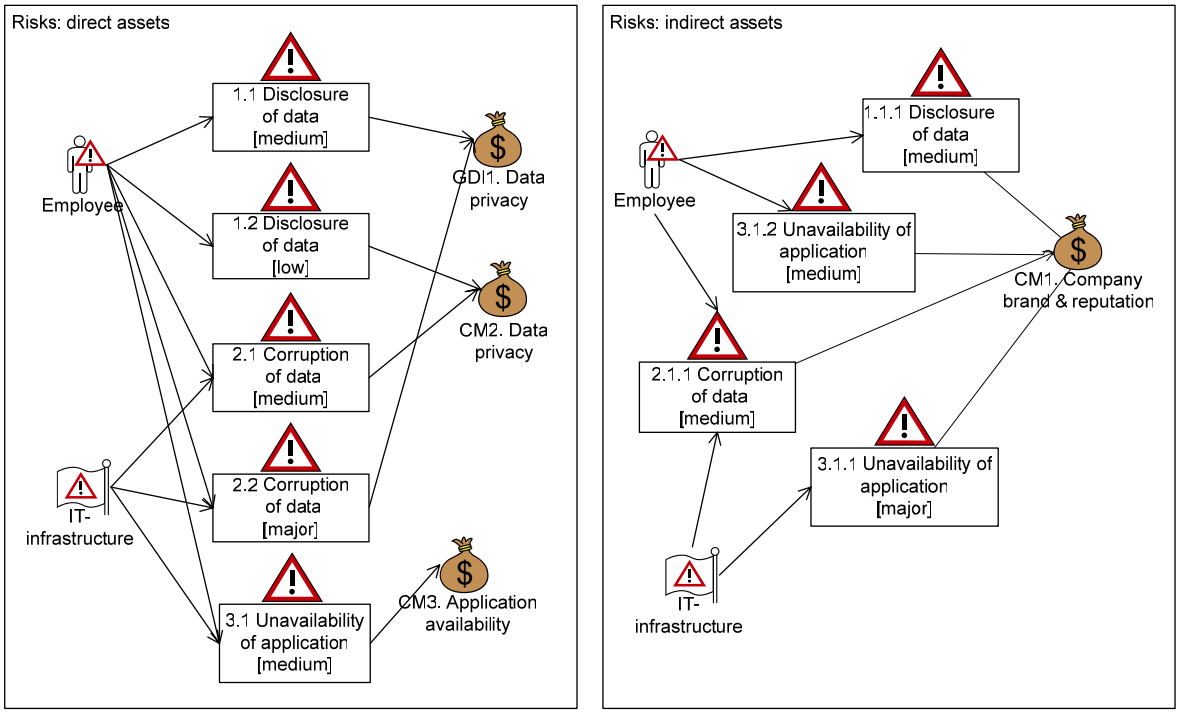


Figure 62 – Risk diagrams

To indicate those risks in the risk diagrams that are above risk acceptance criteria, one may draw a risk evaluation diagram. This is made by replacing the risk value with "acceptable" or "unacceptable" according to the risk acceptance criteria for each stakeholder and asset (Figure 63). We recommend that the difference between the two categories is also visualized using a color (in this case the acceptable risks are made gray). In a modeling tool this would typically be an automatic function.

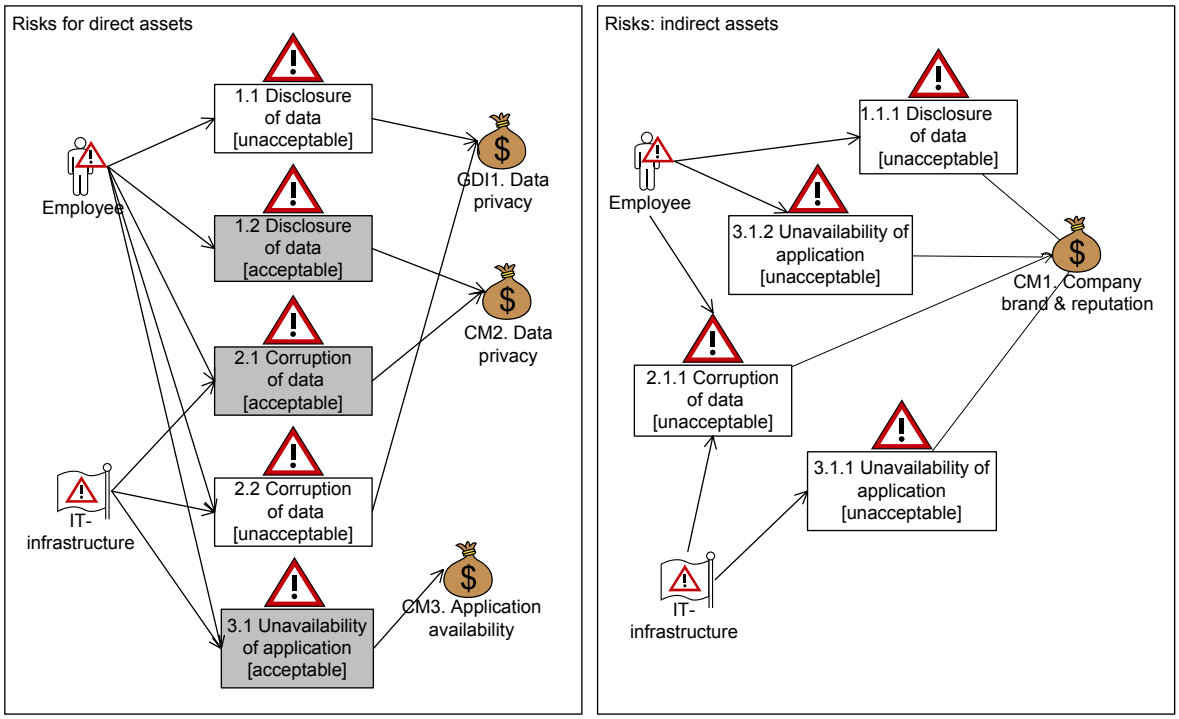


Figure 63 – Risk evaluation diagrams

The risk evaluation step	
Analysis tasks: • Likelihood and consequence estimates should be confirmed or adjusted. • The final adjustments of the acceptable risk levels should be made (if needed). • An overview of the risk may be given in a risk diagram.	People that should participate: • Security analysis leader (required) • Security analysis secretary (required) • Representatives of the client: - Decision makers (required) - Technical expertise regarding the target (required/optional*) - Users (required/optional*) *Depends on whether this step is included in risk estimation or not. If it is part of the risk estimation workshop, all representatives should be present; otherwise it may be sufficient with only the decision makers.
Modeling guideline:	
Risk diagrams: 1. Use the threat diagrams and replace all unwanted incidents with risk symbols, showing the short risk description and the risk value. 2. Remove threat scenarios and vulnerabilities, but keep the relations between the threats and the risks. 3. If useful, split the risk diagrams into several diagrams according to type of threat, part of the target or asset importance (for instance, show all risks related to network, all risks for specific assets etc.). 4. If there are threat diagrams for indirect assets, do step 1 for these. 5. You may indicate those risks that are acceptable using risk evaluation diagrams. Replace the risk value in the risk diagram with "acceptable" or "unacceptable" according to the risk acceptance criteria. We recommend that the difference between the two categories is highlighted using a color label.	

8.2.5 Treatment Identification

During treatment identification, all the risks that are unacceptable are evaluated in order to find means to reduce their likelihood and/or consequence. Since treatments can be costly, they may be assessed with respect to their cost/benefit, before a final treatment plan is made.

The initial treatment diagrams are similar to the final threat diagrams except that every relation between an unwanted incident and an asset representing an *unacceptable risk* is symbolized with a risk icon and an identifier. Threats, threat scenarios, vulnerabilities and unwanted incidents that constitute acceptable risks only, should not be part of the treatment diagram.

Figure 64 shows the proposed treatments for the asset "GDI1.Data privacy": "Upgrade server", "Limit access to network" and "Increase awareness of security risks". The treatments in our example all address vulnerabilities, but one may also specify treatments for threats, threat scenarios or unwanted incidents. In our experience, the participants find it most intuitive to focus on treating vulnerabilities, using the analogy of closing a padlock in the meaning of closing the specific path through the diagram,.

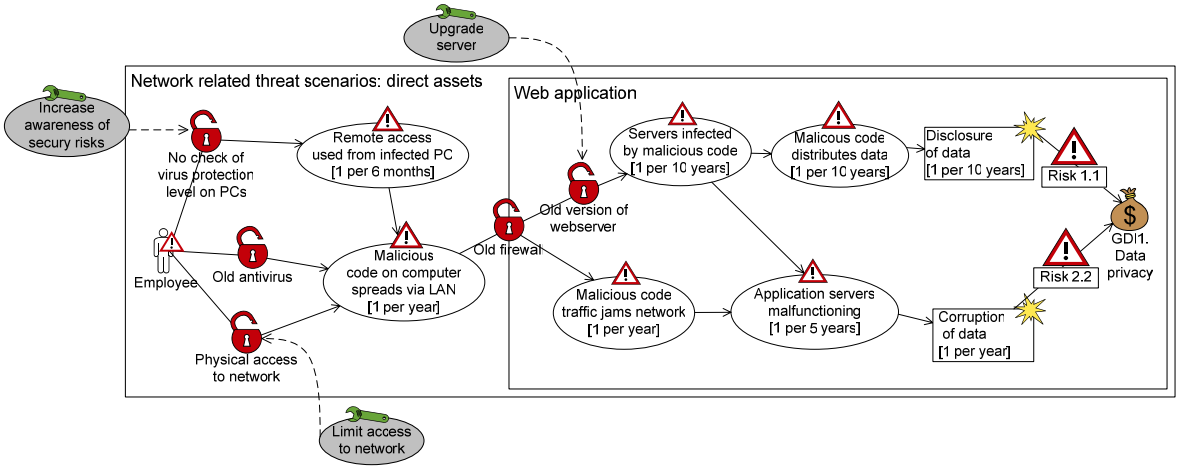


Figure 64 – Treatment diagram for the direct asset "GDI1.Data privacy"

Similarly, the treatments for the indirect asset "CM1.Company brand & reputation" includes the already mentioned treatments, in addition to "New backup solution" (Figure 65).

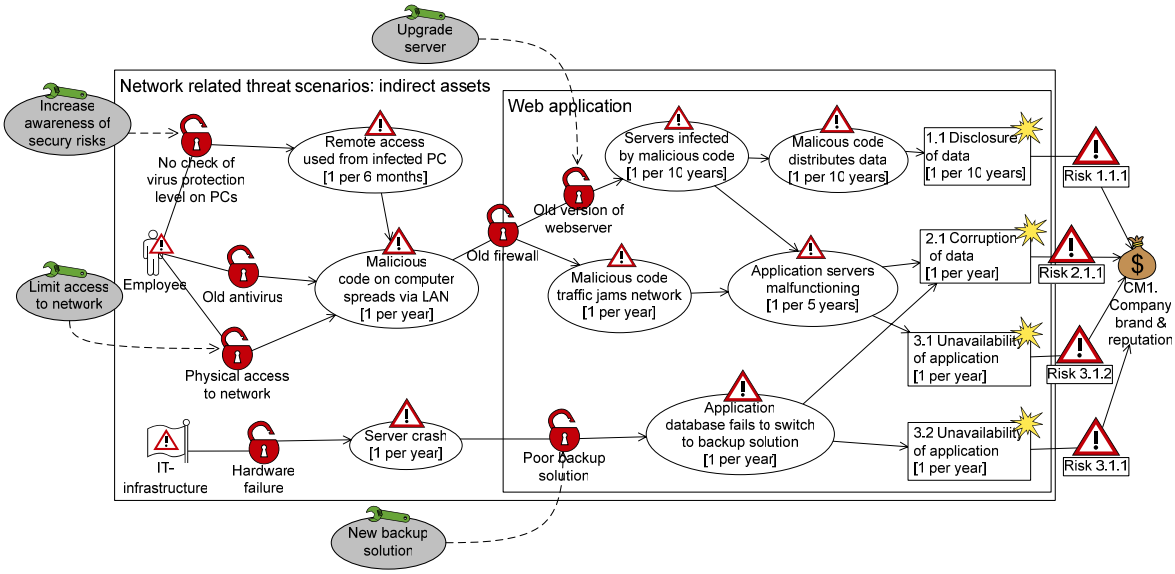


Figure 65 – Treatment diagram for the indirect asset "CM1.Company brand & reputation"

The treatments can be summarized in treatment overview diagrams shown in Figure 66 and Figure 67. To help avoiding cluttered diagrams we recommend gathering treatments that address the same risk in so called "junction points", and only use one arrow from the junction point to the particular risk.

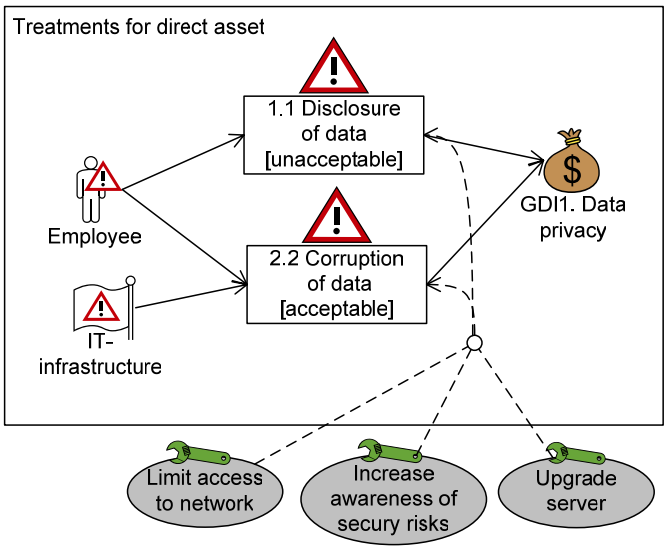


Figure 66 – Treatment overview: direct asset

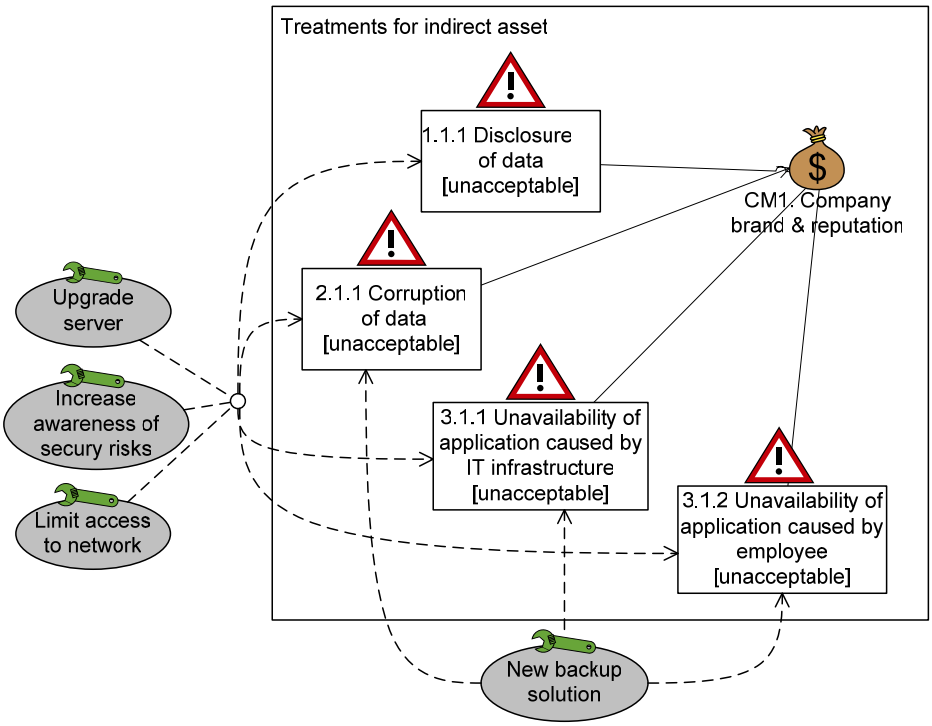


Figure 67 – Treatment overview: indirect asset

The risk treatment step	
Analysis tasks: • Add treatments to threat diagrams. • Estimate the cost/benefit of each treatment and • decide which ones to use. Show treatments in risk overview diagrams.	People that should participate: • Security analysis leader (required) • Security analysis secretary (required) • Representatives of the client: - Decision makers (required) - Technical expertise (required) - Users (required)
Modeling guidelines: *Treatment diagrams:* 1. Use the threat diagrams as a basis and annotate all arrows from unwanted incidents to assets representing *unacceptable* risks with risk icons. Relations and elements representing acceptable risks can be removed (this includes unwanted incidents, vulnerabilities, assets, threats and threats scenarios that do not constitute unacceptable risks). 2. Annotate the diagram with treatments, pointing to where they will be applied 3. If several treatments points towards the same risks (a many-to-many relation) we recommend using "junction points" to avoid multiple, crossing lines *Treatment overview diagrams:* 1. Use the risk diagrams as a basis; remove the acceptable risks (including threats and assets that are not associated with an unacceptable risk). 2. Add treatments as specified in the treatment diagram(s). Treatments directed towards vulnerabilities or threat scenarios should point towards the risks they indirectly treat. 3. If several treatments points towards the same risks (a many-to-many relation) we recommend using "junction points" to avoid multiple, crossing lines.	

8.2.6 Additional Explanations, Tips and Hints

The following explanations and recommendations are based on experiences from the industrial field trials in the SECURIS project. They are meant as additional help to the user.

Choosing the appropriate level of abstraction: this is often a difficult task for an inexperienced modeler. When identifying threat scenarios one should follow the strategy used in fault tree analysis. When modeling a fault tree, one starts with the top node (the unwanted incident) and works oneself down the tree by asking "what may cause this incident". When reaching an appropriate level of detail (i.e. manageable in complexity) the analysis stops. The same tactic should be used for threat diagrams. When adding an unwanted incident one should focus on identifying the threat scenarios that may cause the incident, thereby adding threat scenarios in a "backward" manner towards the threats. This way one avoids starting at a too detailed level by specifying extremely detailed initial threat scenarios, which may not contribute to the overall risk picture.

Regions and asset types: the regions are used as the link between threat diagrams and the target description. When designing the threat diagrams they should have a clear link to the target description, preferably visualized through the use of regions. The threat diagrams should be a more abstract view of the target descriptions, that includes risk related information but suppresses detailed technical information. The target descriptions function as underlying documentation that can be consulted in cases where the participants are uncertain or need to refresh their memory with respect to the target. If the target consists of logical or physical parts with different security levels this may be considered as different regions (e.g. a company's intranet and extranet). When the guideline recommends to place the direct assets within the region and the indirect outside, it really means that the modeler should separate between assets at different abstraction levels. An indirect asset is an asset that is not directly part of the target of analysis, but nevertheless may be indirectly harmed in cases where assets within the target are harmed. We will explain this with an example: the emergency preparedness instructions for a major, international company states that the director of communication should give a statement in a press release if information about security incidents leaks to the press. The communication director must appear trustworthy in order to regain public confidence in the company. In a security analysis of the company's information handling system, which also includes the emergency preparedness related to security incidents, the trustworthiness of the communication director has been defined as an asset. One day it is disclosed that the communication director has a secret Swiss bank account that has been withheld from the national tax authorities. Suddenly the trustworthiness of the communication director is questioned, and indirectly also the trustworthiness of the company itself. Indirect assets are therefore part of the CORAS method to illustrate that the consequences of a harmed asset may have rippling effects beyond the scope of the analysis. When deciding upon the direct assets, it is important to have in mind that these are strongly related to the target of analysis. They may represent functions, services, data or other aspects of the target that are particularly valuable to the company. During a security analysis an indirect asset may change status and become a direct asset and vice versa if findings suggest so or the scope is changed.

To ensure a good understanding of terminology and notation one should:

- Briefly repeat the meaning of central concepts in the beginning of each meeting to avoid confusion.
- Limit the amount of information in one diagram, and rather split it according to threat type, scenario type or asset type.
- Strive towards correctness from the very beginning. This means that the participants must be involved early, enabling them to adjust the diagram as they find appropriate.
- Provide a proper introduction to the notation and the diagrams that will be presented to the participants.

- Keep the assets within the scope of the analysis at the same abstraction level.
- Utilize checklists ensure that all relevant aspects are covered.
- Check whether assets, consequence scales and likelihood scales from previous analyses are relevant and can be reused.
- Document any changes to the diagrams during the brainstorming session including their rationale and origin since this represents important information.

How to decide the correct level of abstraction for the analysis, especially for the consequence scale? This is related to the whole security analysis process, and not specific to the modeling language used. It depends on the scope of the analysis and the experience of the analysis team. Or recommendation is to consult other analysts and previous analyses if available, or study the example given in the guideline to get an impression of how this has been solved in other analyses

How much time to spend on each step and how to prioritize between the activities? In general no analysis is the same, and the time spent depends on the target complexity, the competence of the participants and more. The guideline provides a suggestion for the number of meetings and workshops that is expected during a CORAS analysis. It is up to the analysis team to adjust this on a case to case basis. The analysts should consult examples or previous analyses to get an impression of how much information it is normal to handle during one meeting or workshop.

How to structure the brainstorming session so that most of the participants would be actively involved? If the participants cannot contribute to each others areas, split the brainstorming session into sub sessions with different focus. If participants feel they waste time by being present, they will become demoralized and not contribute to a fruitful discussion. Also in this case it may be a good idea to split up in smaller groups.

Is it necessary to follow all the steps in the guideline? If one chooses not to follow the guideline, it places more responsibility on the analysis team. In our opinion, the only situation where this may be an option is if the analysis team is highly competent in security analysis and thereby knows when to take short-cuts, but also recognize when certain aspects need more detailed analysis. However, for the non-experienced user we recommend following the guideline in detail.

9 Evaluation of the CORAS Language and the Guideline

This summarizes the results from the evaluations of the CORAS language and guideline. The evaluations are based on different approaches and have been conducted at different stages during design and development.

Some evaluations were carried out initially to explore, and capture design recommendations. They were:

- The quality evaluation of the UML profile using the quality evaluation framework described in *Chapter 6*. The UML profile was the starting point of our work. This evaluation has already been summarized in *Chapter 7*, and the full documentation is available in [57].
- The use of questionnaires on both students and professionals to evaluate the conceptual foundation on which our language has been built. See *Chapter 5* for a summary and *Appendices A* and *B* for the full documentation.
- The use of questionnaires on students to evaluate the use of risk related symbols vs. basic UML symbols. See *Chapter 5* for a summary and *Appendix A* for the full documentation.

Some evaluations were carried out during design and development. They were:

- The use of a questionnaire to evaluate and select graphical constructs. The results from this are summarized in Sect. 9.1. See *Appendix G* for the full documentation.

Some evaluations were carried out after, or when our approach was close to completion. They were:

- The use of experience reports from field trials to evaluate the language and guideline in an industrial context. See Sect. 9.2.
- The use of the quality evaluation framework described in *Chapter 6* to evaluate the CORAS language. See Sect. 9.3 below and *Appendix H* for the full details.

9.1 Evaluation Using Questionnaires

As explained in full details in *Appendix G*, questionnaires filled in by a group of professional system developers were used to select between different syntactic alternatives. We looked for ways of improving the readability and comprehensibility of threat- and risk diagrams by adding information in terms of text labels or simple graphical means. From this study, one conclusion was that text labels are preferred over visualization mechanisms like size, color or line type. For our language, this meant for example that that we decided to tag a major risk with the text label "major", rather than representing it with a large risk icon. In the following, we summarize the most important decisions made, based on this study.

To assess threats in a security analysis, it is useful to describe the paths via which they are most likely to harm the assets. By highlighting (attract attention to) or "sublightning" (remove attention from) paths in the threat diagram, one may visualize the most or least likely paths. None of the alternatives we investigated for these purposes were preferred by

the subjects, except for the use of AND-gate symbols. The AND-gate is used to show how the occurrence of a threat scenario in a path depends on the preceding threat scenarios.

There may be different reasons for why none of the alternatives in the other cases were clearly preferred. The alternatives we tested may have been too similar, which mean that the subjects considered them equally good or bad. It may also be the case that the subjects paid little attention to likelihoods of relations *between* threat scenarios, but were more interested in the likelihoods of the actual scenarios. The study resulted in the following decisions regarding the CORAS language:

- the logical AND and OR-gate symbols from FTA and circuit design were included in the language (Figure 68 illustrates the use of the AND gate, for a full explanation of the figure we refer to Sect. 8.4 in *Appendix G*).
- the likelihoods of paths are not shown, likelihood estimates are only assigned to threat scenarios and unwanted incidents (Figure 68).

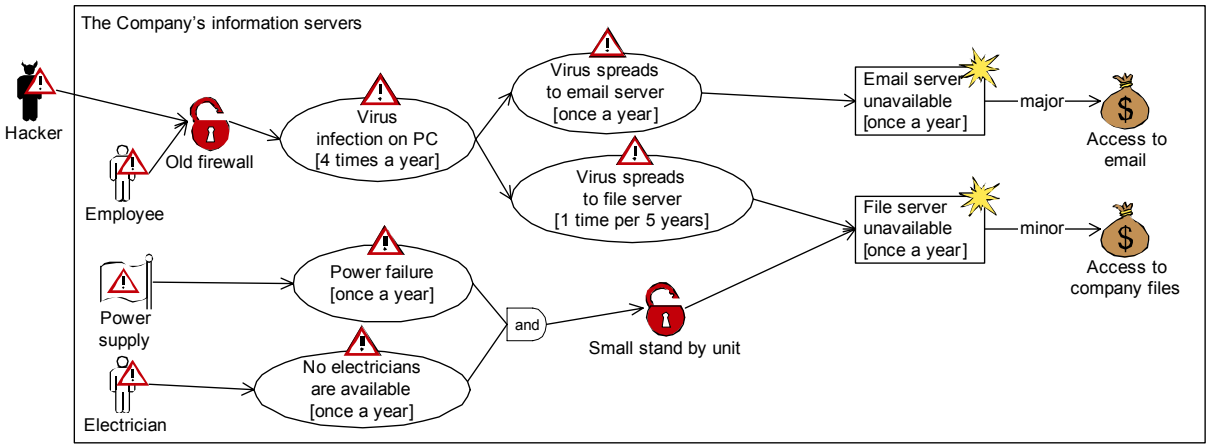


Figure 68 – Example of the use of AND-gate in the threat diagram

Vulnerabilities are interesting in two different settings: it is useful to get an overview of the vulnerabilities of each asset, and it may be interesting to get an overview of which vulnerabilities a threat may exploit along its way in a threat diagram. We did not find it preferred to illustrate both views in the same diagram. One may instead leave out threats, threat scenarios and relations between these when only interested in the assets and their vulnerabilities. Since the CORAS language should be able to express both views, the first should be covered in threat diagrams and the second in specialized asset diagrams. This resulted in the following design decision regarding the language:

- vulnerabilities may annotate all relations in a threat diagram (Figure 68). This way of representing vulnerabilities attracts attention to the actual point where they are exploited. To identify which vulnerabilities an asset is exposed to, one may follow each path from the asset in question towards the threats. Nevertheless, we believe it is better to present this latter information in a specialized asset diagram (Figure 69).

The asset diagram in Figure 69 specifies which vulnerabilities each asset is subject to:

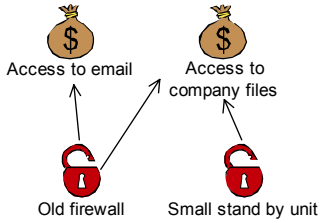


Figure 69 – Example of an asset diagram with particular focus on vulnerabilities

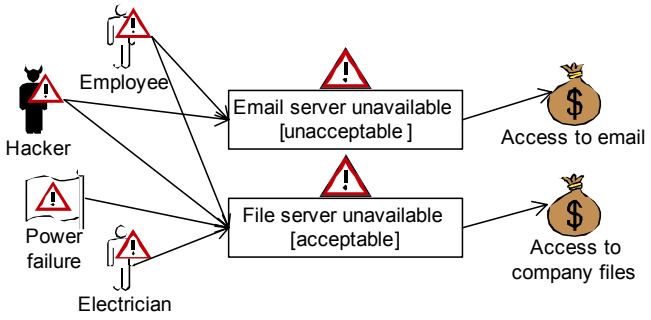


Figure 70 – Risk evaluation diagram illustrating whether the risks are acceptable or not

The risk evaluation diagram in Figure 70 illustrates which risks that are unacceptable/acceptable, the threats that may cause them and the assets that may be harmed. As we see, the employee and the hacker may cause both risks, while power failure and the electrician may only make the file server unavailable.

An important aspect of a security analysis is to identify the risks that are above the level of tolerance. The study indicates that it is preferred to use textual information labels to indicate the severity of risks and unwanted incidents, since size or color is too unambiguous. For the CORAS language, this means that textual information in the diagrams, possibly in combination with visual effects, is preferred over visual mechanisms alone. This resulted in the following design decisions regarding the language:

- we use text labels to indicate whether a risk is acceptable or unacceptable (Figure 70). The severity of unwanted incidents is not specified explicitly, but these are implicitly shown via their textual consequence- and likelihood labels in the threat diagram (Figure 68).

9.2 Evaluation Based on Experience Reports from Field Trials

In *Chapter 2*, we defined a set of success criteria for the thesis work. In this section, these are evaluated with respect to experience reports from industrial field trials. The field trials were carried out within the setting of the research project SECURIS (152839/220) funded by the Research Council of Norway. Each field trial consisted in a security analysis requiring about 250 person hours from the SINTEF analysis team, and 50-100 hours from the client. The analysis reports are confidential, but the scope and type of organization, and whether the author participated is shortly described in Table 25.

Table 25 – Recent field trials

Organization			**Target of analysis**	**Hogganvik participated?**
Name	**Year**	**Type (Approx. # employees)**		
DNV	2004	Vessel classification company (6400)	Information sharing service	Yes
NetCom	2004	Telecom company (700)	Mobile access to personal	Yes

			information	
Statnett	2005	Energy company (550)	Control and supervisory system	Yes
Hydro CSI	2005	Metal production company (1000, on plant)	Control and supervisory system	Yes
Hydro ICT	2006	ICT division, (metal, oil & energy) in Hydro (33 000, world wide)	Information sharing service	No
FLO/IKT	2006	Military organization (1000)	Procedures for handling restricted information	No
Statnett	2007[4]	Energy company (550)	Control and supervisory system	No

Two reports document every field trial: one presents the results from the security analysis and one summarizes experiences with the analysis methods and tools that were used. The first report was made available to the client only. The second report was open to the whole consortium. In this evaluation we consider only experience reports from the last five field trials: Hydro-05 trial [22], Statnett-05 trial [58], Hydro-06 trial [40], FLO/IKT-06 trial [132] and Statnett-07 trial [20]. This is to ensure that the statements reported are of adequate relevance since the way of conducting security analysis according to the CORAS method has evolved over the years (for example, before 2005 only the UML profile was used).

The experience reports provide information on issues such as what worked well, what did not work and why. The reports also contain data on how well the participants understand the diagrams, method, concepts etc. The author of the thesis did not participate in the last three field trials, but played a major role in the others.

The analysis team in the FLO/IKT-06 trial consisted of people with limited knowledge of the CORAS language. This field trial can therefore be seen as an exploration of the language and guideline by non-experienced modelers and their experiences are therefore extremely valuable. Some of their comments identified the need for additional explanations in the guideline that probably has been regarded as implicit by those familiar with the language. In most cases, their comments resulted in changes or the inclusion of new recommendations in the guideline.

In *Chapter 2* we defined the success criteria for the thesis work, targeting four main issues: the conceptual foundation, the language itself, the modeling guideline and the quality evaluation framework. In the evaluation based on the experience reports, only the first three are relevant. In the following, we discuss to what degree the validity of each success criterion is supported or not supported by the experience reports, and explain how identified problems have been addressed. All four success criteria are evaluated further in *Chapter 10*.

[4] Partly overlapping with the FLO/IKT-06 trial

9.2.1 The Conceptual Foundation

The following statements, relevant to the conceptual foundation, were found in the experience reports from the Hydro-05 trial [22], Hydro-06 trial [40], FLO/IKT-06 trial [132] and Statnett-07 trial [20] (nothing of particular relevance to the conceptual foundation was found in the Statnett-05 trial). Each statement is numbered, and followed by a discussion of its relevance including how we have dealt with it.

1. The CORAS terms used in the Hydro-05 trial were for instance threat, vulnerability, frequency, consequence and unwanted incident. Out of these, threat and unwanted incident were judged to be the easiest to grasp. Certain participants also found frequency quite understandable, while others thought this term was more problematic. Risk, consequence and asset were reported as less intuitive in the evaluation forms.

The reported difficulties in understanding the term "asset" is in our view mainly related to the abstract nature of the term itself. We have not seen the need for changing the definition, but have updated the guideline with an explanatory section addressing assets and asset types. Since the term "consequence" depends on the understanding of the term asset, we believe this will also reduce problems related to the notion of consequence. Although the notion of risk has proved difficult to understand in empirical investigations (*Chapter 5*), it has not been reported to be a problematic term in other field trials than Hydro-05. We therefore assume that this was a problem specific to this trial. It is however advisable to provide a careful introduction to the various terms at the start-up of a new analysis session, maybe even a discussion of them, to reach a common understanding. This has been included as a general recommendation in the guideline.

2. The asset identification and asset value estimation were found difficult during the Hydro-06 trial since the purpose of the activity and the asset-term itself were found difficult to understand. Also in the Statnett-07 trial and the FLO/IKT-06 trial, the analysis teams were uncertain of the difference between direct and indirect assets, and the purpose of this differentiation.

Our remarks regarding statement 1 with respect to the term assets is also relevant with respect to statement 2. The conceptual foundation does not distinguish between indirect and direct assets. In order to comply with international standards and be general enough to have interest beyond CORAS, the conceptual model is kept at a high level of abstraction. The differentiation between indirect and direct assets has been introduced in the guideline to help modelers achieve more precise diagrams, but is not really an issue of the conceptual model.

3. The ranking of assets according to importance created much discussion in the FLO/IKT-06 trial. The analysts assumed that this was the result of not all participants having a clear understanding of the concepts "asset", "consequence" and "frequency". There was a tendency to rank assets according to (expected) risk or frequency of related incidents, rather than importance.

The statement above shows that not having a clear understanding of the asset concept may affect also the understanding of related concepts. When the assets are precisely understood, the consequences in terms of potential damage to the asset are much easier to define. The misunderstood use of frequency during the ranking process is probably related to

uncertainty with respect to the ranking of risks. Frequency is one of the most commonly used risk-related terms in the common language. We believe its definition is quite easy to understand, but assigning frequencies, or likelihoods, during an analysis is always difficult. Therefore, we assume the difficulties regarding frequency in this trial was related to identifying the *correct* frequency of threat scenarios and unwanted incidents, rather than understanding the definition itself. Sect. 9.2.3 also addresses the problem of ranking assets.

4. In the FLO/IKT-06 trial, it was found useful to briefly repeat the meaning of central concepts in the beginning of each meeting to avoid confusion.

Although this may appear obvious, it was included into the guideline to help non-experienced security analysis teams.

5. The analysis team in the Hydro-06 trial suggests that the assets within the scope of the analysis should be kept at the same abstraction level.

Also this has later been included in the guideline since it may help non-experienced modelers to manage their challenges related to asset identification.

Conclusion: The first success criterion for the thesis work is defined as follows:

1. The work of this thesis should include a conceptual foundation for security analysis, that:

a. *specifies the main security analysis concepts and their relationships*
b. *uses a terminology that is easy to understand*
c. *uses a terminology that is in accordance with international standards*
d. *is described in a simple and understandable manner*
e. *provides a solid, underlying basis for a security risk modeling approach*

Of these only 1b and 1d are of relevance here. The main problem related to the conceptual foundation, which has been reported in the five experience reports addressed here, is related to the concept asset (statements 1, 2 and 3). Except from these statements, we have not found any complaints concerning the understandability of terms in the experience reports. This fact may actually suggest that the terminology used in CORAS is well defined and quite easy to understand.

Statement 4 and 5 are not criticisms of the conceptual foundation itself, but rather recommendations based on practical use of the CORAS method that may be helpful to inexperienced users and therefore have been included in the guideline.

9.2.2 The CORAS Language

Statements relevant to the CORAS language were found in experience reports from these field trials: Hydro-05 [22], Statnett-05 [58], FLO/IKT-06 [132] and Statnett-07 [20]. Each statement is numbered, and then followed by a discussion of its relevance, including how we have dealt with it.

1. The use of the graphical models made it easier to involve the participants actively, and helped ensure an effective communication between the analysis team and the participants [Hydro-05].

This statement supports the appropriateness of the CORAS language for use in structured brainstorming sessions.

2. The overall experience from the use of the threat-modeling tool[5] during the Statnett-07 trial to model threats "on-the-fly" compared to the previous approach, where the analyst team made the threat diagrams between workshop 1 and 2, were good. The threat diagrams capture threat scenarios in an intuitive manner and it seemed easy for the participants to relate to the diagrams and provide input during the brainstorming sessions. The subjective experience of the analyst secretary is that threat diagrams are much better suited for storing threat related information during sessions than tables, as the notation is more compact and flexible and more suited to the format of the information obtained.

This statement supports the appropriateness of the CORAS language for being used actively to model the findings during structured brainstorming sessions. The language is considered easy to understand by the participants, and in particular suitable for documenting the information that is gathered during each step of a security analysis.

3. The participants emphasized that they need proper introduction to the notation and sufficient time to understand the information presented to them [Hydro-05].

Since one may schedule the meetings in a security analysis with weeks in between, the representatives of the client may not remember the details of the language from the previous meetings. Moreover, some representatives may be new to the analysis. We have therefore updated the guideline to remind the analysis team to repeat definitions of notation in each meeting.

4. One should limit the amount of information in one diagram, and rather split it according to threat type, scenario type or asset type [Hydro-05].

It may be challenging to a non-experienced modeler to know how much information to capture in a single diagram. As a general recommendation, it is better to include too little information per diagram than too much. A rule of thumb is that the diagrams must be readable when projected on a whiteboard in a normal meeting room. To help non-experienced modelers we updated the guideline to reflect this.

5. It is important to strive towards correctness from the very beginning. This means that the participants must be involved early, enabling them to adjust the diagrams as they find appropriate [Hydro-05].

This is particularly important with respect to the target descriptions since they often are prepared before the meeting. The client should be able to recognize their intended target of analysis in the descriptions. One should not waste precious time in the meetings to correct erroneously diagrams since this draws attention away from the purpose of the meeting. To ensure this, it is advisable that the client reviews the target descriptions before the actual meeting. We later updated the guideline to reflect this.

[5] The CORAS tool for security analysis and threat modeling (http://coras.sourceforge.net)

6. The diagrams capture information gathered during brainstorming sessions and one of the main concerns of the participants was whether the diagrams would be complete [Hydro-05].

To help overcome this we advice the analysis team to utilize relevant checklists to ensure that every important aspect is covered. This kind of checklist may be part of the CORAS method. This is a recommendation in the guideline.

7. Any changes to the diagrams during the brainstorming session represent important information and the analysis secretary should capture their rationale [Hydro-05].

Gathering some notes on the decisions made during the analysis is an important task of the analysis secretary. Participants may be present at one meeting, but not the next and vice versa, and therefore it is necessary to capture the rationale behind each decision. This includes information about who provided the information, why and how it affected the diagrams. Depending on how the analysis is organized, it may be the responsibility of the analysis leader to take these notes so that the secretary may concentrate on modeling. To stress this issue, we have included it as a general recommendation in the guideline.

8. In the Hydro-05 trial, the participants found the notation itself easy to understand and remember. It was considered to be a good way of visualizing threat scenarios and very suitable for presentations. According to one of the participants, this type of visualization emphasizes the "message" or the purpose of the analysis.

This statement supports the CORAS language's appropriateness for brainstorming sessions. It also shows that the participants in the analyses perceive it to be easy to understand, and that it successfully supports and documents the findings from each step of the analysis process.

9. In the Statnett-05 trial, the client found the CORAS language of high value when conducting threat and vulnerability assessments of complex ICT systems. The visualization method was in particular suitable for presenting problems and alternative solutions, prior to a final decision. In fact, the graphical visualization provided by the threat diagrams revealed the need for some specific security measures that were directly accepted by the participants. The client also considered the method useful in training of personnel to increase their awareness of security related issues of complex ICT systems [9].

This emphasizes the previous statement since it is taken from a different field trial.

10. The client in the Hydro-05 trial considered the way the CORAS language helps specifying the relations between threats and the chain of events that they may cause, the various states of the target, and potential incidents, as one of its main benefits. According to the participants, the modeling method made them more conscious of the target of analysis and its risks by representing threats and vulnerabilities more explicitly than "just talking" about them.

This statement supports one of the main goals of the CORAS language: to facilitate a precise documentation of the security analysis relevant information. It also shows that the

participants in the analyses consider our graphical way of illustrating risks to make the security analysis more effective.

11. In the Hydro-05 trial, the participants said that the language provided an opportunity for documenting cause-consequence relations in a precise and detailed manner.

This statement also supports one of the main goals of the CORAS language, which is being a precise way of documenting the security analysis.

12. One of the hypotheses of the FLO/IKT-06 trial was that the use of CORAS diagrams facilitates the risk analysis process and documentation. The analysis team concluded that the hypothesis was confirmed, at least to the extent that it could be done without comparing the use of CORAS diagrams to alternative approaches. CORAS diagrams were used extensively in meetings #4 and #5 [risk identification and estimation]. The participants seemed to have a good understanding of the diagrams, and there were very little communication problems between the participants and the analysts. The analysis team assumed that one of the reasons for this was that the CORAS symbols are intuitive. The team also considered the technique of adding frequency values to the unwanted incidents, and consequence values to the unwanted incident-asset relations, during the risk estimation meeting to work very well.

In addition to fit the nature of the brainstorming session, the CORAS language should be easily understandable for the participants, and support and document the findings from each step of the analysis process. The result from the hypothesis testing in the statement above suggests that the CORAS language meets these requirements.

13. A justified question from the FLO/IKT-06 team is why threats but not vulnerabilities are included in the risk diagrams. As they point out, vulnerabilities are just as important as threats with respect to treatments.

The risk diagram was developed to provide a simple overview of the risks, from the initiator to the affected asset without any unnecessary "clutter". We understand the analysis team's concerns and recommend a feature like this to be implemented in the CORAS tool. This could for example be a function that lets the modeler right-click and choose "hide" on any element within a threat diagram.

14. Representatives from the Norwegian Defense and Research Establishment (FFI) observed the use of the CORAS approach during the Statnett-05 trial. Their evaluation summary [10] with respect to the language concluded: "Modeling = useful! Facilitates a common understanding and makes it easier to communicate the results".

Also the independent observer of this field trial agreed that the CORAS language facilitated the analysis process.

Conclusion: The second success criterion for the thesis work is defined as follows:

2. The work of this thesis should specify a language for describing security risks, that:
- *a. is suitable for use in structured brainstorming sessions*
- *b. is easily understandable for the participants in the brainstorming, including those who receive the analysis results afterwards*
- *c. has a precise syntax, meaning its design should be based on:*
  - *i. best practice within information visualization*
  - *ii. experiences with realistic security risk scenarios*
  - *iii. users' preferences*
  - *iv. existing risk modeling techniques*
- *d. has a structured semantics that translates arbitrary diagrams into English*
- *e. supports and documents the different steps in the security analysis process*

Of these, only 2a, 2b and 2e are of relevance here.

2a) The suitability of the CORAS language for use in structured brainstorming is supported by statements 1, 2, 8, 9, 12 and 14.

2b) According to statements 2, 8, 9, 12 and 14, the language is easy to understand for the participants in the analysis. However, there is little empirical data on how well people who have not participated in the actual analysis understand the language (see discussion in Sect. 9.4.2).

2e) Statements 2, 8, 9, 10, 11, 12 and 14 together indicate that the language is suited for documenting each step in a security analysis.

Statements 3, 4, 5, 6, 7 and 13 provide advice and hints related to the use of the language that we have later included as recommendations in the guideline, or suggestions for tool support.

9.2.3 The Guideline

The modeling guideline was applied for the first time in the Hydro-06 field trial. Statements relevant to this guideline are therefore found in experience reports from the Hydro-06 [40], FLO/IKT-06 [132] and Statnett-07 [20] field trials.

1. The guideline was applied for the first time in the Hydro-06 trial, and was found to be very helpful in risk identification.

Since the modeler in this field trial was a non-experienced user of the CORAS language, we interpret this statement as a confirmation of the guidelines suitability for non-experienced modelers.

2. In the Statnett-07 field trial, the team chose not to conduct the risk treatment as a separate workshop, but rather include it at the end of the risk estimation. This meant that the analysts did not have time to remove the acceptable risks before treatments were suggested.

Better tool-support is probably the best way to address problems of this kind.

3. In the FLO/IKT-06 trial, the analysts chose not to follow the guidelines for the risk identification. Afterwards they concluded that by following the guidelines a better result could have been achieved, in terms of simpler diagrams and less work before and after the meeting.

We have developed the guideline based on a large amount of experience from several field trials. If one chooses not to follow the guideline, it places responsibility on the analysis team. In our opinion, the only situation where this may be an option is if the analysis team is highly competent in security analysis and thereby knows when to take short cuts.

4. In the FLO/IKT-06 trial, each step of the guideline was evaluated, providing these judgments:
   - Context establishment:
     - The modeling guidelines for asset diagram in the context establishment raised some questions and caused some confusion related to the difference between direct and indirect assets. The analysis team also pointed out the problem of having point 4 (indicate with arrows which assets may affect other assets) after point 2 (place the direct assets within the region). The team felt it more natural to decide how assets affect other assets before separating indirect assets from direct assets.
     - The modeling guidelines for target descriptions were found very helpful and the analysis team managed to produce target descriptions that were well understood by the participants, and also useful for later analysis, by following the instructions closely and using suggested languages (UML class diagrams and activity diagrams).
     - The guidelines for the approval step were also found adequate.
   - Risk identification: Although the analysis team deviated from the guideline at this step, they believed that the modeling guidelines for risk identification were helpful, but that they did not give sufficient guidance to ensure that an inexperienced analysis team was able to conduct a successful risk identification step.
   - Risk estimation: the guideline provides sufficient support for this step.
   - Risk evaluation: the guideline provides sufficient support for this step.
   - Risk treatment: the guideline provides sufficient support for this step.

Based on this feedback from the FLO/IKT-06 trial we chose to change the order and text of point 2-4 for the asset diagram. In addition, we have added a more detailed explanation of assets to the guideline. Except for the risk identification step, the statements above show that the analysis team judged the different parts of the guideline to provide sufficient support. With respect to the risk identification, we have later updated the guideline with more explanations and recommendations. As mentioned before, the analysis team in this trial consisted of inexperienced CORAS users.

5. The FLO/IKT-06 trial suggested that the guidelines should allow direct assignment of likelihood values to unwanted incidents in cases where such estimates can be just as good as estimates for threat scenarios

Since one may use the CORAS language both in a high level manner, as well as at a more detailed level, this simplification should be allowed. We have later updated the guideline to reflect this decision.

6. The importance of ranking the assets (in asset diagrams and in the approval step) was not clear to the analysis team in FLO/IKT-06 since the guideline does not use this information in the later steps.

During the field trials, the ranking of assets has shown more problematic than anticipated. Often it creates much debate and discussion without being of great importance for the rest of the analysis. A ranking of assets may be achieved indirectly from the risk acceptance levels. An asset for which only a low risk level is accepted is "ranked" as more important than an asset for which one allows a higher risk level. The guideline has therefore eased this requirement and made it optional.

7. The Statnett-07 trial pointed out difficulties in deciding the correct level of abstraction for the analysis, especially for the consequence scale.

This is related to the whole security analysis process, and not specific to the modeling language used. In our opinion, it depends on the scope of the analysis and the experience of the analysis team. Our recommendation is to consult other analysts and previous analyses if available, or study the example given in the guideline to get an impression of how other analyses have been solved.

8. In the FLO/IKT-06 trial, the analysis team requested suggestions for asset, consequence and likelihood scales.

For a non-experienced analysis team this may be very helpful, but it is not related to the graphical language in particular. In our opinion, this should be included as part of the CORAS method description.

9. The analysis team in the Hydro-06 trial suggested that one should keep proposing examples of assets in order to engage the participants.

This comment is a result of the difficulties the analysis team had with the asset identification. Instead of proposing numerous possible assets, the team should instead carefully explain the notion of assets and by this involve the participants in the discussion. When an analysis team has conducted a couple of security analyses, it will be better prepared since the assets in the various analyses often have quite similar characteristics. To help non-experienced users, the CORAS method should provide a list of assets based on experiences from field trials that one may use as a checklist. Assets like a company's brand & reputation, the availability of a service, the effectiveness of a work task and more, are examples of real, but also very general assets.

10. The analysts in the Statnett-07 trial requested tips on how to structure the brainstorming session so that most of the participants would be actively involved.

This depends on the scope of the analysis and the selection of participants. If the participants cannot contribute to each other's areas, our recommendation is to split the brainstorming session into smaller sessions focusing on issues that are more specialized. If

participants feel they waste time by being present, they will become demoralized and not contribute to a fruitful discussion.

11. The FLO/IKT-06 trial requested guidance on how much time to spend on each step, and how to prioritize between the activities.

In general, no analysis is the same, and the time spent depends on the target complexity, the competence of the participants and more. The guideline provides a suggestion for the number of meetings and workshops that is expected during a CORAS analysis. It is up to the analysis team to judge this from case to case.

Conclusion: the third success criterion for the thesis work is defined as follows:

3. This thesis work should provide a modeling guideline for the modeler, that:
- *a. is rich and detailed with realistic examples*
- *b. addresses non-experienced modelers*
- *c. has recommendations based on user experiences*
- *d. follows the security analysis process step by step*

The statements extracted from the experience reports above, provide support for all of the sub criteria.

3a) we have extended the guideline with additional explanations to make the most difficult concepts more understandable (statements 2 and 3).

3b) several of the comments suggests that the guideline is suitable for non-experienced users, and after the extension with more explanations and recommendations it should have improved even more in this point (statements 1, 2, 3 and 4).

3c) the guideline is based on the field trials, and has been updated to include a number of practical recommendations based on experiences (statements 2, 3, 4, 7, 8, 9, 10 and 11).

3d) the guideline follows the security analysis process, and deviations from its instructions seem to be unwise (statements 2, 3 and 4).

9.3 Evaluation Using the Quality Framework

This section provides a summary of the quality evaluation of the CORAS language found in *Appendix H*. It includes an evaluation of how well the modeling language performs in modeling a set of "core security risk scenarios". It also presents the results and conclusions from the quality evaluation according to the language requirements described in *Chapter 6*.

9.3.1 Modeling the Core Security Risk Scenarios

The "core security risk scenarios" is a textual description of typical information gathered during a security risk analysis. The CORAS language has shown capable of modeling the relevant parts of this information. In the next paragraphs, we summarize the results from *Appendix H*.

Phase 1 – Context establishment:
The CORAS language was capable of modeling:
- The assets, including dependencies between them.
- The stakeholders.
- The indirect and direct assets.
- The risk acceptance levels of the stakeholder for each asset.
- The two stakeholders' conflicting risk acceptance level for the same asset.

This information was not modeled:
- The full description of the assets.
- The full description of the stakeholders.
- The scales of the assets, likelihood and consequence values.

When the stakeholders accept different risk values for an asset, the asset is split into two assets with reference to their stakeholder. If only one of the stakeholders value an asset, this is reflected by a specific reference to the stakeholder in the asset's name. We have chosen not to include more than the name of the assets and stakeholders in the models, because any extra information will clutter the diagram. The diagrams should be accompanied with textual explanations, which is a more appropriate place to give detailed descriptions of the elements in the diagram. Anything that are not directly necessary in order to read and understand the diagram should be avoided. The decision of not modeling the value scales was taken from a usability perspective. We believe this kind of information is more suitable for presentation in a table than in a graphical model because it only consists of static definitions.

Phase 2 – Risk identification:
The CORAS language was able to model:
- The threats, divided into human (accidental and deliberate) and non-human threats.
- The vulnerabilities, placed in a logical sequence of events with association to the threats that may exploit them.
- The threat scenarios with associations to relevant threats, vulnerabilities and unwanted incidents.
- The unwanted incidents with associations to the assets they may affect.
- A logical region symbolizing the target, for each type of threat.

This information was not modeled:
- The full description of the vulnerabilities.
- The full description of threats.
- The full description of assets.

As for the asset diagram, we decided to include only the names of the threats, vulnerabilities and assets to avoid messy and confusing diagrams. Any additional information about the elements should be stated in the accompanying textual description, which typically is required for the final analysis report.

Phase 3 – Risk estimation:
The CORAS language was able to model:
- The consequence of each unwanted incident towards the assets.
- The likelihood of each unwanted incident.

This information was not modeled:
- None.

It required very little effort to model the information from this phase using the CORAS language. In practice, the modeling consisted of adding textual information about likelihoods and consequences to the threat diagrams made in the previous phase. This makes the notation very suitable for modeling "on-the-fly" during risk estimation.

Phase 4 – Risk evaluation:
The CORAS language was able to express:
- The risks including their risk value.
- A separation of acceptable and unacceptable risks according to the risk acceptance levels of each stakeholder.

This information was not modeled:
- None.

The risk evaluation phase basically takes the information gathered in the risk identification and risk estimation and compares it to the risk acceptance levels from the first phase. The CORAS language had no problems modeling this information. The language provides a set of overview diagrams that can be very useful when presenting the findings of the analysis. The risk diagrams are created by "hiding" the vulnerabilities and threat scenarios between a threat and an unwanted incident in the threat diagrams. The unwanted incident symbol is replaced with a risk symbol with the appropriate risk value. Depending on the modelers choice of tools, this can be done either manually (like in Microsoft Visio) or semi-automatically in the CORAS tool. The risk evaluation diagram is a variation of the risk diagram where acceptable risks are grayed out. The latter diagram type is not necessary in order to model the core security risk scenarios.

Phase 5 – Treatment identification:
The CORAS language was able to express:
- The treatments, pointing towards the element in the threat diagram where they should be applied.
- An overview of the treatments that may be applied for each risk.
- An example of the risk diagram after applying a specific treatment.

This information was not modeled:
- The treatments' effect on the unacceptable risks.
- The estimated change in risk value each treatment may have on each of the unacceptable risks.

The decision of not modeling treatment effects in the diagrams is based on experience with the UML profile where this was a part of the notation. The treatment effect labels had a tendency of making the diagrams quite unreadable since they often conflicted with existing elements and relations. The CORAS language has the option for adding this kind of information to the treatment arrows, but we do not recommend using it, unless for very simple diagrams. The treatment overview diagram is an extra diagram type, suitable for presentation purposes in reports etc., but not required in order to model the core security risk scenarios. The CORAS language has no customized diagram for modeling the

expected effects of the treatments, but one may illustrate this using a risk diagram. In our opinion, it not necessary to spend time on modeling this information since it is more efficient to present it in a table like in the core security risk scenarios.

9.3.2 Evaluating Against the Quality Requirements

The table below summarizes the scores for each category of quality requirements in the quality framework (see Sect. 6.2). It is important to keep in mind that the number of requirements within each category and their level of details vary. We have not weighted the categories to adjust for this, and therefore the scores provide an *indication* of weak and strong aspects of the CORAS language. The highest possible average score for each topic within the categories is 3.

Table 26 – Evaluation scores

Category		Must-requirements			Should-requirements		
		Score	# req.	Avg.	Score	# req.	Avg.
Domain appropriateness:	- modeling perspectives	6	2	3		-	-
	- concepts	41,8	14	3		-	-
	- symbols	30	10	3	9	3*	3
Participant language knowledge appropriateness		9	3	3		-	-
Knowledge externalizability appropriateness		8	3	2,7	1	1	1
Comprehensibility appropriateness:	- internal representation	18	7*	2,3		-	-
	- external representation	17	6*	2,8	8	3	2,7
Technical actor interpretation appropriateness		3	1	3	3	1	3
Organizational appropriateness		8	3	2,7		-	-

"# req." denotes the number of requirements
"-" means no requirements of this kind
* requirements were omitted, see *Appendix H* for details

As we can see from the scores, the CORAS language is judged to have maximum appropriateness factor for being used during security analyses (domain appropriateness). This means that it has all the required modeling perspectives, and includes the necessary concepts and symbols related to the security analysis domain.

The CORAS language is also perceived as easy to learn and understand based on the users' familiarity with other modeling languages (participant language knowledge appropriateness). Its resemblance with use cases and fault trees makes it easy to understand for new users, even though they are not familiar with the notation.

Due to its already existing tool support and its formal, textual syntax, it has a high appropriateness factor for computerized tools (technical actor interpretation appropriateness).

There is however some aspects that still can be further improved, mainly related to comprehensibility appropriateness, the language's ability to express all security related scenarios (knowledge externalizability appropriateness) and its appropriateness for various types of organizations (organizational appropriateness).

The comprehensibility appropriateness of the internal representation of the language (its concepts and relationships between these) received the lowest score. As the investigations of security analysis terminology has shown, there are terms that are more difficult than other terms, for example risk and frequency measures (See details in *Appendices A* and *B*).

On the other hand, these findings are not found to the same extent in the realistic settings of the field trials. Based on this we have reasons to believe that the artificial setting of the questionnaires influences the results, and that the concepts in fact are better understood than the initial investigations indicate. The language have been found to deviate from the requirement of using general rather than specialized concepts since threat is specialized into human or non-human threats, and human threats are either categorized as having deliberate origin or accidental (i.e. sabotage or blunder). This decision was made from a usability point of view, since the threat concept was found too general to comprehend for novices to security analysis and it was necessary to bring it down to a more pragmatic level. This is an example of where one of the general requirements in the quality evaluation requirement conflicts with requirements that should ensure understandability.

The criticism regarding the comprehensibility appropriateness of the external representation of the CORAS language is valid for almost all modeling notations (UML included) and is related to drawing aesthetically pleasing models. When the CORAS diagrams grow in size they can easily become cluttered and difficult to read. One may avoid this by dividing the diagrams into smaller and more manageable pieces of information. Crossing lines may also be a problem, for example from treatment to treated entity, unwanted incident to assets or from threat to threat scenarios. This means that the modeler relies on experience to know when the maximum level of tolerable complexity in a diagram is reached, and a splitting is necessary.

The main difficulty of achieving full score in knowledge externalizability appropriateness is related to the brainstorming setting and its free and creative nature. Since it is highly recommended to document the findings from the brainstorming immediately, preferably by modeling it "on-the-fly", it sets high demands to the analysis secretary. The secretary has to actively model new information, change and update existing models during the entire brainstorming session. The language receives top score for all the must requirements in this section, but not entirely the one that says modeling should be easy to conduct as part of the work task. The quality requirement is defined from a general modeling language point of view, and does not take into account the special modeling situation in a security analysis. This means that it may have to be interpreted in a slightly different manner, and assuming the modeler has been provided sufficiently tool support, the requirement may in fact be sufficiently fulfilled.

One of the requirements for organizational appropriateness is that the language must help the organization to achieve its goals, which in a security analysis setting means controlling its security risks. In order to validate this requirement one should monitor the organization before and after the introduction of the CORAS language, but since this is beyond the scope of our work, the requirement is considered sufficiently supported, but not fully supported. Such an evaluation may be a subject of future work. We have however found it to ease the explanation of security risks within the organization, and this is an important aspect of the process of controlling risks.

The low score on knowledge externalizability appropriateness for the should-requirements is related to expressing event trees. Should-requirements are not that important to the language quality, but more of the type "nice-to-have". Although event tree is an important technique within security analysis, it requires additional constructs beyond the current syntax of the CORAS language. The intention or meaning of event trees can however be expressed, but then in a form that does not look like traditional event tree notation.

9.4 Threats to the Validity of the Evaluations

This section summarizes the main weaknesses of the evaluations of the CORAS language.

9.4.1 The Validity of the Design Investigation

In this section, we describe the main threats to validity of the investigation of graphical preferences from Sect. 9.1 (details in *Appendix G*). First, the diagrams we used were less detailed and complex than real diagrams, but more than just examples of notation. Real risk and threat diagrams contain more descriptive text, and this additional text might make the text labels less visible. This weakness is difficult to avoid, and we should therefore validate our findings by testing them in real threat diagrams in a future field trial.

In this design investigation, we included several alternative representations addressing a selection of security analysis modeling challenges. It would be both interesting and useful to test also other aspects and modeling alternatives, but we had to limit the size of the material to increase the likelihood of it being completed and returned by the subjects.

The various modeling alternatives were tested on a population with considerable technical background. A background that is similar to that of most participants involved in security analyses and therefore considered a representative population. Nevertheless, it would have been useful to confirm our findings by testing them on a group with less technical and more management related background.

Since the experiment material was distributed by email we could not control how the subjects chose to fill in the questionnaire (helping facilities, time spent etc.). We do not believe this affected the overall validity of the results since the subjects were asked to choose the alternatives they *preferred,* i.e. a completely subjective measure.

The subjects received no other introduction to security analysis than the short background provided with the material. We do not know whether the results would have been different if the subjects had been familiar with security analysis. One of the success factors of the CORAS language is however that people without a background in modeling and security analysis may understand it. Therefore, we believe that it was a good idea to first test the language on this type of population, and rather test it on subjects with more competence on security risk analysis at a later occasion.

9.4.2 Strengths and Weaknesses Related to the Field Trials

The SECURIS field trials took place in realistic settings and addressed real security issues within the organization of the client. The representatives of the client contributed to the diagrams by providing information during the brainstorming sessions, but leadership and modeling was the responsibility of the analysis team.

A clear weakness of the field trials is how we have collected empirical data. The evaluation reports are based on the analysts' opinions of how well the method, language and modeling tool worked (except for the Hydro-05 and Hydro-06 trials where the participants responded to questionnaires). From the Statnett-05 trial the client and the observers from the Norwegian Defense and Research Establishment (FFI) have summarized their experiences with the CORAS approach [9, 10]. This means that the evaluation involved a kind of "self-evaluation" of our own method and language. To make this self-evaluation of the CORAS language as independent as possible, the author of the thesis did not participate in the

analysis team for the last three field trials. The limited number of hours available often made the analysis team choose between completing the analysis tasks and spending the client's time on evaluation. When the aim of the field trials was to test CORAS in a realistic setting and deliver complete analysis results to the client (as if they had hired in a consultancy firm), the extent to which the client was involved in the evaluation part had to be reduced.

One of the points where we lack empirical support is related to how well the analysis documentation in form of CORAS models are understood by decision makers and others that receive the final analysis report. Within the scope of the SECURIS field trials there was no room for additional instigations beyond the actual security analyses, but this is subject to further work.

9.4.3 Threats to the Validity of the Quality Framework Evaluation

The quality framework evaluation consists of two parts, the core security risk scenarios and the quality requirement evaluation.

The core security risk scenarios represent a typical security risk analysis based on our experience from several industrial field trials [22-24, 58]. The scenarios give a step-by-step example of the information that is expected to be identified during the analysis and how it can be modeled. This can be used to compare different analysis and modeling strategies like a "benchmarking test". The scenarios are influenced by analyses within various industrial sectors that all had different targets and scopes. They have been carefully defined to make them as general as possible, but also include the appropriate level of details. Later field trials have shown the need for AND-gates, but the scenarios do not include an example of this. If one chooses to revise the scenarios, this is suggested for inclusion. The core security scenarios can to some extent be modeled in different styles, depending on the preferences of the modeler. Deciding how well a modeling language expresses the scenarios is partly a subjective measure. One the one side, the modeling language should be able to express the main parts of the information gathered at each step in the analysis, but on the other side one has to ensure that the amount of details does not affect the readability of the diagrams.

The quality framework SEQUAL has been developed for modeling languages in general. It has therefore been necessary to add several requirements that are specific to security risk analysis. One may claim that the set of extra requirements is incomplete or contains superfluous requirements, but at the time it was defined it seemed complete and well justified. Nevertheless, after two applications of the framework we have found some requirements to be overlapping, some are a bit irrelevant and some need a particular interpretation when applied to security analysis modeling languages (see *Appendix H*).

The quality evaluation framework has not been evaluated or tested by others than the author. Moreover, the second application of the framework was to evaluate a language made by the author. This is not sufficient to claim its validity as a general framework suitable for use by people unfamiliar with CORAS, SEQUAL and more. To support its validity, it should be tested by applying it to other security modeling languages, or at least use it for an independent evaluation of the CORAS language to see whether the results in *Appendix H* are confirmed.

10 Discussion

This chapter consists of two parts. In the first part, Sect.10.1, we evaluate and discuss to what extent we have reached our research objectives. In the second part, Sect. 10.2, we outline alternative research directives and identify some interesting issues for further research.

10.1 Fulfilling the Success Criteria

In *Chapter 2*, we characterized our research objectives in terms of success criteria. In the following, we investigate to what extent they have been fulfilled.

10.1.1 The First Success Criterion

1. The work of this thesis should include a conceptual foundation for security analysis, that:
    a. specifies the main security analysis concepts and their relationships
    b. uses a terminology that is easy to understand
    c. uses a terminology that is in accordance with international standards
    d. is described in a simple and understandable manner
    e. provides a solid, underlying basis for a security risk modeling approach

The sub criteria of the success criterion are supported as follows:

1a) The main security analysis concepts and their relationships are defined in the conceptual model in *Chapter 5*. See *Appendix A* and *Appendix B* for the full documentation.

1b) Many security and risk related concepts are used informally in English and other natural languages. To avoid misunderstandings we have conducted empirical investigations of how common people understand these concepts. We summarize the results of these in *Chapter 5* and present the full details in *Appendix A* and *Appendix B*. These investigations have helped ensure that the conceptual foundation captures security analysis concepts in accordance with common understanding. The experience reports from five industrial field trials [20, 22, 40, 58, 132] in which the CORAS language and conceptual foundation have been used, indicate very few problems related to the concepts and relations between them (see also *Chapter 9*).

1c) The underlying foundation supports the CORAS security analysis method which follows the Australian/New Zealand Standard for Risk Management [6]. This standard has later been incorporated in the ISO17799/ISO27002 standard [82]. Terms specific to information security complies with the ISO/IEC13335 standard [76].

1d) The conceptual model in *Chapter 5* is described using a simple UML 2.0 class diagram, and is accompanied with an explanatory example.

1e) The conceptual foundation for the CORAS language may function as a basis for security risk modeling approaches that use concepts from the international standards that the definitions in our foundation are taken from.

10.1.2 The Second Success Criterion

2. The work of this thesis should specify a language for describing security risks, that:
   a. is suitable for use in structured brainstorming sessions
   b. is easily understandable for the participants in the brainstorming, including those who receive the analysis results afterwards
   c. has a precise syntax, meaning its design should be based on:
      c.i best practice within information visualization
      c.ii experiences with realistic security risk scenarios
      c.iii users' preferences
      c.iv existing risk modeling techniques
   d. has a structured semantics that translates arbitrary diagrams into English
   e. supports and documents the different steps in the security analysis process

The sub criteria of the success criterion are supported as follows:

2a) The CORAS method is based on extensive use of structured brainstorming. The CORAS language has been designed for use in workshops where the diagrams are modeled "on-the-fly". Since the language may be used in high level as well as more detailed analyses, the diagrams may be extended with more information as the analysis progresses. In the industrial field trials [20, 22, 40, 58, 132] we have often received positive feedback with respect to the suitability of the CORAS language in brainstorming sessions. For instance, the client in the Statnett-05 field trial stated that the language was of high value when conducting threat and vulnerability assessments of complex ICT systems [9]. More evidence from the field trial evaluations supporting the fulfillment of this sub criterion is found in Sect. 9.2.2.

2b) The language has been judged easy to understand by participants in several field trials. Statements supporting this are provided in Sect. 9.2.2. According to one of the participants in the Hydro-05 trial, the language is considered to be a good way of visualizing threat scenarios and very suitable for presentation. We lack empirical data on how the CORAS models are understood by decision makers and others that read the final report, and this is suggested for further work.

2c) The graphical and textual syntaxes of the CORAS language are summarized in *Chapter 8*, and defined in detail in *Appendix D*.

2c.i) The empirical investigation summarized in Sect. 9.1 tested several different syntax alternatives adhering to best practice within information visualization, to identify the most suitable way of capturing relevant aspects syntactically. We built heavily on the results form these in the selection of concrete syntax. A professional, graphical designer has designed the icons with the aim to match the concepts they symbolize in an intuitive manner. For instance, we emphasized that the icons should be easily discriminated, both in color and in black/white print.

2c.ii) The syntax has evolved in order to handle the security analysis scenarios we have encountered in the field trials. The core security risk scenarios defined in *Chapter 6* are inspired by experiences from the field trials [22-24, 58], and as argued in *Appendix H* the CORAS language is well suited to capture these. Based on this, and experiences from the

field trials [20, 22, 40, 58, 132] we claim that the CORAS language can handle realistic security risk scenarios.

2c.iii) During the entire development of the CORAS language, we have had a strong focus on user preferences. The underlying foundation, on which the language builds, was defined to meet the users' intuitive understanding of the terminology (see investigations in Sect. 5.2). The decision on whether to use special risk related graphical icons was based on empirical investigations involving users (see investigation in Sect. 5.3). The final decisions regarding the concrete syntax were based on the results from the empirical investigation described in Sect. 9.1 which also focused on users.

2c.iv) The development has been inspired of well-known risk analysis techniques like fault trees and event trees (often called cause-consequence diagrams when put together). Since we identified the need for tree-like graphs with multiple top events (not allowed in fault trees) we could not adapt the fault tree notation completely. The intention of event trees may also be described. If one needs to express fault trees we have included the characteristic AND/OR gates from this notation. The inclusion of these operators is a result from the empirical investigation described in Sect. 9.1.

2d) The semantics of the CORAS language facilitates a schematic translation of any diagram into a paragraph in English (Sect. 8.1.3 and 8.1.4, full details in *Appendix D*). The semantics translates a diagram in two steps: (1) from the graphical syntax to the corresponding textual syntax, (2) from the textual syntax into corresponding sentences in English. The semantics is structured in the sense that we may translate each vertex and each relation separately. The meaning of a diagram as a whole is the set of sentences one gets from applying the semantic mapping to its vertices and relations.

2e) The modeling language offers customized diagrams for each step of the security analysis. Which kind of diagram that is suited for each step is described in the modeling guideline in *Chapter 8*, including detailed modeling instructions. Compared to traditional documentation techniques, which are mainly based on the use of tables that are filled in manually, the CORAS language is based on the vision that the users should concentrate on the graphical modeling of threat scenarios and let a tool extract the relevant information and automatically fill in tables. Field trial evaluation reports support the CORAS language's appropriateness for documenting the process in this manner (*Chapter 9*).

10.1.3 The Third Success Criterion

3. This thesis work should provide a modeling guideline for the modeler, that: a. is rich and detailed with realistic examples b. addresses non-experienced modelers c. has recommendations based on user experiences d. follows the security analysis process step by step

The sub criteria of the success criterion are supported as follows:

3a) The guideline provides an example-driven introduction to the CORAS method and CORAS language. The example addresses a security analysis of the same kind as those we have experience from in the field trials. In this example, we follow the analysis process from the start until the end. The guideline has been extended and revised several times

based on the experience reports [20, 22, 40, 58, 132], and practical recommendations has been included (Sect. 8.2.6).

3b) We have designed the guideline to address non-experienced modelers and statements found in the experience reports indicate that it really does (see Sect. 9.2.3). More recently, we have updated the guideline with an additional section (Sect. 8.2.6) containing recommendations, tips and hints based on experiences from practical use.

3c) The development of the guideline has been driven by experiences from the industrial field trials. From the experience reports we have gathered recommendations, tips and hints and included them into the guideline (see Sect. 9.2 and Sect. 8.2.6).

3d) The guideline in *Chapter 8* is structured according to the steps in a security analysis process. It describes the purpose of, and the expected outcome from each step, including detailed modeling instructions. The guideline also suggests which roles that should be present from the client's side at each meeting or workshop.

10.1.4 The Fourth Success Criterion

4. This thesis work should provide a framework for evaluating the quality of security risk modeling approaches, that: a. covers quality requirements related to modeling languages in general b. includes quality requirements special to security risk analysis c. includes a modeling test case representing core security risk scenarios

The quality evaluation framework defined in *Chapter 6* was first applied to evaluate the UML profile [124] and then to evaluate the CORAS language (*Appendix H*). The evaluations have provided a good overview of the strengths and weaknesses of both languages. The sub criteria of this success criterion are supported as follows:

4a) To ensure the inclusion of quality requirements relevant to modeling languages we have used the quality framework SEQUAL [100-104] as a basis when defining the quality requirements (see *Chapter 6*). SEQUAL provides a set of general requirements that any modeling language should fulfill.

4b) In addition to the quality requirements from SEQUAL, we have added requirements aiming to capture the specific needs of the security analysis domain. These are related to security analysis concepts and the relevant modeling needs identified through the industrial field trials.

4c) The quality evaluation framework also includes a realistic modeling test case (the core security risk scenarios) to ensure its practical usefulness. The core security risk scenarios are inspired of the analysis topics addressed in the DNV-04 [23], NetCom-04 [24], Statnett-05 [58] and Hydro-05 [22] field trials. These analyses are of course confidential, but they provided good indications regarding the complexity of the information gathered in a security analysis. They also increased our awareness of the challenges posed to a security risk modeling language. By embedding these experiences into general security related problems, relevant to any modern business organization, we ensured the relevance of the core security risk scenarios. The core security risk scenarios are used as a benchmarking

test with respect to the expressive power of the language, its modeling ease, elegance and more.

10.2 Things We Could Have Done Differently and Future Work

The research method we have used and the design we ended up with requires many choices on our behalf. In this section, we account for various alternative strategies and solutions and explain why we did not select them. We distinguish between strategies and solutions related to the design of our artifacts and strategies for evaluation. We also indicate some issues for future research.

10.2.1 Strategies and Solutions Related to Design

We believe one should adhere to international standards when defining risk related concepts for inclusion in the language. However, this does not ensure that these concepts match intuition. In our work, we particular experienced this kind of problem in relation to the concept asset. We believe it would have been fruitful to work more on the explanations related to asset identification; in particular develop more detailed guidelines for how to conduct the asset identification step of the analysis. The guidelines should focus on describing how to identify the "correct" assets, how to ensure that they are kept at the right level of abstraction, and how to define suitable consequence scales. We know this is one of the most difficult parts of the CORAS method, and it should be subject to further work.

Deciding upon, and designing the graphical icons for the language depends on two main tasks, first selecting a concept that represents the term, and second designing a graphical icon that represents the concept. For some of the terms we had problems in finding alternative concepts that could represent the term (e.g. vulnerability was difficult), whereas other terms were more straightforward (e.g. asset may be represented as monetary values). A professional designer made 1-3 alternative graphical icons for each concept, and we selected the one we felt represented the term best (for example for the asset we selected the moneybag icon over the coin and the gold barrel). Due to limited financial resources, it was not possible to have the icons tested on a broader test group in for instance an empirical study. It is quite expensive to hire a graphical designer, and the project could only fund one design iteration for the graphical icons. We are not aware of other concepts and icons that may represent the terms better, but cannot exclude that such exist.

A concept that may be included in future versions of CORAS (in the conceptual foundation, modeling language and guideline) is the notion of "barriers". Barriers represent existing security mechanisms in the target that have been implemented to reduce the likelihood of threat scenarios and unwanted incidents (e.g. a firewall is a barrier). Barriers may also be implemented to reduce the consequences of potential unwanted incidents. The concept of barriers can be very useful in the process of estimating risks. If the target has several strong barriers, the likelihood of unwanted incidents is reduced, whereas weak or non-existent barriers means that unwanted incidents are more likely to happen. One may also judge the strengths of the barriers by classifying them into different categories according to type. An "organizational barrier" in the form of a written security policy is weaker than a physically separated network, which represents a "physical barrier". A "symbolic barrier", in the form of a pop-up message that warns the user when he/she connects to an unsecured wireless network, is stronger than a policy against using

unsecured networks (organizational barrier), but weaker than a physical barrier that makes it impossible to access unsecured networks.

The current version of the CORAS approach provides little support for combining results from different analyses (for instance, analyses of two components of the same system). This kind of modularity might for example reduce the effort related to keeping an analysis documentation updated, since only the part where a change has been made needs to be re-analyzed. This has not been within the scope of our work, but ongoing research [21] is investigating this problem, and this may in the future result in new versions of the CORAS approach.

To facilitate more advanced tool-based support, a semantics based on translating to paragraphs in English, may not be sufficiently precise. A more precise semantics is necessary, particularly with respect to developing automatic analysis functions:

- From field trials, we know that there is a need for automatic computation of likelihood values of unwanted incidents.
- In treatment identification there is a need for functions that help choosing the optimal combination of treatments. These should be optimal with respect to risk level, costs, efficiency and more. This will require a more detailed specification of each treatment than the current version of the CORAS approach offers.
- The risk evaluation diagrams should be generated automatically based on the risk diagrams and the risk acceptance levels. This will also allow for an investigation of how an adjustment of the risk acceptance levels will affect the overall risk picture.

The development of a sufficiently precise semantics for such purposes has however, not been within the scope of our work, but is a subject for further work.

10.2.2 Strategies for Evaluation

The project context and setting of the industrial field trials required us to conduct security analysis according to the CORAS method in a professional manner, resembling commercial analyses. This was a prerequisite for involvement by many of the industrial partners. This of course, ensured that the CORAS method was tested in realistic settings, but unfortunately it also restricted our possibilities to explore different modeling and analysis strategies. It would for instance been interesting to analyze the same system twice for the same client, one using the CORAS language and one using their own approach to see the differences. There are of course a number of practical and economical considerations related to this kind of experiments. Amongst others, the client would have to provide two sets of participants with similar competence (often very difficult in practice) and the time they spend on the second analysis might have to be compensated financially.

To improve the guideline further, it would be interesting to test it on users without previous background in CORAS and get them to report their experiences. Within our research project, there have not been resources to carry out this kind of test.

One of the points where we lack empirical support is related to how well the documentation in the form of models in the security analysis reports are understood by decision makers and others who are supposed to make use of the results and recommendations. This has much to do with the confidentiality aspects of the analyses. One way we may test this is to have the client pointing out a trusted person within its organization, who have no previous knowledge of the analysis, and have him/her to read the analysis report. The analysis team could then interview this person to see whether the

diagrams were understood. It would also be interesting to revisit the client after some time, to see how the findings of the analysis have had consequences for the organization. This has however not been possible within the setting of our work, but may be considered in the future.

The structured semantics has only been tested by the developers. An interesting experiment would be to have one student group translating a set of diagrams into sentences in English, and then have a different group modeling these textual descriptions back to diagrams. By doing this, one could explore whether some information is lost in the translation. If so, it means that there is some information in the diagrams that is not captured by the semantics. This could for instance be the order or the placement of elements in the diagram.

References

[1] Aerospace Recommended Practice (ARP) 4761: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment: Society of Automotive Engineers Inc, 1996.

[2] AIChE, "Guidelines for Chemical Process Quantitative Risk Analysis", American Institute of Chemical Engineers, Center for Chemical Process Safety, 1989.

[3] Alberts, C., J. ; and Dorofee, A. J., "OCTAVE Criteria Version 2.0", Tech. report CMU/SEI-2001-TR-016. ESC-TR-2001-016, 2001.

[4] Alghathbar, K. and Wijesejera, D., "authUML: a three-phased framwork to analyze access control specifications in use cases", in Proc. ACM Workshop on Formal methods in security engineering (FMSE'03), pp. 77-86, 2003.

[5] Appukkutty, K., Ammar, H. H., and Goseva-Popstojanova, K., "Software Requirement Risk Assessment using UML (Short paper)", in Proc. ACS/IEEE International Conference on Computer Systems and Applications, pp. 112-115, 2005.

[6] AS/NZS4360, Australian/New Zealand Standard for Risk Management: Standards Australia/Standards New Zealand, 2004.

[7] AS/NZS4360, Australian/New Zealand Standard for Risk Management: Standards Australia/Standards New Zealand, 1999.

[8] Barber, B. and Davey, J., "The Use of the CCTA Risk Analysis and Management Methodology CRAMM in Health Information Systems", in Proc. MEDINFO'92, pp. 1589-1593, 1992.

[9] "BAS5 - Statnett-case ved bruk av CORAS-metoden: ROS-analyse av IKT-systemene ved RS' kontrollsenter" (document summarizing the client's experiences), Statnett, 2006-03-06.

[10] "BAS5: ROS og caseanalyser " (power point presentation that summarizes the experiences). FFI - Forsvarets forskningsinstitutt, 2006-03-06.

[11] Basin, D., Doser, J., and Lodderstedt, T., "Model Driven Security for Process-Oriented Systems", in Proc. 8th ACM symposium on Access control models and technologies (SACMAT'03), pp. 100-110, 2003.

[12] Basin, D., Doser, J., and Lodderstedt, T., "Model Driven Security: from UML Models to Access Control Infrastructures", *ACM Transactions on Software Engineering and Methodology*, vol. 15 (1), pp. 39-91, 2006.

[13] Baskerville, R. L., "Investigating Information Systems with Action Research", *Communications of the Association for Information Systems*, vol. 2 (19), 1999.

[14] Becker, R. A., Eick, S. G., and Wilks, A. R., "Visualizing Network Data", *IEEE Transactions on Visualization and Computer Graphics*, vol. 1 (1), pp. 16-21, 1995.

[15] Bell, R., "Introduction to IEC 61508", in Proc. 10th Australian Workshop on Safety Critical Systems and Software (SCS'05), Conferences in Research and Practice in Information Technology, (55), pp. 3-12, 2005.

[16] Bistarelli, S., Fioravanti, F., and Peretti, P., "Defense tree for economic evaluations of security investment", in Proc. 1st Int. Conference on Availability, Reliability and Security (ARES'06), pp. 416-423, 2006.

[17] Bouti , A. and Kadi , A. D., "A state-of-the-art review of FMEA/FMECA", *International Journal of Reliability, Quality and Safety Engineering*, vol. 1, pp. 515-543, 1994.

[18] Bratthall, L. and Wohlin, C., "Is it Possible to Decorate Graphical Software Design and Architecture Models with Qualitative Information? - An Experiment", *IEEE Transactions on Software Engineering*, vol. 28 (12), pp. 1181-1193, 2002.

[19] Britton, C. and Jones, S., "The Untrained Eye: How Languages for Software Specification Support Understanding in Untrained Users", *Human Computer Interaction*, vol. 14 (1&2), pp. 191-244, 1999.

[20] Brændeland, G. and Dahl, H., "Evaluation of the method and tool used during the 10th SECURIS field trial" (Statnett). Memo, SINTEF ICT, 2007.

[21] Brændeland, G., Dahl, H., Engan, I., and Stølen, K., Using Dependent CORAS Diagrams to Analyse Mutual Dependency, in *Proc. of CRITIS'07 (to be published)*, 2007.

[22] Brændeland, G., Hogganvik, I., and Engan, I., "Evaluation of the methodology and tool used during the 7th SECURIS field trial" (Hydro CSI). Tech.rep., STF90 A0631, SINTEF ICT, 2005.

[23] Brændeland, G., Hogganvik, I., and Seehusen, F., "Evaluation of risk analysis methodology and tools used during the 4th SECURIS field trial" (DNV). Tech.rep., STF90 F04081, SINTEF ICT, 2004.

[24] Brændeland, G., Hogganvik, I., and Seehusen, F., "Evaluation of the methodology and tool used during the 5th SECURIS field trial" (NetCom). Tech.rep., STF90 F05400, SINTEF ICT, 2005.

[25] BS7799-1, Information Technology - Code of practice for information security management (since 2000 adopted by ISO/IEC17799) British Standards Institute (BSI), 1995.

[26] BS7799-2, Information Security Management Systems - Specification with guidance for use (replaced by ISO27001): British Standards Institute (BSI), 1999.

[27] Cahill, M.-C. and Carter, R. C. J., "Color code size for searching displays of different density", *Human Factors*, vol. 18 (3), pp. 273-280, 1976.

[28] Cavanagh, J. P., "Relationship between the immediate memory span and the memory search rate", *Psychological Review*, vol. 79, pp. 525-530, 1972.

[29] Chattratichart, J. and Kuljis, J., "An Assessment of Visual Representations for the 'Flow of Control'", in Proc. 12th Workshop of the Psychology of Programming Interest Group (PPIG'00), pp. 45-48, 2000.

[30] Christ, R. E., Research for evaluating visual display codes: an emphasis on colour coding., in *Information Design: The design and evaluation of signs and printed material*, R. Easterby and H. Zwaga, Eds.: John Wiley and Sons Ltd, 1984, pp. 209-228.

[31] Christ, R. E., "Review and analysis of color coding research for visual displays", *Human Factors*, vol. 17 (6), pp. 542-570, 1975.

[32] Cleveland, W. S. and McGill, R., "Graphical perception and graphical methods for analyzing scientific data." *Science*, vol. 229, pp. 828-833, 1985.

[33] "Common Criteria for Information Technology Security Evaluation v3.1", CCMB-2006-09-001, 2006.

[34] Cortellessa, V., Goseva-Popstojanova, K., Appukkutty, K., Guedem, A., Hassan, A. E., Elnaggar, R., Abdelmoez, W., and Ammar, H. H., "Model-based performance risk analysis", *IEEE Transactions on Software Engineering*, vol. 31 (1), pp. 3-20, 2005.

[35] CSI/FBI, "COMPUTER CRIME AND SECURITY SURVEY", 2005.

[36] Dahl, H. E. I., Hogganvik, I., and Stølen, K., "Structured Semantics for the CORAS Security Risk Modelling Language." in Proc. 2nd Int. Workshop on Interoperability Solutions on Trust, Security, Policies and QoS for Enhanced Systems (IS-TSPQ'07), pp. 79-92, 2007.

[37] Davison, R., Martinsons, M. G., and Kock, N., "Principles of canonical action research", *Information Systems Journal*, vol. 14 (1), pp. 65-86, 2004.

[38] Demarco, T. and Plauger, P. J., *Structured Analysis and Systems Specification.* : Prentice-Hall, 1979.

[39] Doan, T., Demurjian, S., Ting, T. C., and Ketterl, A., "MAC and UML for Secure Software Design", in Proc. ACM Workshop on Formal Methods in Security Engineering (FMSE'04), pp. 75-85, 2004.

[40] Engan, I., Lund, M. S., and Torsbakken, S., "Evaluation of the method and tool used during the 8th SECURIS field trial" (Hydro ICT). Memo, SINTEF ICT, 2007.

[41] Fenton, N., Krause, P., and Neil, M., "Software Measurement: Uncertainty and Causal Modeling", *IEEE Software*, vol. 19 (4), pp. 116-122, 2002.

[42] Fenton, N. and Neil, M., Combining evidence in Risk Analysis using Bayesian Networks: Agena White Paper, W0704/01, v01.01, 2004.

[43] Fenton, N. and Ohlsson, N., "Quantitative Analysis of Faults and Failures in a Complex Software System", *IEEE Transactions on Software Engineering*, vol. 26 (8), pp. 797-814, 2000.

[44] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R., "Proposed NIST standard for role-based acess control", *ACM Transactions on Information System Security (TISSEC)*, vol. 4 (3), pp. 224-274, 2001.

[45] Freeman, R. E., *Strategic Management: A Stakeholder Approach*: Ballinger Publishing, 1984.

[46] Gane, C. and Sarson, T., *Structured Systems Analysis: Tools and Techniques*: Mcdonnell Douglas Infomation, 1977.

[47] Genrich, H. J., "Predicate/transition nets", in *Advances in Petri Nets* vol. LNCS 254-255, W. Brauer, W. Resig, and G. Rozenberg, Eds.: Springer-Verlag, 1987, pp. 207-247.

[48] Georg, G., France, R., and Ray, I., "An Aspect-Based Approach to Modeling Security Concerns", in Proc. CSDUML'02, pp. 107-120, 2002.

[49] Goodman, N., *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis: Hackett, 1976.

[50] Goseva-Popstojanova, K., Hassan, A. E., Guedem, A., Abdelmoez, W., Nassar, D. E. M., Ammar, H. H., and Mili, A., "Architectural-level risk analysis using UML", *IEEE Transactions on Software Engineering*, vol. 29 (10), pp. 946-960, 2003.

[51] Green, T., "Cognitive dimensions of notations", in Proc. Human Computer Interaction (HCI'89), People and Computers V, pp. 443-459, 1989.

[52] Green, T., "Describing information artefacts with cognitive dimensions and structure maps", in Proc. Human Computer Interaction (HCI'91), People and Computers VI, pp. 297-317, 1991.

[53] Hassan, A. E., Goseva-Popstojanova, K., and Ammar, H. H., "UML Based Severity Analysis Methodology", in Proc. RAMS'05, pp. 158-164, 2005.

[54] HB231, Information security risk management guidelines: Standards Australia/Standards New Zealand, 2000.

[55] HB231, Information security risk management guidelines: Standards Australia/Standards New Zealand, 2004.

[56] Hevner, A. R., March, S. T., Park, J., and Ram, S., "Design Science in Information Systems Research", *MIS Quarterly*, vol. 28 (1), pp. 75-105, 2004.

[57] Hogganvik, I., Lund, M. S., and Stølen, K., "Quality Evaluation of the CORAS UML profile", SINTEF ICT, Tech.rep. SINTEF A2199, 2007.

[58] Hogganvik, I., Seehusen, F., and Solhaug, B., "Evaluations of the methodology and tool used during the 6th SECURIS field trial" (Statnett). Tech.rep., STF90 A06032, SINTEF ICT, 2005.

[59] Howard, M. and LeBlanc, D., *Writing Secure Code*, 2 ed: Microsoft Press, 2003.

[60] Howard, M. and Lipner, S., *The Security Development Lifecycle*: Microsoft Press, 2006.

[61] Howard, R. A., *Dynamic Probabilistic Systems: Volume 1: Markov models*: John Wiley & Sons, 1971.

[62] Humprey, A., "SWOT - Strengths, Weaknesses, Opportunities, Threats" Standford University, 1960-1970.

[63] IEC16085, Information technology - Software life cycle processes - Risk management, 2004.

[64] IEC60300-3-9, Dependability management - Part 3: Application guide - Section 9: Risk analysis of technological systems - Event Tree Analysis (ETA), 1995.

[65] IEC60812, Analysis techniques for system reliability - Procedures for failure mode and effect analysis (FMEA and FMECA), 1985.

[66] IEC61025, Fault Tree Analysis (FTA), 1990.

[67] IEC61078, Analysis techniques for dependability - Reliability block diagram method, 1991.

[68] IEC61165, Application of Markov techniques, 1995.

[69] IEC61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related (E/E/PE) Systems: IEC, 1998-2000.

[70] IEC61882, Hazard and operability studies (HAZOP studies) - Application guide, 2001.

[71] ISO9001, Quality management systems – Requirements, 2000.

[72] ISO14001, Environmental management systems – Requirements with guidance for use, 2004.

[73] ISO15408, Information technology – Security techniques – Evaluation criteria for IT security (Common Criteria), 2005.

[74] ISO21827, Information technology – Systems Security Engineering – Capability Maturity Model, 2002.

[75] ISO27001, Information technology – Security techniques – Information security management systems – Requirements, 2005.

[76] ISO/IEC13335-1, Information technology – Security techniques – Management of information and communications technology security - Part 1:Concepts and models for information and communications technology security management, 2004.

[77] ISO/IEC13335-3, Information technology – Guidelines for the management of IT Security – Part 3: Techniques for the management of IT Security, 1998.

[78] ISO/IEC13335-4, Information technology – Guidelines for the management of IT Security – Part 4: Selection of safeguards, 2000.

[79] ISO/IEC13335-5, Information technology – Guidelines for the management of IT Security – Part 5: Management guidance on network security, 2001.

[80] ISO/IEC14516, Information technology – Security techniques – Guidelines for the use and management of Trusted Third Party services, 2002.

[81] ISO/IEC14977, Information technology – Syntactic metalanguage – Extended Backus-Naur Form (EBNF), 1996.

[82] ISO/IEC17799, Information technology – Security techniques – Code of practice for information security management (ISO27002), 2005.

[83] ISO/IEC18028, Information technology – Security techniques – IT network security, 2005-2006.

[84] ISO/IEC20000, Information technology – Service management (Part 1 and 2), 2005.

[85] ISO/IECTR18044, Information technology – Security techniques – Information security incident management, 2004.

[86] ITU-T X.800: Security architecture for open system interconnection for CCITT applications. (Technically aligned with ISO 7498-2), 1991.

[87] Johnson, W. G., *MORT Safety Assurance Systems*: Marcel Dekker, 1980.

[88] Jubis, R. M. T., "Coding effects on performance in a process control task with uniparameter and multiparameter displays", *Human Factors*, vol. 32 (3), pp. 287-297, 1990.

[89] Jürjens, J., *Secure Systems Development with UML*: Springer, 2005.

[90] Jürjens, J., "Sound Methods and Effective Tools for Model-based Security Engineering with UML", in Proc. Int. Conference on Software Engineering (ICSE'05), pp. 322-331, 2005.

[91] Jürjens, J., "UMLsec: Extending UML for secure systems development", in Proc. UML 2002 - The Unified Modeling Language, (LNCS 2460), pp. 412-425, 2002.

[92] Jøsang, A., Bradley, D., and Knapskog, S. J., "Belief-Based Risk Analysis", in Proc. Australasian Information Security Workshop 2004 (AISW'04), (32), pp. 63-68, 2004.

[93] Kavianian, H. R., Rao, J. K., and Brown, G. V., "Application of Hazard Evaluation Techniques to the Design of Potentially Hazardous Industrial Chemical Processes", U.S. National Institute for Occupational Safety and Health, 1992.

[94] Kemeny, J. G. and Snell, J. L., *Finite Markov chains*: Springer-Verlag, 1976.

[95] Kemmerer, R. A., "Cybersecurity", in Proc. Int. Conference on Software Engineering (ICSE'02), pp. 705-717, 2002.

[96] Kjellén, U., Tinmannsvik, R. K., Ulleberg, T., Olsen, P. E., and Saxvik, B., *SMORT - Sikkerhetsanalyse av industriell organisasjon (Norwegian only)*: Yrkeslitteratur, 1987.

[97] Kontio, J., Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation, in *Dept. of Computer Science and Engineering*: Helsinki University of Technology, 2001.

[98] Krause, P. and Clark, D., *Representing Uncertain Knowledge: An Artificial Intelligence Approach*: Intellect Press, 1993.

[99] Krogstie, J., Conceptual Modeling for Computerized Information Systems Support in Organizations, in *Faculty of Electrical Engineering and Computer Science*: The Norwegian Institute of Technology, The University of Trondheim, 1995.

[100] Krogstie, J., "Evaluating UML Using a Generic Quality Framework", in *UML and the Unified Process*, L. Favre, Ed.: IRM Press, 2003, pp. 1-22.

[101] Krogstie, J., "Using a semiotic framework to evaluate UML for the development of models of high quality", in *Unified Modeling Language: Systems analysis, design, and development issues*, K. Siau and T. Halpin, Eds.: Idea Group Publishing, 2001, pp. 89-106.

[102] Krogstie, J. and Arnesen, S. d. F., "Assessing Enterprise Modeling Languages Using a Generic Quality Framework", in *Information Modeling Methods and Methodologies*: Idea Group, 2005, pp. 63-79.

[103] Krogstie, J., Lindland, O. I., and Sindre, G., "Defining quality aspects for conceptual models", in Proc. IFIP8.1 Working Conference on Information Systems Concepts (ISCO3), Towards a Consolidation of Views, pp. 216-231, 1995.

[104] Krogstie, J. and Sølvberg, A., *Information Systems Engineering: Conceptual Modeling in a Quality Perspective*: Kompendiumforlaget, 2003.

[105] Kuzniarz, L., Staron, M., and Wohlin, C., "An Empirical Study on Using Stereotypes to Improve Understanding of UML Models", in Proc. 12th Int. Workshop on Program Comprehension (IWPC'04), pp. 14-23, 2004.

[106] Larkin, J. H. and Simon, H. A., "Why a Diagram Is (Sometimes) Worth Ten Thousand Words", *Cognitive Science*, vol. 11, pp. 65-99, 1987.

[107] Lees, F. P. and Mannan, S., *Lees' loss prevention in the process industries : hazard identification, assessment and control*, vol. 1, 3rd ed: Elsevier Butterworth-Heinemann, 2005.

[108] Lindley, D. V., *Introduction to Probability and Statistics from a Bayesian Viewpoint*: Cambridge University Press, 1965.

[109] Linos, P. K., Aubet, P., Dumas, L., Helleboid, Y., Lejeune, D., and Tulula, P., "Visualizing program dependencies: An experimental study", *Software Practice and Experience*, vol. 24 (4), pp. 387-403, 1994.

[110] Lodderstedt, T., Basin, D., and Doser, J., "SecureUML: A UML-Based Modeling Language for Model-Driven Security", in Proc. UML'02, LNCS, (2460), pp. 426-441, 2002.

[111] Lund, M. S., den Braber, F., Stølen, K., and Vraalsen, F., "A UML profile for the identification and analysis of security risks during structured brainstorming." SINTEF ICT, Tech.rep. STF40 A03067, 2004.

[112] Lund, M. S., Hogganvik, I., Seehusen, F., and Stølen, K., "UML profile for security assessment", SINTEF ICT, Technical report STF40 A03066, 2003.

[113] March, S. T. and Smith, G. F., "Design and natural science research on information technology", *Design Support Systems, Elsevier Science*, vol. 15, pp. 251-266, 1995.

[114] Martin, N., "Simple Technique for Illustrating Risk (STIR)", SANS Institute, 2002.

[115] McDermott, J., "Abuse-case-based assurance arguments", in Proc. 17th Computer Security Applications Conference (ACSAC'O1), pp. 366-374, 2001.

[116] McDermott, J., "Visual Security Protocol Modeling", in Proc. New Security Paradigms Workshop (NSPW'05), pp. 97-109, 2005.

[117] McDermott, J. and Fox, C., "Using abuse case models for security requirements analysis", in Proc. 15th Computer Security Applications Conference (ACSAC'99), pp. 55-66, 1999.

[118] McGrath, J. E., *Groups: interaction and performance*: Prentice-Hall, 1984.

[119] MEHARI (Méthode Harmonisée d'Analyse de Risques Informatiques): CLUSIF (Club de la Sécurité de l'Information Français), 2004.

[120] Microsoft, "The Security Risk Management Guideline", Microsoft Solutions for Security and Compliance, Microsoft Security Centre of Excellence, 2006.

[121] Myagmar, S., Lee, A. J., and Yurcik, W., "Threat Modeling as a Basis for Security Requirements", in Proc. Symposium on Requirements Engineering for Information Security (SREIS'05), 2005.

[122] Nielsen, D. S., *The Cause/Consequence Diagram Method as a Basis for Quantitative Accident Analysis*: Danish Atomic Energy Commission, RISO-M-1374, 1971.

[123] NIST, "Risk Management Guide for Information Technology Systems", U.S. National Institute of Standards and Technology (NIST), NIST Special Publication SP800-30, 2002.

[124] OMG, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms", Object Management Group, 2006.

[125] OMG, "Unified Modeling Language (UML): Superstructure, version 2.0", Object Management Group, 2005.

[126] Pauli, J. and Xu, D., "Threat-driven architectural design of secure information systems", in Proc. 7th International Conference on Enterprise Information Systems (ICEIS'05), pp. 136-143, 2005.

[127] Poulsen, K., Sluggish movement on power grid cyber security, in *SecurityFocus* http://www.securityfocus.com/news/9328. 2004-08-13.

[128] PricewaterhouseCoopers, "The Information Security Breaches Survey 2006", UK Department of Trade and Industry, 2006.

[129] Rausand, M. and Høyland, A., *System reliability Theory: Models, Statistical Methods, and Applications*, 2 ed: Wiley, 2004.

[130] Ray, I., Li, N., France, R., and Kim, D.-K., "Using UML To Visualize Role-Based Access Control Constraints", in Proc. SACMAT'04, pp. 115-124, 2004.

[131] Redmill, F., Chudleigh, M., and Catmur, J., *HAZOP and Software HAZOP*: Wiley, 1999.

[132] Refsdal, A. and Solhaug, B., "Evaluations of the methodology and tool used during the 9th SECURIS field trial" (FLO/IKT). Tech.rep., SINTEF A1532, SINTEF ICT, 2007.

[133] *Risk and Reliability - An Introductory Text*, 5 ed: Risk & Reliability Associates (R2A), 2005.

[134] Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modeling Language Reference Manual*: Addison Wesley Longman, Inc., 1998.

[135] Sampson, G., *Writing Systems*: Hutchinson, 1985.

[136] Savage, L. J., *The Foundations of Statistical Inference*: J. Wiley, 1962.

[137] Schneider, W. and Shiffren, R. M., "Controlled and automatic human information processing I: Detection, search, and attention." *Psychological Review*, vol. 84, pp. 1-66, 1977.

[138] Schneier, B., "Attack trees: Modeling security threats", *Dr. Dobb's Journal*, vol. 24 (12), pp. 21-29, 1999.

[139] Schneier, B., *Secrets & Lies: Digital Security in a Networked World*: John Wiley & Sons, 2000.

[140] Sedra, A. S. and Smith, K. C., *Microelectronic Circuits*: Oxford University Press, 2003.

[141] Seehusen, F. and Stølen, K., "Graphical specification of dynamic network structure." in Proc. 7th Int. Conference on Enterprise Information Systems (ICEIS'05), (3), pp. 203-209, 2005.

[142] Shneiderman, B., *Designing the User Interface*: Addison-Wesley, 1992.

[143] Siegel, S. and Castellan, J., *Non-parametric Statistics for the Behavioural Sciences*, 2 ed: McGraw-Hill International Editions, 1988.

[144] Simon, H. A., *The Sciences of the Artificial*, 3 ed: MIT Press, 1996.

[145] Sindre, G. and Opdahl, A. L., "Capturing Security Requirements through Misuse Cases", in Proc. 14th Norwegian Informatics Conference (NIK'2001), pp. 219-230, 2001.

[146] Sindre, G. and Opdahl, A. L., "Eliciting Security Requirements by Misuse Cases", in Proc. TOOLS-PACIFIC, pp. 120-131, 2000.

[147] Sindre, G. and Opdahl, A. L., "Templates for Misuse Case Description", in Proc. Workshop of Requirements Engineering: Foundation of Software Quality (REFSQ'01), pp. 125-136, 2001.

[148] Smith, L. and Thomas, D., "Color versus shape coding in information displays", *Journal of Applied Psychology*, vol. 48 (3), pp. 137-146, 1964.

[149] Song, H. and Leangsuksun, C., "A Framework for Cluster Availability Specification and Evaluation", in Proc. 43th ACM Southeast Conference (Student poster in "Architecture and distributed system session"), pp. 202-203, 2005.

[150] Spiegelhalter, D. J., "Probabilistic reasoning in predictive expert systems", in Proc. 2nd Annual Conference on Uncertainty in Artificial Intelligence (UAI'86), pp. 47-67, 1986.

[151] Stølen, K. and Solheim, I., "Technology Research Explained", SINTEF ICT, Tech.rep. A313, 2006.

[152] Stålhane, T. and Wedde, K. J., "Practical experience with the application of HazOp to a software intensive system", in Proc. Joint ESCOM and ENCRESS Conference, pp. 271-281, 1998.

[153] Swiderski, F. and Snyder, W., *Threat Modeling*: Microsoft Press, 2004.

[154] Tamassia, R., Di Battista, G., and Batini, C., "Automatic Graph Drawing and Readability of Diagrams." *IEEE Transactions on Systems, Man, and Cybernetics.*, vol. 18 (1), pp. 61-79, 1988.

[155] Tan, K. C., Effects of Stimulus Class on Short -Term Memory Workload in Complex Information Displays, in *Department of Industrial Engineering and Operations Research*: Virginia Technical University, 1990.

[156] Treisman, A. and Gormican, S., "Feature analysis in early vision: Evidence from search asymmetries", *Psychological Review*, vol. 95 (1), pp. 15-48, 1988.

[157] Vetterling, M., Wimmel, G., and Wisspeintner, A., "Secure Systems Development Based on the Common Criteria: The PaIME Project", in Proc. 10th ACM SIGSOFT symposium on Foundations of software engineering (SIGSOFT'02/FSE-10), pp. 129-138, 2002.

[158] Vorster, A. and Labuschagne, L., "A Framework for Comparing Different Information Security Risk Analysis Methodologies", in Proc. SAICSIT'05, ACM International Conference Proceeding Series, (150), pp. 95-103, 2005.

[159] Walpole, R. E., Myers, R. H., and Myers, S. L., *Probability and Statistics for Engineers and Scientists*, 6th ed: Prentice Hall International, 1998.

[160] Ware, C., *Information Visualization: Perception for Design*, 2nd ed: Elsevier, 2004.

[161] Wertheimer, M., *Laws of Organization in Peceptual Forms [English Translation of: "Untersuchungen zur Lehre von der Gestalt", II, Psychologische Forschung 4 (1923), pp. 301 –350.] In Willis D. Ellis (ed.): A Source Book of Gestalt Psychology*: Routledge & Kegan Paul, 1923.

[162] West, S. and Andrews, A. D., "OCTAVE-Best Practices Comparative Analysis", Prepared for U.S. Army Medical Research and Materiel Command, ATI IPT Technical Report 03-4, 2003.

[163] Wickens, C. D., *Engineering Psychology and Human Performance*, 2 ed: HarperCollins, 1992.

[164] Winn, W., "An Account of How Readers Search for Information in Diagrams", *Contemporary Educational Psychology*, vol. 18, pp. 162-185, 1993.

[165] Winn, W., "Perceptual strategies used with flow diagrams having normal and unanticipated formats", *Perceptual and Motor Skills*, vol. 57, pp. 751-762, 1983.

[166] Winn, W., "The role of diagrammatic representation in learning sequences, identification, and classification as a function of verbal and spatial ability", *Journal of Research in Science Teaching*, vol. 19, pp. 79-89, 1982.

[167] Winn, W. and Solomon, C., "The effect of the rhetorical structure of diagrams on the interpretation of simple sentences", *University of Washington, unpublished manuscript*, 1991.

[168] Winograd, T. and Flores, F., *Understanding Computers and Cognition*: Addison-Wesley Professional, 1987.

[169] Xu, D. and Nygard, K., "A Threat-Driven Approach to Modeling aand Verifying Secure Software", in Proc. Int. Conference on Automated Software Engineering (ASE'05), pp. 342-346, 2005.

Table of Figures

FIGURE 1 – READING GUIDE 8
FIGURE 2 – RISKS AND THREAT SCENARIOS 10
FIGURE 3 – RESULTS FROM THE CORAS PROJECT 17
FIGURE 4 – THE OCTAVE PROCESS 17
FIGURE 5 – THE CRAMM PROCESS 18
FIGURE 6 – MICROSOFT'S SECURITY RISK MANAGEMENT PROCESS 19
FIGURE 7 – RISK MANAGEMENT PROCESS OVERVIEW 21
FIGURE 8 – FAULT TREE EXAMPLE 26
FIGURE 9 – EVENT TREE EXAMPLE 27
FIGURE 10 – "CAUSE CONSEQUENCE" DIAGRAM 28
FIGURE 11 – ATTACK TREE EXAMPLE 29
FIGURE 12 – OCTAVE THREAT TREE 30
FIGURE 13 – EXAMPLE OF A BN FROM [129] 31
FIGURE 14 – MARKOV MODEL 31
FIGURE 15 – BLOCK DIAGRAM EXAMPLE 32
FIGURE 16 – VALUE DEFINITIONS DIAGRAM IN THE UML PROFILE 34
FIGURE 17 – ASSET SPECIFICATION DIAGRAM IN THE UML PROFILE 35
FIGURE 18 – THREAT DIAGRAM IN THE UML PROFILE 35
FIGURE 19 – UNWANTED INCIDENT DIAGRAM IN THE UML PROFILE 35
FIGURE 20 – RISK DIAGRAM IN THE UML PROFILE 36
FIGURE 21 – RISK THEME DIAGRAM IN THE UML PROFILE 36
FIGURE 22 – TREATMENT DIAGRAM IN THE UML PROFILE 37
FIGURE 23 – TREATMENT EFFECT DIAGRAM IN THE UML PROFILE 37
FIGURE 24 – MISUSE CASE EXAMPLE 39
FIGURE 25 – ASSET-THREAT-SAFEGUARD DIAGRAM 40
FIGURE 26 – EXAMPLE OF A RISKIT GRAPH 41
FIGURE 27 – TYPICAL THREAT SCENARIO SITUATION 42
FIGURE 28 – NODE-LINK CONFIGURATIONS 43
FIGURE 29 – THE AND-GATE SYMBOL 43
FIGURE 30 – COLOR 45
FIGURE 31 – SIZE 45
FIGURE 32 – SHAPES 45
FIGURE 33 – EVALUATION METHODS [118] 49
FIGURE 34 – OUR RESEARCH METHOD 49
FIGURE 35 – METHOD USED FOR DEVELOPING THE FOUNDATION 50
FIGURE 36 – METHOD USED FOR DEVELOPING THE SECURITY RISK MODELING LANGUAGE 52
FIGURE 37 – ICONS USED IN THE ICON INVESTIGATION 54
FIGURE 38 – INSTRUCTIONS FOR FILLING OUT THE MODELING PREFERENCE TASKS 55
FIGURE 39 – METHOD USED FOR DEVELOPING THE GUIDELINE 56
FIGURE 40 – METHOD USED FOR DEVELOPING THE QUALITY FRAMEWORK 57
FIGURE 41 – THE ORIGINAL CONCEPTUAL MODEL OF CORAS 59
FIGURE 42 – THE FIRST REVISION OF THE CONCEPTUAL FOUNDATION 61
FIGURE 43 – THE FINAL REVISION OF THE CONCEPTUAL FOUNDATION 62
FIGURE 44 – STANDARD UML ICONS AND THE UML PROFILE SYMBOLS 63
FIGURE 45 – RELATIONSHIPS BETWEEN SYSTEM, MODEL AND DIAGRAMS 65
FIGURE 46 – RISK VALUE MATRIX 68
FIGURE 47 – RISKS PLACED IN THE RISK EVALUATION MATRIX 72
FIGURE 48 – THE QUALITY FRAMEWORK 75
FIGURE 49 – THE CONCEPTUAL FOUNDATION 93
FIGURE 50 – THE GRAPHICAL SYNTAX 94
FIGURE 51 – THE GRAPHICAL SYNTAX OF ASSET DIAGRAMS 95
FIGURE 52 – PROTECT RELATION 96
FIGURE 53 – INDIRECT HARM RELATION 96
FIGURE 54 – EXAMPLE OF AN ASSET DIAGRAM 97
FIGURE 55 – ASSET DIAGRAM: INDIRECT AND DIRECT ASSETS 99
FIGURE 56 – ASSET DIAGRAM: RISK ACCEPTANCE VALUES 99
FIGURE 57 – ASSET DIAGRAM: COMPLETE WITH ALL DETAILS 100

FIGURE 58 – THREAT DIAGRAM: DIRECT ASSETS....102
FIGURE 59 – THREAT DIAGRAM: INDIRECT ASSETS....102
FIGURE 60 – THREAT DIAGRAM (DIRECT ASSETS): LIKELIHOOD AND CONSEQUENCE ESTIMATION....104
FIGURE 61 – THREAT DIAGRAM (ONLY FOR THE INDIRECT ASSET): LIKELIHOOD AND CONSEQUENCE ESTIMATION....105
FIGURE 62 – RISK DIAGRAMS....106
FIGURE 63 – RISK EVALUATION DIAGRAMS....107
FIGURE 64 – TREATMENT DIAGRAM FOR THE DIRECT ASSET "GDI1.DATA PRIVACY"....108
FIGURE 65 – TREATMENT DIAGRAM FOR THE INDIRECT ASSET "CM1.COMPANY BRAND & REPUTATION"....109
FIGURE 66 – TREATMENT OVERVIEW: DIRECT ASSET....109
FIGURE 67 – TREATMENT OVERVIEW: INDIRECT ASSET....110
FIGURE 68 – EXAMPLE OF THE USE OF AND-GATE IN THE THREAT DIAGRAM....114
FIGURE 69 – EXAMPLE OF AN ASSET DIAGRAM WITH PARTICULAR FOCUS ON VULNERABILITIES....115
FIGURE 70 – RISK EVALUATION DIAGRAM ILLUSTRATING WHETHER THE RISKS ARE ACCEPTABLE OR NOT....115

List of Tables

TABLE 1 – SET-UP FOR TERMINOLOGY INVESTIGATION 1 50
TABLE 2 – SET-UP FOR TERMINOLOGY INVESTIGATION 2 51
TABLE 3 – SET-UP FOR THE ICON INVESTIGATION 53
TABLE 4 – SET-UP FOR THE MODELING PREFERENCES INVESTIGATION 54
TABLE 5 – RISK EVALUATION CRITERIA 68
TABLE 6 – RISKS WITH CONSEQUENCE AND LIKELIHOOD ESTIMATES 71
TABLE 7 – RISKS WITH RISK VALUES 72
TABLE 8 – TREATMENT EFFECTS 73
TABLE 9 – TREATMENT EFFECTS ON RISK VALUES 73
TABLE 10 – SEQUAL ASPECTS 75
TABLE 11 – MODELING TASK DEFINITION IN A SECURITY ANALYSIS SETTING 76
TABLE 12 – DOMAIN APPROPRIATENESS: MODELING PERSPECTIVES 77
TABLE 13 – DOMAIN APPROPRIATENESS: CONCEPTS 77
TABLE 14 – DOMAIN APPROPRIATENESS: SYMBOLS 80
TABLE 15 – PARTICIPANT LANGUAGE KNOWLEDGE APPROPRIATENESS 81
TABLE 16 – KNOWLEDGE EXTERNALIZABILITY APPROPRIATENESS 82
TABLE 17 – COMPREHENSIBILITY APPROPRIATENESS: INTERNAL REPRESENTATION 82
TABLE 18 – COMPREHENSION APPROPRIATENESS: EXTERNAL REPRESENTATION 84
TABLE 19 – TECHNICAL ACTOR INTERPRETATION APPROPRIATENESS 86
TABLE 20 – ORGANIZATIONAL APPROPRIATENESS 87
TABLE 21 – EVALUATION SCORES 90
TABLE 22 – NAMING CONVENTIONS 95
TABLE 23 – STEPS WITHIN THE DIFFERENT PHASES IN THE CORAS APPROACH 98
TABLE 24 – EXAMPLE OF HOW ONE MAY COMBINE LIKELIHOOD ESTIMATES 104
TABLE 25 – RECENT FIELD TRIALS 115
TABLE 26 – EVALUATION SCORES 128

Appendix A

Empirical Investigations of the CORAS Language for Structured Brainstorming

SINTEF

SINTEF ICT

Address: P.O.Box 124, Blindern
0314 Oslo NORWAY
Location: Forskningsveien 1
0373 Oslo
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

Empirical Investigations of the CORAS Language for Structured Brainstorming

AUTHOR(S)

Ida Hogganvik, Ketil Stølen

CLIENT(S)

Norges Forskningsråd (NFR) / The Research Council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
STF90 A05041	Unrestricted	NFR152839/220	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
A	82-14-03657-7	40332800	41/2
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.)	CHECKED BY (NAME, SIGN.)
		Ketil Stølen	Fredrik Seehusen
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.)	
	2005-01-30	Bjørn Skjellaug, Research director	

ABSTRACT

Methods for security risk analysis are often based on structured brainstorming (e.g. what [20] calls HazOp). A structured brainstorming gathers a group of different system experts and the idea is that they will find more risks as a team than one-by-one.
The CORAS modeling language [19] has been designed to support the brainstorming and to document incident scenarios identified during these structured brainstorming sessions. The language is graphical, based upon the unified modeling language (UML) [21], and is recommended by OMG [9].
This paper reports the results from two experiments with the CORAS language and the lessons learned from these. The first experiment aimed to test the understanding of the language's internal conceptual model, while the second tested the graphical representation. The findings show that the interpretation of the conceptual model is a possible source for misunderstandings in a security analysis and we point out the ways to avoid this. The graphical icons in the CORAS language make model "navigation" faster, but the models are not necessarily understood more correctly than those without graphical icons.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Security risk analysis	Sikkerhetsanalyse
GROUP 2	Modeling language	Modelleringsspråk
SELECTED BY AUTHOR	Empirical experiments	Empiriske eksperimenter
	Usability	Brukskvalitet

TABLE OF CONTENTS

1 INTRODUCTION 5

2 STRUCTURED BRAINSTORMING AND THE CORAS LANGUAGE 7

2.1 STRUCTURED BRAINSTORMING 7
2.2 THE UNDERLYING CONCEPTUAL MODEL FOR THE CORAS LANGUAGE 7
2.3 STEREOTYPING WITH GRAPHICAL ICONS IN THE CORAS LANGUAGE 8

3 DESCRIPTION OF THE TWO EXPERIMENTS 11

3.1 MATERIAL FOR THE TERMINOLOGY EXPERIMENT 11
3.2 MATERIAL FOR THE ICON EXPERIMENT 12
3.3 EXECUTION AND PRACTICAL CONSIDERATIONS 13
3.4 ANALYSIS MODELS 13

4 RESULTS FROM THE TWO EXPERIMENTS 15

4.1 DESCRIPTIVE STATISTICS FROM THE TERMINOLOGY EXPERIMENT 15
4.2 DESCRIPTIVE STATISTICS FOR THE ICON EXPERIMENT 16
4.3 STATISTICAL TESTS FOR THE ICON EXPERIMENT 17

5 DISCUSSION 19

5.1 TERMINOLOGY EXPERIMENT 19
5.2 ICON EXPERIMENT 21
5.3 THREATS TO VALIDITY FOR THE EXPERIMENTS 22
5.3.1 The terminology experiment 23
5.3.2 The icon experiment 23
5.3.3 General 23

6 LESSONS LEARNED 25

7 CONCLUSIONS AND RECOMMENDATIONS 27

FUTURE WORK 27
ACKNOWLEDGEMENTS 28

REFERENCES 29

APPENDIX A – ICON EXPERIMENT MATERIAL 31

A.1 MATERIAL WITH CORAS ICONS 31
Part 1 31
Part 2 32
Part 3 33
A.2 MATERIAL WITHOUT CORAS ICONS 35
Part 1 35
Part 2 36
Part 3 37

APPENDIX B SUGGESTED SOLUTIONS FOR THE ICON EXPERIMENT 40

Part 1 40
Part 2 40
Part 3 40

LIST OF FIGURES

Figure 1 Components in the CORAS language 5
Figure 2 The CORAS conceptual model of security analysis terms 7
Figure 3 Stereotyping with CORAS icons 9
Figure 4 Stereotyping without CORAS icons 9
Figure 5 Risk scenario for a smartcard 9
Figure 6 Relations covered by the questionnaire 12
Figure 7 Histogram of data from terminology experiment 15
Figure 8 Correct answers 16
Figure 9 Well understood relations 19
Figure 10 Misunderstood relations 20
Figure 11 Relations creating uncertainty (answer="I don't know") 21
Figure 12 Class diagram with and without icons 22
Figure 13 The revised conceptual model 26
Figure 14 Smartcard 31
Figure 15 Webservices 32
Figure 16 Webservices 33
Figure 17 Smartcard 34
Figure 18 Smartcard 35
Figure 19 Webservices 36
Figure 20 Webservices 37
Figure 21 Smartcard 39

LIST OF TABLES

Table 1 Terminology questionnaire 11
Table 2 Experiment material 13
Table 3 Descriptive statistics of the statements 15
Table 4 Descriptive statistics from icon experiment 17
Table 5 Mann-Whitney test 17

1 Introduction

The participants in a structured brainstorming are system experts with different expertise. The idea is that the experts see the system from different viewpoints and therefore in collaboration will identify more risks than through individual expert judgments. The participants may have little or no previous experience with security risk analysis and therefore they need a common, intuitive way of documenting the security risk scenarios they identify. In the following we call security risk analysis just *security analysis*. The CORAS language [19] is a graphical modeling language that has been developed to support brainstorming and help document the results. The CORAS language was delivered within the EU project "CORAS" [3, 10, 12, 14-16] which gathered well known risk analysis techniques into one common methodology for security analysis. The CORAS language is built on UML [21] as a UML profile and is recommended by the international standardization consortium OMG [9].

The aim of this study is to investigate the usability of the CORAS language. We will look at both the conceptual model, on which the abstract syntax is based, and the external graphical representation.

The underlying conceptual model incorporates terminology from several relevant standards [1, 2, 4-8]. The abstract syntax of the CORAS language is based on this conceptual model for security analysis. Clearly, this model must be intuitive for the CORAS language to be successful during brainstorming sessions.

The external representation of the CORAS language is recognized by its special graphical icons that symbolizes the different terms in the conceptual model. The icons are believed to make the models easier to read and understand. Often graphical icons are seen as merely "decoration" just to make the models look nicer, but we believe that for people that are unfamiliar with system modeling, these icons help understanding the models.

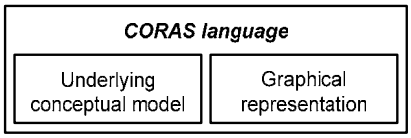


Figure 1 Components in the CORAS language

We report the results from two student experiments that have been conducted to investigate the issues discussed above. The first tested the understanding of the underlying conceptual model and showed that the interpretation of security analysis terms is influenced by how they are commonly used in daily language. The second experiment tested the use of graphical icons and the effect on the understandability of models. The icons were found to increase the speed of model navigation (i.e. identification of different elements), but not the correctness of model interpretation.

The paper is structured as follows: Section 2 provides an introduction to structured brainstorming and the CORAS language. Section 3 describes the design of the two experiments and Section 4 reports the results from statistical analyses. In Section 5 we

discuss these results and possible interpretations, Section 6 reports lessons learned and Section 7 concludes this report.

2 Structured brainstorming and the CORAS language

A structured brainstorming is a methodical assessment of a system where people with different roles and competence participate. Through discussion they identify risks and suggest treatments under the guidance of an analysis leader. The participants in a structured brainstorming need to have a clear understanding of the terminology the techniques is based on. Terms like frequency, probability, vulnerability and asset are commonly used, but the individual interpretation may differ depending on the background and experience of the person in question. Several of the concepts are commonly used in everyday language, but not with one unique context independent interpretation.

2.1 Structured brainstorming

The participants in a structured brainstorming are potentially anyone with an interest in the target system. The participants may be customer support personnel, system administrator, system developer, company security responsible or other roles. They receive no special training in advance, only an introduction in the beginning of the brainstorming session. We believe that if an analyst is not clear in what terms to use, and how to explain them, participants may be left confused and will not contribute optimally to the analysis. With this assumption in mind, we wanted to explore how people with software background and nearly no experience with security analysis interpret the most common concepts and relationships in the CORAS conceptual model.

2.2 The underlying conceptual model for the CORAS language

The most important terms of the CORAS conceptual model are: risk, unwanted incident, threat, vulnerability, treatment, consequence, frequency, asset, probability and stakeholder (**Figure 2**). The associations between the elements have cardinalities that say how many instances of one element can be related to one instance of the other. Example: "a stakeholder has at least one and maximum infinite assets; and an asset belongs to only one stakeholder".

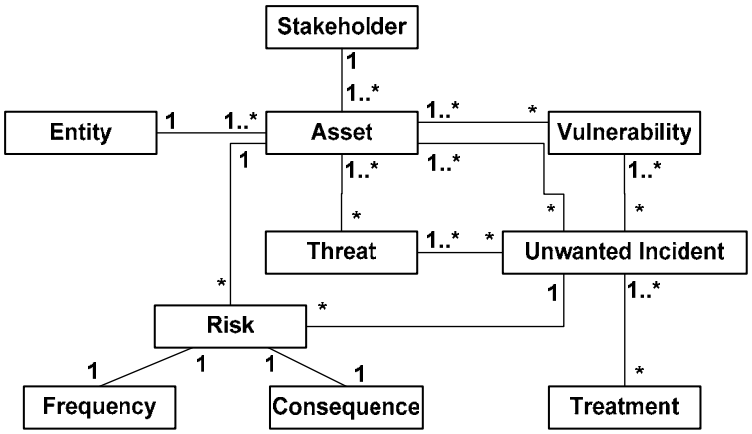


Figure 2 The CORAS conceptual model of security analysis terms

Originally **Figure 2** was not designed to be "stand-alone" model with a unique association line for every possible relation between the risk analysis terms. The

understanding of the figure also relies on a textual description.We explain **Figure 2** as follows: the system or part of a system, assessed during a security analysis is called the **target of evaluation**. Everyone with interests in the target is **stakeholders** of the system. System users, system maintainers and system developers are typical stakeholders. Different stakeholders often value the system differently; a system user who is dependent on the system will put a high value on it, while other stakeholders might not value the system equally high. The same **entity** may be assigned different values by different stakeholders. We refer to these entities with their values as assets. An **asset** is something to which a stakeholder directly assigns value and, hence, for which the stakeholder requires protection. An asset is therefore uniquely linked to a single stakeholder. A stakeholder wants to protect his/her assets from loosing value. Examples of assets are customer information, source code, company routines, critical system services etc. Target system stakeholders and their assets are normally identified early in the security analysis process. Figure 1 includes four important security analysis concepts related to asset: vulnerability, unwanted incident, threat and risk. A **vulnerability** is a weakness or lack, making an asset vulnerable to harmful actions. One may understand a vulnerability as something that is missing, e.g. if a company network lacks a firewall then this may be a vulnerability with respect to some assets in the network. An **unwanted incident** is an event that may harm the asset and something we want to prevent. An unwanted incident is the result of a threat exploiting a vulnerability. If the company network is an asset, then an unwanted incident is unauthorized access to the network by intruders. A **threat** is someone or something that wants to destroy, remove or interfere with the asset and a risk is the chance of this happening. With respect to the already mentioned company network a threat may be a person who knows or discovers the vulnerability and wants to exploit it. First the company does not recognize the situation as a potential risk because nobody outside the company is aware of the security hole, but when an employee is fired, they suddenly realize that there is a risk for unauthorized network access by people familiar with the company infrastructure. The risk is characterized by a **risk value** (e.g. low, medium, high or other scales) which is based upon the estimated **frequency** for it to happen and its **consequence** in loss of asset value. If a risk is estimated to occur two times a year and the consequence is a loss of 200000 dollars each time, the risk value could be “high” which means the risk should be treated. The **treatment** is applied either to the unwanted incident, the threat or the asset’s vulnerability and the desired effect is reduced frequency and/or consequence, i.e. a reduced risk value.

2.3 Stereotyping with graphical icons in the CORAS language

A security risk scenario will typical illustrate how unwanted incidents relate to assets and treatments, it illustrates the threats and how they through various threat scenarios can harm the assets. A security risk scenario shows associations and relationships that are infeasible to describe with text only. To make the scenarios easier to read, the concepts in the CORAS language are stereotyped with special icons and text-labels as illustrated in **Figure 3**. Stereotyping is a technique in UML that lets you mark normal UML elements with “labels” to emphasize that they are of a certain type, for more about this we refer to [11, 21]. Our focus is mainly on the graphical icon stereotyping and less on the stereotype name. We are not aware of other experiments with UML stereotyping, focusing on understanding, than [18]. The

results from this experiment showed that icon stereotyping makes the models easier and faster to understand. The treatment, unwanted incident and the threat scenario are all special cases of a UML use case, and the asset and threat (here: "attacker") are UML actors. In our setting this implies that a risk scenario without icons will use normal UML symbols for use case and actor, illustrated in **Figure 4**.

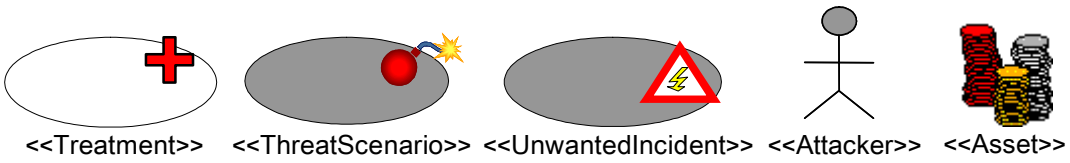


Figure 3 Stereotyping with CORAS icons

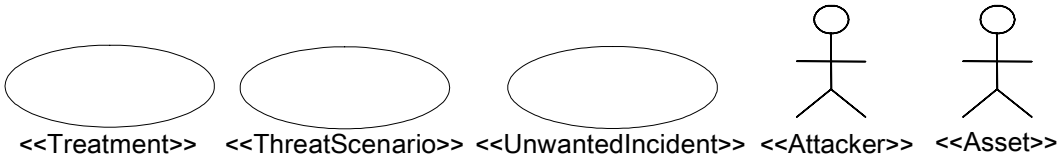


Figure 4 Stereotyping without CORAS icons

Figure 5 illustrates one of the risk scenarios that were used as material in our icon experiment. The model illustrates three assets (software token, hardware token and service availability) associated with two unwanted incidents (token stolen and token missed) and two ways this can happen (the two threat scenarios). One of the threat scenarios is initiated by an attacker (malicious person). Two treatments are identified, "educate user" will reduce the likelihood for token missed and "revoke token" will reduce the consequence of both token missed and token stolen.

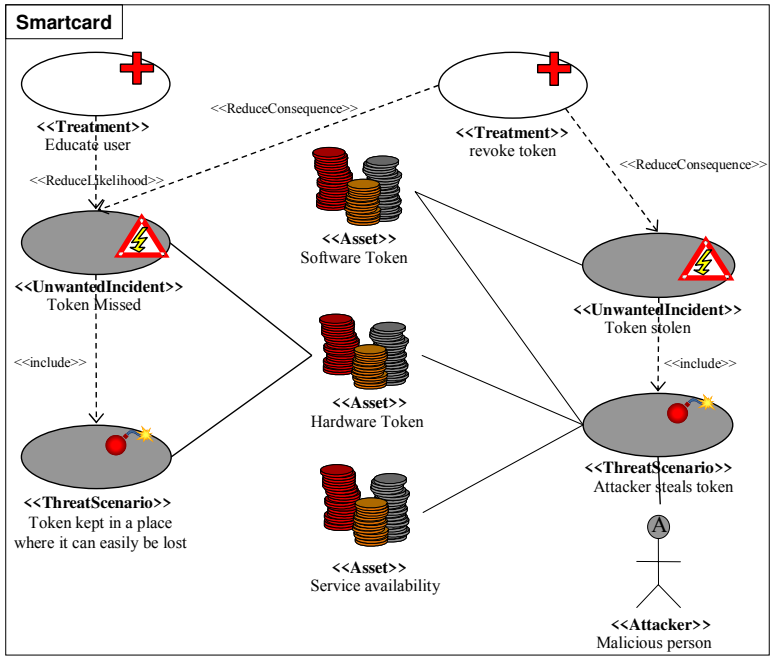


Figure 5 Risk scenario for a smartcard

3 Description of the two experiments

The experiments were conducted during the lectures in a university course at the institute for informatics, University of Oslo. The course has focus on a theoretic foundation for UML, refinement concepts in a UML context, modularity through contract-oriented specifications and model-based security analysis. The purposes of the experiments were:

- Terminology experiment: to explore the comprehension of risk terms taken from the underlying conceptual model of the CORAS language.
- Icon experiment: to investigate the effect of the graphical stereotyping used by the CORAS language.

3.1 Material for the terminology experiment

The terminology experiment material was a questionnaire with 20 short statements. Each covered the relationship between 2-4 security analysis concepts that was either correct or incorrect. In addition to judging the statement, the subjects were asked to give a short explanation for their answer. The first section of the questionnaire asked some demographic questions about the subject's background and personal perception of the topic that should help us interpret the results from the analysis. We asked if they:

- had attended the previous lesson about security analysis (attendance is voluntarily)
- feel the concepts are difficult to understand
- believe security analysis is useful
- think they understand the concepts well

To ensure that the questionnaire was completed on time, the subjects had to "tic off" an answer alternative for all the statements before writing their explanations. The questionnaire is presented in **Table 1** and had three different answer alternatives: "correct", "wrong" and "I don't know" (the original is in Norwegian):

Table 1 Terminology questionnaire

1. An unwanted incident can initiate new unwanted incidents by exploiting new vulnerabilities (correct).
2. A treatment will eliminate at least one unwanted incident (wrong).
3. The purpose of a treatment is to reduce risk, not necessary reduce vulnerability (correct).
4. An unwanted incident consists of consequence, frequency and a risk (wrong).
5. A treatment is always directed towards a vulnerability (wrong).
6. A threat is something that can initiate an unwanted incident (correct).
7. A risk is related to an asset's vulnerability (correct).
8. A threat can create a risk (correct).
9. There has to exist a threat before one think of an unwanted incident as a risk (correct).
10. A risk is something that will reduce the value of an asset (correct).
11. A vulnerability is the same as a risk (wrong).
12. A threat is something that has a potential to reduce the value of an asset (correct).
13. An unwanted incident cannot consist of multiple unwanted incidents (wrong).
14. An entity without value can also be considered as an asset (wrong).
15. A treatment can be directed towards a threat (correct).
16. A stakeholder is someone that wants to protect his or her assets (correct).
17. A threat can exploit a vulnerability (correct).

18. A successful treatment must reduce both consequence and frequency (wrong).
19. A risk and an unwanted incident is the same (wrong).
20. *One has to know both probability and frequency to calculate the risk value (wrong).*

A challenge in designing the questionnaire was which concepts to include, and which relationships to test. We selected 10 of the most common security analysis terms from CORAS' conceptual model in addition to "probability" which is commonly used to describe one kind of frequency. Some concepts are associated with more concepts than others, and this is reflected in the number of times they appear in different statements in the questionnaire (see also **Figure 8**). E.g. the term "risk" is included in more statements than "frequency" since it is associated with more concepts. The CORAS conceptual model (**Figure 2**) displays only the most common relations between its terms, but there are other possible relations which we aim to capture in our experiment illustrated in **Figure 6**.

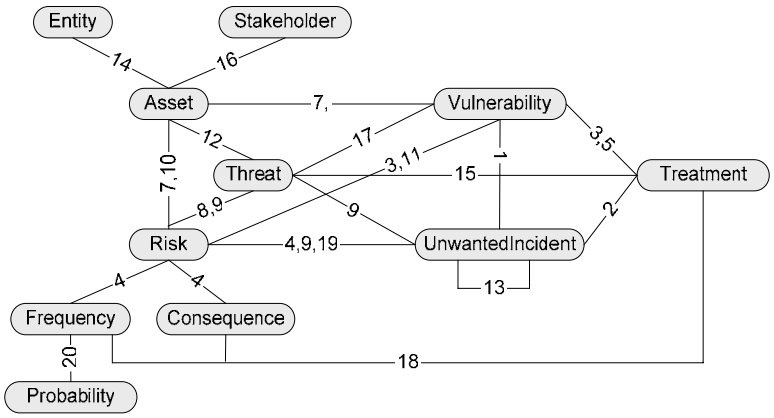


Figure 6 Relations covered by the questionnaire

3.2 Material for the icon experiment

The icon experiment material was a set of security risk scenarios modeled with the CORAS language (example in **Figure 5**) and a related questionnaire. The diagrams were taken from a report on security threat modeling project resulting from a research collaboration between Microsoft Research in Belgium and SINTEF ICT [13]. The questions focused on general understanding of notation making domain knowledge less important. Nevertheless, the models we used illustrated general security issues related to smartcard (credit card sized storage medium) and web services (software that makes itself available over the Internet using standardized XML) that could be known to some of the students.

Half of the subjects received models stereotyped with both CORAS icons and stereotype names while the other half had standard UML use case icons and stereotype names. The questions were the same for both groups. This experiment did not test stereotyping in general since we kept the stereotype name, but rather the graphical icon stereotyping.

The questionnaire was divided into three parts:

Table 2 Experiment material

	Max time	Type of task	# Models, # Questions	Max score	Variables measured
Part 1	3 min	Training task for model navigation	1 model, 7 questions	7p	Score, # questions answered
Part 2	1.5 min	Model navigation	1 model , 10 questions	10p	Score, # questions answered
Part 3	15 min	Model understanding	3 models, 11 (4+5+2) questions	11p	Score, # questions answered, time used pr. model

Part 1 and 2 had questions that tested whether the subjects were able to quickly identify specific elements like treatment, risk, threat agent etc in a model (model navigation). The questions were of the type "*How many assets are explicitly modeled in this diagram?"* Part 3 focused on how the models were understood and interpreted. An example of the type of question used here is "*Which unwanted incidents affect the asset "Web server" in this model?"* Since part 3 included three different sets of models with belonging questions, the students had to record the start time on each of the sets to make us able to analyze the time spent (in minutes) on each sets.

3.3 Execution and practical considerations

The master students attending the experiments had not been given information in advance since they were executed as exercises during the lessons. At this level of education it is natural to assume that the students know some UML.

For the terminology experiment, we faced the situation that the subjects' previous knowledge about security analysis could vary from 0-4 hours. We believed this could influence the result and therefore we asked about this in one of the demographic questions. A short introduction to the type of task was given and the students were given 20 minutes to fill in the questionnaire.

The icon experiment took place in the same setting two weeks later. The lesson before the experiment the students had a 45 min introduction to the CORAS language. Also in a real structured brainstorming the participants will be given an introduction to the notation in advance, but then considerably shorter. The questionnaires were handed out in a randomized manner, 13 subjects received questionnaires with CORAS icons and 12 received the one with standard UML use case icons.

3.4 Analysis models

The two experiments measured different variables, and had therefore different analysis models. In the terminology experiment we measured score for each statement looking for particular easy or difficult concept-relationships.

The icon experiment measured both score and time used per question /number of questions accomplished. The two groups of subjects were compared to see whether the one with CORAS icons had a higher score than the group without. In the same manner we analyzed whether one group managed to complete more questions than the other group or used less time pr. model-question-set. Since the sample set was expected to be relatively small and possible not normally distributed, we chose to use a one-tailed, non-parametric Mann-Whitney test with alpha level = 0.05.

4 Results from the two experiments

4.1 Descriptive statistics from the terminology experiment

We had 31 subjects in the experiment. The shaded numbers in the table indicates statements that obtained a high percentage of either “correct answers”, “incorrect answers” or “I don’t know answers”.

Table 3 Descriptive statistics of the statements

Statement (terms used)	Correct answers		Incorrect answers		"I don't know" answers	
	#	%	#	%	#	%
1 (unwanted incident, vulnerability)	23	74.2	3	9.7	5	16.1
2 (treatment, unwanted incident)	12	38.7	16	51.6	3	9.7
3 (treatment, risk, vulnerability)	20	64.5	9	29.0	2	6.5
4 (unwanted incident, consequence, frequency, risk)	9	29.0	19	61.3	3	9.7
5 (treatment, vulnerability)	14	45.2	9	29.0	8	25.8
6 (threat, unwanted incident)	25	80.6	3	9.7	3	9.7
7 (risk, asset, vulnerability)	23	74.2	4	12.9	4	12.9
8 (threat, risk)	19	61.3	8	25.8	4	12.9
9 (threat, unwanted incident, risk)	13	32.3	10	41.9	8	25.8
10 (risk, asset)	13	41.9	18	58.1	0	0.0
11 (vulnerability, risk)	25	80.6	4	12.9	2	6.5
12 (threat, asset)	25	80.6	6	19.4	0	0.0
13 (unwanted incident)	22	71.0	3	9.7	6	19.4
14 (entity, asset)	16	51.6	9	29.0	6	19.4
15 (treatment, threat)	19	61.3	8	25.8	4	12.9
16 (stakeholder, asset)	29	93.5	1	3.2	1	3.2
17 (threat, vulnerability)	28	90.3	0	0.0	3	9.7
18 (treatment, consequence, frequency)	16	51.6	11	35.5	4	12.9
19 (risk, unwanted incident)	23	74.2	4	12.9	4	12.9
20 (probability, frequency)	5	16.1	20	64.5	6	19.4

The distribution of data is also illustrated using histogram in **Figure 7**.

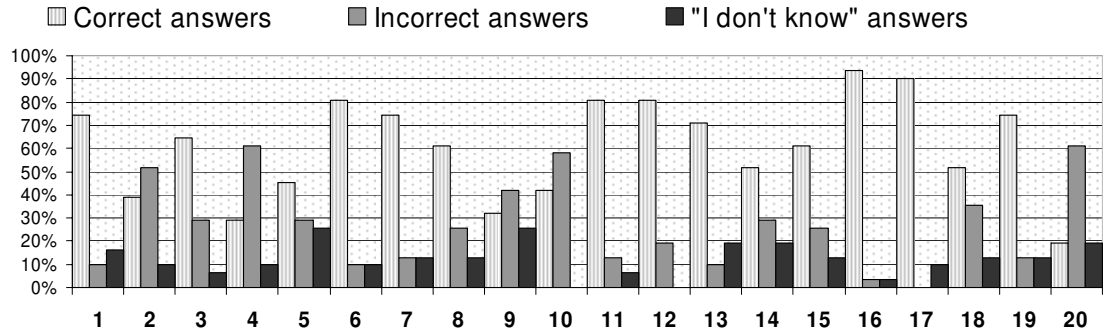


Figure 7 Histogram of data from terminology experiment

Table 2 shows how many of the subjects that had attended the previous lesson about security analysis and how they scored.

Table 2 Lesson attendance and score

Attended previous lesson about security analysis				
	#	Correct answers	Incorrect answers	"I don't know"
“yes"	18	60.8%	26.1%	13.1%
"no"	11	60.5%	28.6%	10.9%
n/a	2			

Not all concepts are used equally many times in the statements, **Figure 8** illustrates how often a concept is used, the most tested term is “risk” (40% of the statements contain this term) while the least tested term is “probability” (5% of the statements). Statements that contained the term “risk” received in average 57.3% correct answers, while for “probability” the result was 16.% correct answers. This shows that one have to take into consideration how often a terms is used in the questionnaire when comparing the level of correct answers.

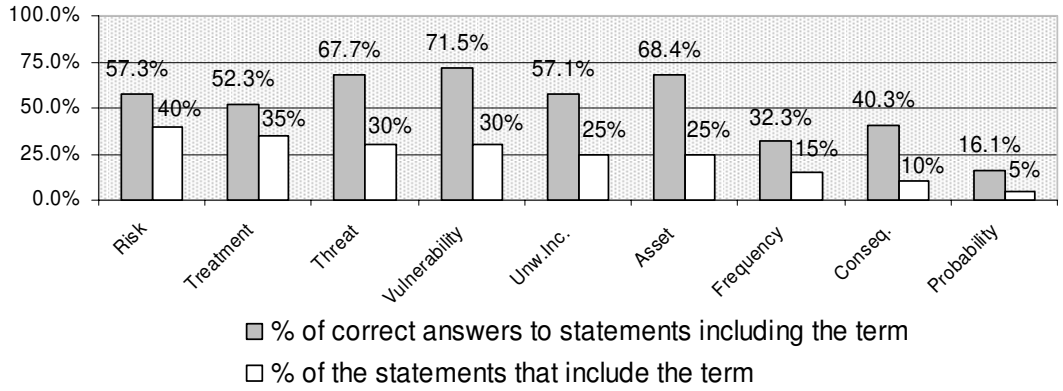


Figure 8 Usage of terms and correct answers

4.2 Descriptive statistics for the icon experiment

We had 25 subjects. 13 of them received questionnaires with CORAS icons, 12 got questionnaires with standard UML use case icons. **Table 4** contains descriptive statistics from the icon experiment. The results are grouped according to which part of the questionnaire they represent. **Score** denotes how many points the students scored, **# Answers** are the number of answers the students managed to complete within time and **Time used** is the minutes used pr set of questions in part 3. We report mean and standard deviation for each of the groups (with and without icons).

Table 4 Descriptive statistics from icon experiment

		CORAS icons		UML icons	
		mean	**st.dev**	**mean**	**st.dev**
Part 1	**Score**	4.5	1.38	4.77	1.09
Part 2	**Score**	2.88	0.91	2.58	0.95
	#Answers	4.75	1.54	3.69	1.18
Part 3	Score set 1	3.04	0.96	3.38	0.92
	Score set 2	3.62	0.68	3.7	1
	Score set 3	0.63	0.8	0.27	0.6
	Score all sets	7.29	1.41	7.36	1.82
	Time used set 1	5.58	1.44	6	1.53
	Time used set 2	5.3	1.34	6.7	1.25
	Time used set 3	2.71	0.95	-*	-*
Total	**Score**	14.67	2.11	14.7	2.51

*only one subject finished

4.3 Statistical tests for the icon experiment

The results from the Mann-Whitney test is presented in **Table 5**, the critical value in column five is taken from [22]. Both for #Answers (part 2) and Time used set 2 (part3) the results showed a significant difference between the two groups.

Table 5 Mann-Whitney test

		n1	**n2**	**critical value**	**u**
Part 1	Score	12	13	<=47	71.5
Part 2	Score	12	13	<=47	67.5
	# Answers	12	13	<=47	46
Part 3	Score set 1	12	13	<=47	62
	Score set 2	12	13	<=47	70.5
	Score set 3	12	13	<=47	56.5
	Score all sets	12	13	<=47	77.5
	Time used set 1	12	13	<=47	71
	Time used set 2	10	10	<=27	21.5
	Time used set 3	7	1	-	-
Total	Score	12	13	<=47	76

$\alpha = 0.05$

5 Discussion

This section discusses both experiments, first the terminology experiment, then the icon experiment.

5.1 Terminology experiment

The results from our terminology experiment show that most of the concepts and relationships from the CORAS' conceptual model are well understood. The easiest statements seem to be the ones which include terms used in the daily language, like *stakeholder*, *asset*, *threat* and *vulnerability*.

Figure 6 illustrates the relationships that we tested; and in the following we discuss those that obtained high percentages of correct (**Figure 9**), incorrect (**Figure 10**) or "I don't know" answers (**Figure 11**). In the remaining we refer to the statements using their number: #3 means statement 3 in the questionnaire (in the figures the "#" is not used).

The results show that terms that are quite common in everyday language, seem to be best understood and not so easily mixed up with other concepts (**Figure 9**). The terms that stand out are threat, vulnerability, asset and stakeholder. The single most intuitive relation is the asset-stakeholder relationship (#16). A large majority of the subjects (93.5%) believe "a stakeholder is someone that wants to protect his or her assets" is correct. 90.3% of the subjects agreed in "a threat can exploit a vulnerability" (#17). The relation between asset and threat (#12) obtains over 80% correct answers and over two third agree in "a risk is related to an asset's vulnerability" (#7).

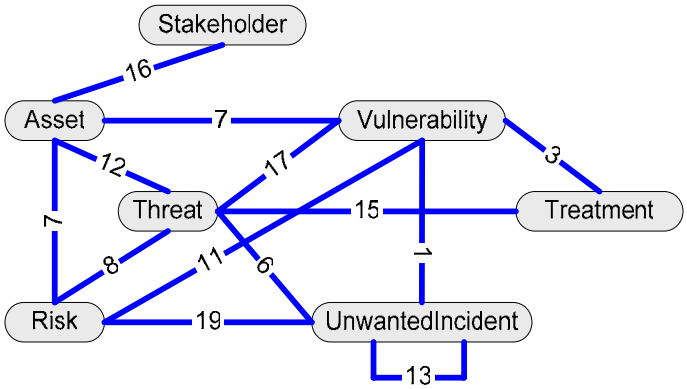


Figure 9 Well understood relations

Our impression is that the subjects find it easier to deal with concrete concepts, i.e. an unwanted incident is something real, than the more abstract term "risk". As an example the statement "a threat is something that can initiate an unwanted incident" (#6) obtained a high level of correct answers (80.6%), but in a similar statement (#8), where we substituted unwanted incident with risk, we received considerably less correct answers (61.3%).

The subjects seem to be unaware of the fact that the purpose of a treatment often is to *reduce* frequency or consequence. More than half of the subjects (51.6 %) believe a treatment always will *eliminate* at least one unwanted incident (#2, **Figure 10**).

In the CORAS conceptual model a risk is **always** associated with an unwanted incident, a frequency value, a consequence value and an asset which represent concrete scenarios, values or entities, but "risk" in itself is rather a composition of them. This means that risk in a way is an abstract concept, something that seems troublesome for the subjects. The terms "unwanted incident" and "risk" are often confused, 71 % of the subjects were wrong or uncertain on whether a risk includes an unwanted incident or vice versa (#4). **Figure 8** shows that statements that include the term *unwanted incident* in average only get 52.3% correct answers and this should be considered when revising the model.

In statement #10: "a risk is something that will reduce the value of an asset", more than half of the subjects disagreed (58.1%). They disagreed in the use of "will" in the statement, claiming that a risk with frequency near zero "*may* reduce...", not "*will* reduce...". We agree in their argumentation and will reformulate the statement in further work with the conceptual model.

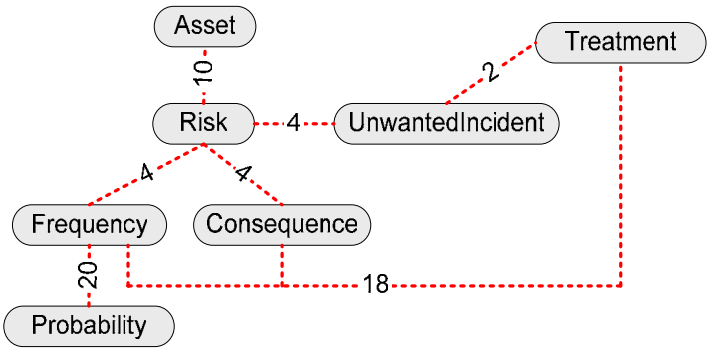


Figure 10 Misunderstood relations

The single most difficult statement in the questionnaire was "one has to know both probability and frequency to calculate the risk value" (#20). A risk value is a function of a consequence and a frequency measure, and some may have answered "I don't know" because they did not know what a risk value is. On the other hand it should be possible to answer this statement just by knowing that probability and frequency are two measures for the same thing and therefore knowing only one of these is sufficient to calculate risk value. A large group (83.9%) either gave incorrect answer or did not know the answer on this statement. Neither frequency nor probability is particularly common in the everyday language and this is probably why we get a high percentage of incorrect or uncertain answers. It is possible that just by avoiding these exact terms and rather ask "how often?" or "how many out of 100?" is enough to reduce the uncertainty among the participants in a structured brainstorming.

More than half of the subjects (54.8%) either think a treatment will always be directed towards a vulnerability or they are uncertain on the answer (#5). The subjects had not understood that a treatment can also be applied to an unwanted incident or a threat.

According to CORAS' conceptual model an unwanted incident is always associated with one or more threats, but a threat is not necessary associated with an unwanted incident. The statement "There has to exist a threat before one think of an unwanted incident as a risk" (#9) confused the subjects. A majority either disagreed (41.9%) with this interpretation or were uncertain (25.8%), and they did not accept this as an intuitive explanation for the relation between threat, unwanted incident and risk. We believe that some of this confusion arises from a misunderstanding that an unwanted incident in itself is a threat (comments made by the subjects in the questionnaire).

It was unclear how an entity relates to an asset (#14) and the subjects were inconsistent in their answers. The reason may be that "entity" is a term that almost does not exist in the daily language and we think this created more uncertainty than the actual relationship to asset.

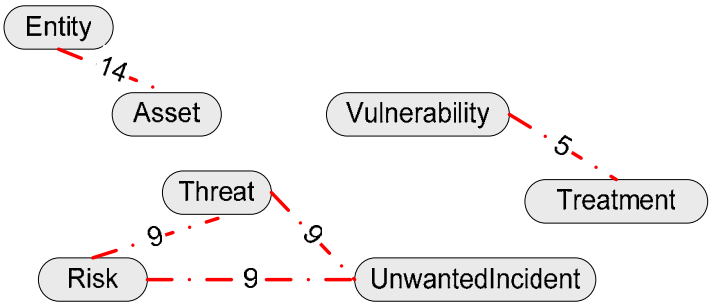


Figure 11 Relations creating uncertainty (answer="I don't know")

There were not identified correlations between data from the initial demographic questions and score. Surprisingly enough, attendance to the previous lesson did not affect the overall score on the questionnaire. This indicates that many security analysis concepts are already known to people, but often with another interpretation.

5.2 Icon experiment

What we can see from the icon experiment is that stereotyping with CORAS icons does not affect the "correctness" of the answers, but the subjects managed to answer more questions than the group without CORAS icons. Both model sets kept the stereotype name; otherwise it would be impossible to distinguish the UML icons. If all the subjects had read the text under the icons there had not been any difference between the two groups, but looking at the results it is possible to assume that the stereotype name is ignored. We did ten statistical tests of the data from the icon experiment (not including the test on *time used for set 3* since the number of subjects was too small). For two of the tests we obtained a statistical significant difference between the two groups of subjects at an alpha level of 0.05.

Part 1 and 2 had questions that aimed to investigate whether the subjects were able to differentiate between the elements in the model. They were asked to count the number of different security analysis concepts, some were in fact represented in the models and some were not. Part 1 was a training task with no time pressure; therefore the differences between the groups are insignificant both with respect to correctness and number of answers. Part 2 was similar to part 1 but with a much less time available

and here we obtained a significant difference in the number of questions answered between the CORAS icon subjects and the other group. The subjects with CORAS icons manage to answer in average 4.75 questions, while the result for the group without CORAS icons is 3.69. There was insignificant difference with respect to the correctness of the answers, the group with CORAS icons scored in average 2.88 points while the others scored 2.58 points.

Part 3 of the questionnaire had a different style than the previous parts. The section focused more on interpretation of models, and less on navigation. We used three models with belonging sets of questions (Set 1, Set 2 and Set 3). All sets were measured for score and time used pr set. The results show that there was no significant difference between the groups in Set 1. We see this set as a training set since the type of questions was different from the previous ones and therefore both groups used some time to get used to the questions. For Set 2 we obtained a significant difference with respect to the time used. The group with CORAS icons used less time to complete the questions (average: 5.3 min) than the group using standard UML use case icons (average: 6.7 min). But also here there were no difference in score between the two groups. It was not possible to statistically analyze the difference between the two groups in Set 3 since only eight finished it. 7 from the CORAS icon group finished in contrast to only one using UML icons, which can indicate that the CORAS icon group performed better also in this set.

[18] reports the results from a similar experiment investigating the understanding of stereotyped UML class diagrams. The experiment tested icon-stereotyping vs. normal class stereotyping (roughly sketched in **Figure 12)**. They substituted traditional stereotyped classes (Alt.1) with special stereotype icons (Alt.2) and found this to improve both the speed of navigation and the correctness of model interpretation.

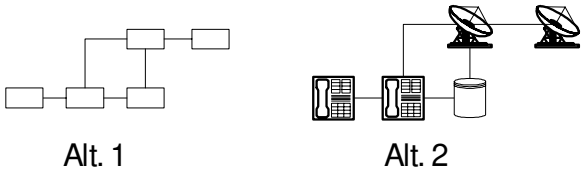


Figure 12 Class diagram with and without icons

According to their findings we should expect to find an improvement in both areas but we only found the difference in the navigation speed. A possible reason is that many of the security risk concepts do not have obvious graphical representations and that our icons may be ambiguous. Also the interpretation of a model may depend on other factors than icons like naming of elements, modeling style and familiarity with the system or domain that is modeled. At this point our results support the belief that icons affect how fast the subjects navigate through the model to identify different elements, but less on how they interpret the model.

5.3 Threats to validity for the experiments

Experiments will always have some threats with respect to the validity of its results. In the following we explain the ones we have identified.

5.3.1 The terminology experiment

- The questionnaire design: developing a good questionnaire using natural language is difficult. There is always possible to write text that is interpreted different from its intention. Even though our questionnaire was both pilot tested and reviewed, we found some weaknesses after the experiment. We believe statement 11 "a vulnerability is the same as a risk" failed to express what we really intended to check. We wanted to investigate whether a vulnerability automatically is considered as a risk, or do the subjects recognize the need for identification of unwanted incidents also. Unfortunately it was probably quite easy to guess that the two terms they were not identical and mark "Incorrect". The same goes also for statement 19 " a risk and an unwanted incident is the same" where the intention was to investigate whether the subjects were aware of that unwanted incident is a part of a risk, not if the subjects see the difference between two words.
- Data interpretation: since one of the alternative answers for each statement was "I don't know", we choose to interpret missing answers as "I don't know" This gave us some data points that are not directly filled in by subjects, but since there were no reasons for not answering, neither short time nor fear of getting a low score on the test we decided that this was legitimate.
- Time pressure: we know that the 20 minutes given to complete the terminology questionnaire was sufficient time, but still some did not answer all statements.

5.3.2 The icon experiment

- Domain knowledge and questionnaire design: the students were not tested in their understanding of the specific domain modeled and the questions dealt with the CORAS language notation only. The models were not too technical detailed, but taken from a real security analysis and therefore representative for its type.
- Familiarity with material: the students had seen the CORAS icons before and were more used to them than the other icons. This is a problem, even if we used identical models, kept the stereotype name in both and replaced the CORAS icons with two standard UML icons the students with CORAS models had a small advantage. Still we believe this was too small to make a real difference, in most cases the group without CORAS icons had equal or higher mean score than the other group.
- Time pressure: it was important to have some time pressure to investigate fast model navigation. The questionnaire also contained extra tasks in the end to make sure that no subjects finished before time and thereby could disturb the other subjects.

5.3.3 General

- Norwegian experiment material: the findings reported in this paper are based on experiments with Norwegian students. Even though the experiment material was mainly in Norwegian and used Norwegian subjects we assume that our findings are not limited to apply only for Norwegians. The conceptual model is developed in an international project and all the concepts have precise translations into Norwegian.
- Student as subjects: Using students as subjects will always be criticized, but we believe that students are well suited for this explorative study. There are no other requirements than being familiar with a part of the system assessed in order to

participate in or understand the documentation from a security analysis. These experiments did not require knowledge about a particular system; we only tested terminology and modeling notation. In our opinion, unless a company already conducts security analyses using this type of terminology, the concepts tested here will be just as unfamiliar for industrial employees as for students. Many of the students were 6-9 months from graduating with a master degree in informatics, and many will become employees in companies that conduct this type of security analyses. They are comparable to technical experts in an analysis.

- Selection of subjects: The participants in our experiments were volunteers and not picked out according to background or experience. Like in a real security analysis we do not require the participants to know terminology or notation in advance.
- The "student-teacher"-setting: both the experiments were completely anonymous and the students did not have to worry about grades which could potentially influence their behavior.

6 Lessons learned

While the original model needs a textual explanation in order to be understood correctly, we aim to develop a new model that needs a minimum of extra explanation. The results from the terminology experiment have taught us that some parts of the conceptual model are not as intuitive as desired. On the basis of these results we propose a set of changes to the original model (see **Figure 13**):

- To avoid confusion about frequency, probability and consequence we have included probability as a specialization of frequency.
- In order to emphasize that a risk consists of a frequency value, consequence value and an unwanted incident, these terms have been grouped together in a logical manner that illustrates how they are components of a risk. The black diamond symbolizes that if a risk is eliminated this will also remove the frequency and consequence values (composition). The white diamond means that the unwanted incident is a part of the risk, but if the risk is eliminated the unwanted incident may still exist in other risks (aggregation).
- The direct relation between treatment and unwanted incident has been removed and instead we have connected treatment to risk. By doing this we specify that a treatment is directed towards a risk, but not whether it targets a vulnerability, a threat, an unwanted incident or a combination of these.
- The association between asset and risk was a major source for misunderstandings. Nevertheless we have decided to keep the relation because we believe it is important. The subjects seem to think of an unwanted incident in the way we use the term risk, and by removing the direct association between asset and unwanted incident we hope that this misunderstanding will be less common.
- We chose to remove "entity" from the model. The term was often misunderstood (48.9% incorrect or uncertain answers) and it is not natural to speak of general entities in a structured brainstorming, only entities assigned value which then are called assets.
- The association between asset and vulnerability has been changed from a regular relation to become a part of the asset in the sense that an asset *can* have a vulnerability.
- In the original conceptual model it is not explicitly modeled that "a threat exploits a vulnerability to initiate an unwanted incident". This is an intuitive relation that achieved a high percentage of correct answers and we feel it is correct to make it clearer than it was in the original model.
- The new model tries to emphasize that an unwanted incident is a part of a risk and therefore one or more threats are always associated with the risk through its unwanted incident.
- We have chosen to highlight concepts grouped together in the form of compositions and aggregations, i.e. vulnerability is tightly connected to asset and risk is a concept that includes three other concepts.

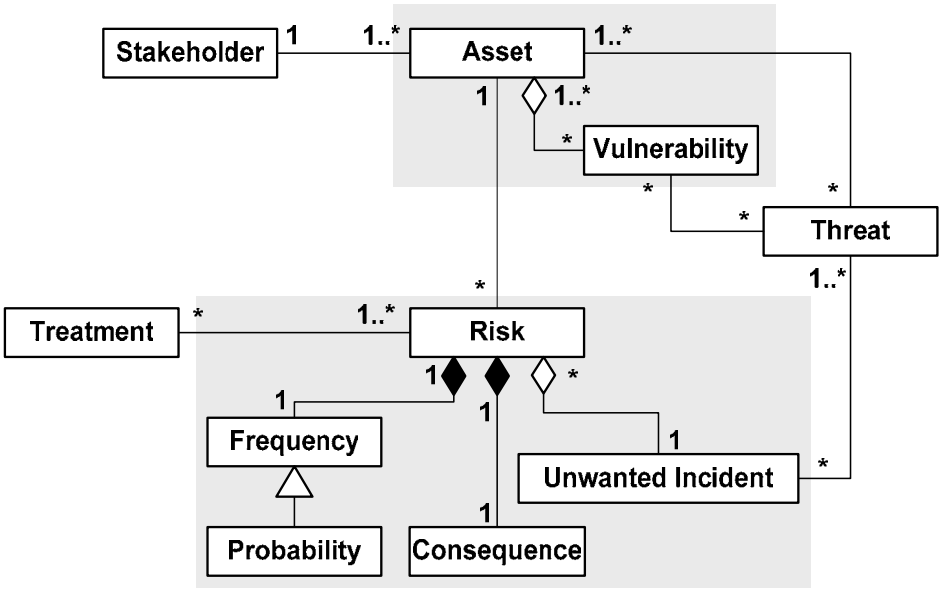


Figure 13 The revised conceptual model

7 Conclusions and recommendations

A structured brainstorming within security analysis gathers system experts with the aim of identifying risks and treatments for the system. The participants need to have a common understanding of concepts like asset, stakeholder and threat, in addition to more traditional security analysis terms. If the analysis leader asks them to identify unwanted incidents for the system, it is important that the participants understand that this is different from risks or threats. They need to understand a security risk scenario quickly and interpret it in the same way. The CORAS language for structured brainstorming is specially developed to support structured brainstorming and document the results. The language is based on a conceptual model, consisting of security analysis specific terms. The suitability of its notation and its underlying conceptual model is very important for the usability of the language. We aimed to investigate this in two experiments, one focusing on the terminology in the underlying conceptual model and the other on the use of graphical icons. We used students as subjects and the results from this study will be used as input in further experiments with professionals to see if we obtain similar results.

The results from the terminology experiment showed that few subjects had problems with relations like stakeholder and asset or threat and vulnerability. These terms are part of the daily language and most people have an intuitive understanding of them. Concepts like frequency, consequence and probability created more confusion and a large majority was not aware that probability is kind of frequency. These results show the importance of using terms that are intuitive and do not conflict with the daily language.

Even though the CORAS language is a UML profile, it uses special graphical icons to symbolize selected security analysis terms. The results from the icon experiment showed that stereotyping with CORAS icons vs. UML icons improves the speed of navigation, i.e. identification of specific model elements. The results showed a statistical significant difference in the number of questions completed in favor of the group using CORAS icons. The icons did not affect the correctness of interpretation of models.

The conclusion from our icon experiment is that using CORAS' graphical icons helps the participants in a structured brainstorming session to identify the same model elements faster than if one does not use CORAS icons. When it comes to correctly understand the models we believe it is necessary to look at other aspects as well as graphical icons.

Future Work

The future goal of our work is to develop a modeling language that supports and documents structured brainstorming, whose usability and suitability is thoroughly supported by empirical and analytical evidence. The assessment of the CORAS language has provided us with information that will be used as input for design of this new language. In addition to more student experiments, we have planned an analytical assessment of the CORAS language, based on the quality framework of [17]. Through industrial field studies in the SECURIS project we aim to test our hypotheses and

gather experience from real users. The language will be tested in SINTEF ICT's usability laboratory using subjects that are representative for real users. The results from these investigations will help us identify the requirements for a language that in the best possible ways supports and documents brainstorming sessions in security analysis.

Acknowledgements

The research on which this paper reports has partly been funded by the Research Council of Norway project SECURIS (152839/220). The authors thank Jan Heim and Fredrik Seehusen (SINTEF ICT) for valuable input.

References

[1] Australian/New Zealand Standard for Risk Management (AS/NZS 4360): Strathfield Standards Australia, 1999.

[2] British Standard BS4778: Quality vocabulary: Availability, reliability and maintainability terms. Glossary of international terms: British Standards Institute, 1991.

[3] CORAS, http://coras.sourceforge.net

[4] IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related (E/E/PE) Systems, 1998-2000.

[5] IEEE-1471: "Recommended Practice for Architectural Description of Software-intensive Systems" (IEEE1471-2000 Std): IEEE, 2000.

[6] IEEE Std 610.12: IEEE Standard Glossary of Software Engineering Terminology, 1990.

[7] ISO/IEC 13335: Information technology - Guidelines for management of IT Security, 1996-2000.

[8] ITU-T X.800: Security architecture for open system interconnection for CCITT applications. (Technically aligned with ISO 7498-2). Geneva, 1991.

[9] Object Management Group (OMG), http://www.omg.org

[10] J. Ø. Aagedal, F. den Braber, T. Dimitrakos, B. A. Gran, D. Raptis, and K. Stølen, "Model-based risk assessment to improve enterprise security," *Proc. Enterprise Distributed Object Communication (EDOC'2002)*, pp. 51-62, 2002.

[11] C. Atkinson, T. Kühne, and B. Henderson-Sellers, "Stereotypical Encounters of the Third Kind," *Proc. 5th International Conference on The Unified Modeling Language*, pp. 100-114, Dresden, 2002.

[12] F. den Braber, T. Dimitrakos, B. A. Gran, M. Soldal Lund, K. Stølen, and J. Ø. Aagedal, *UML and the Unified Process*: IRM Press, 2003.

[13] F. den Braber, M. S. Lund, and K. Stølen, "Using the CORAS Threat Modelling Language to Document Threat Scenarios for several Microsoft relevant Technologies," SINTEF ICT, Technical Report (STF90 A04057), 2004.

[14] F. den Braber, M. S. Lund, K. Stølen, and F. Vraalsen, "Integrating Security in the Development Process with UML," in *Encyclopedia of Information Science and Technology*, M. Khosrow-Pour, Ed.: Idea Group Reference, 2005, pp. 1560-1566.

[15] F. den Braber, A.-B. Mildal, J. Nes, K. Stølen, and F. Vraalsen, Experiences from Using the CORAS Methodology to Analyze a Web Application, in *To appear in Journal of Cases on Information Technology*, 2005.

[16] T. Dimitrakos, B. Ritchie, D. Raptis, J. Ø. Aagedal, F. den Braber, K. Stølen, and S. H. Houmb, " Integrating model-based security risk management into eBusiness systems development - the CORAS approach.," *Proc. 2nd IFIP Conference on E-Commerce, E-Business, E-Government (I3E'2003)*, pp.159-175, 2002.

[17] J. Krogstie and S. d. F. Arnesen, "Assessing Enterprise Modeling Languages Using a Generic Quality Framework," in *Information Modeling Methods and Methodologies*: Idea Group, 2005, pp. 63-79.

[18] L. Kuzniarz, M. Staron, and C. Wohlin, "An Empirical Study on Using Stereotypes to Improve Understanding of UML Models," *Proc. 12th IEEE*

International Workshop on Program Comprehension (IWPC'04), pp.14-23, Bari, Italy, 2004.

[19] M. S. Lund, I. Hogganvik, F. Seehusen, and K. Stølen, "UML profile for security assessment," SINTEF ICT, Technical report (STF40 A03066), 2003.

[20] F. Redmill, M. Chudleigh, and J. Catmur, *Hazop and software Hazop*: Wiley, 1999.

[21] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*: Addison Wesley Longman, Inc., 1998.

[22] R. E. Walpole, R. H. Myers, and S. L. Myers, *Probability and Statistics for Engineers and Scientists*, 6 ed: Prentice Hall International, Inc., 1998.

Appendix A – Icon Experiment Material

The material used for the icons experiment was made in two versions, one with CORAS icons and one with standard UML icons, in the following we present both versions. The material is divided into three parts, Part 1 and 2 have one set of questions each and Part 3 has 4 sets. The fourth set of questions is a "fill-in" task, included to make sure that the subjects do not finish too early.

A.1 Material with CORAS icons

Part 1

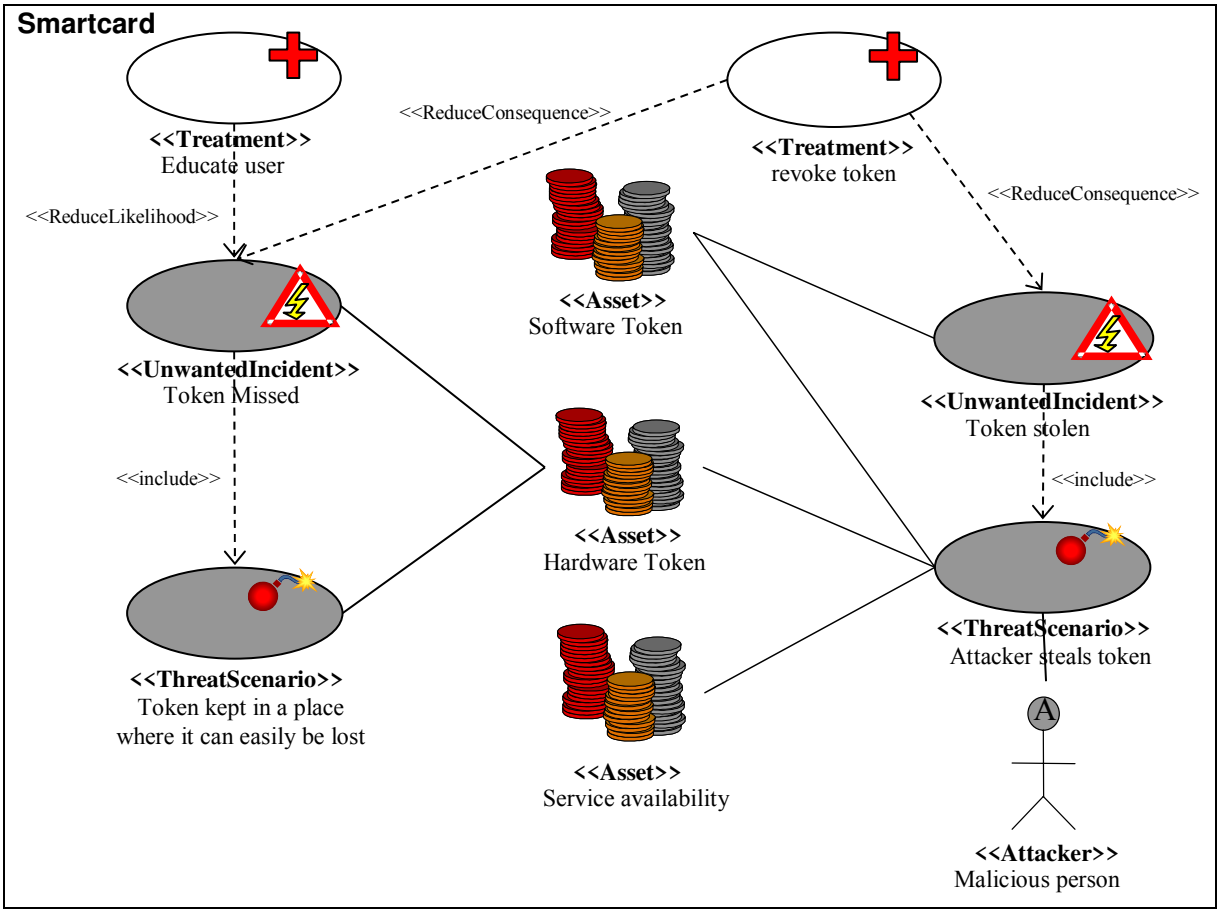


Figure 14 Smartcard

Set 1 – Smartcard

A. How many assets are explicitly modeled in the diagram above?

B. How many unwanted incidents are explicitly modeled in the diagram above?

C. How many threats are explicitly modeled in the diagram above?

D. How many risks are explicitly modeled in the diagram above?

E. How many threat scenarios are explicitly modeled in the diagram above?

F. How many treatments are explicitly modeled in the diagram above?

G. How many vulnerabilities are explicitly modeled in the diagram above?

Part 2

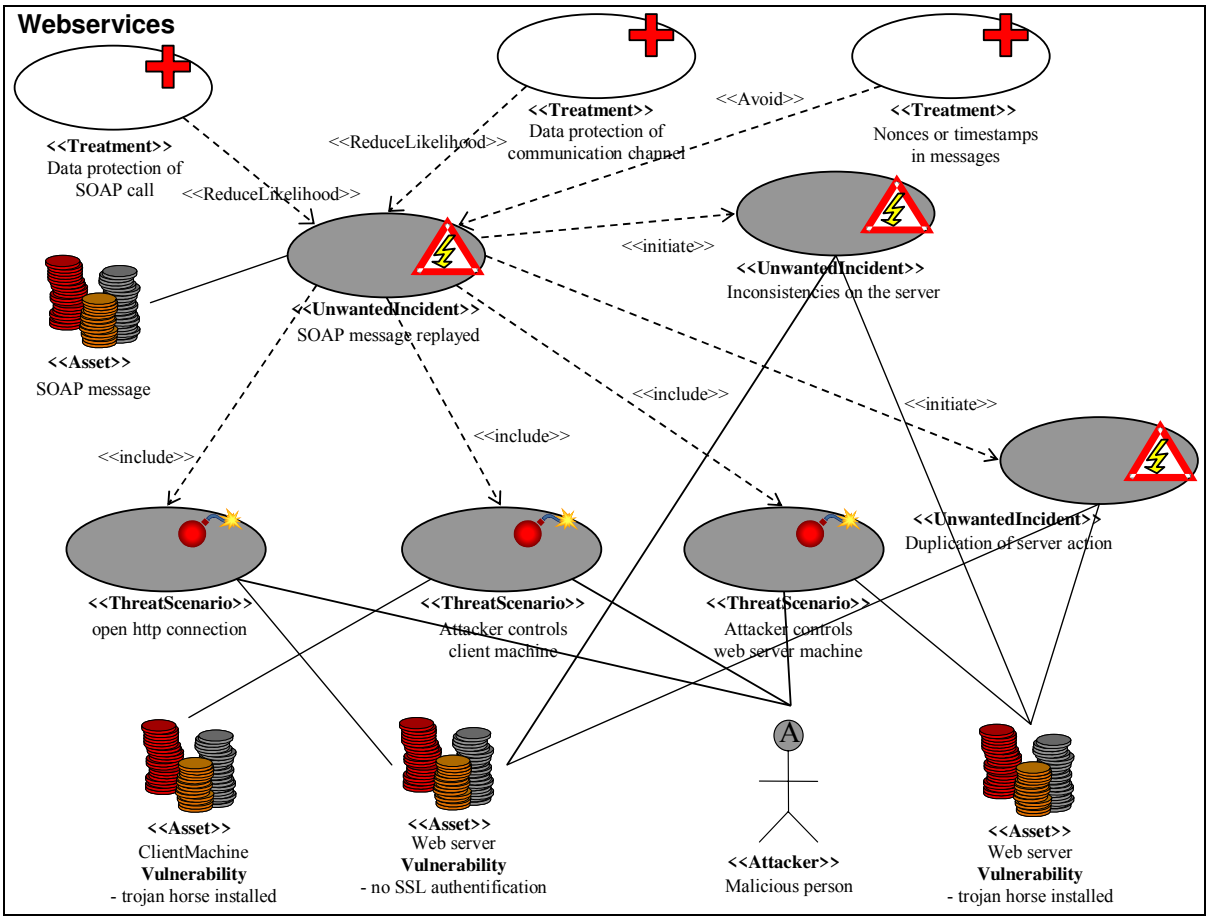


Figure 15 Web services

Set 1 – Web services

A. How many threat scenarios are explicitly modeled in the diagram above?

B. How many unwanted incidents are explicitly modeled in the diagram above?

C. How many risks are explicitly modeled in the diagram above?

D. How many types of icons are used in the diagram above?

E. Which assets are affected by the unwanted incident "SOAP message replayed"?

F. How many assets are explicitly modeled in the diagram above?

G. How many vulnerabilities are explicitly modeled in the diagram above?

H. Which threat scenarios affect the asset "Web server"?

I. How many threats are explicitly modeled in the diagram above?

J. How many treatments are explicitly modeled in the diagram above?

Part 3

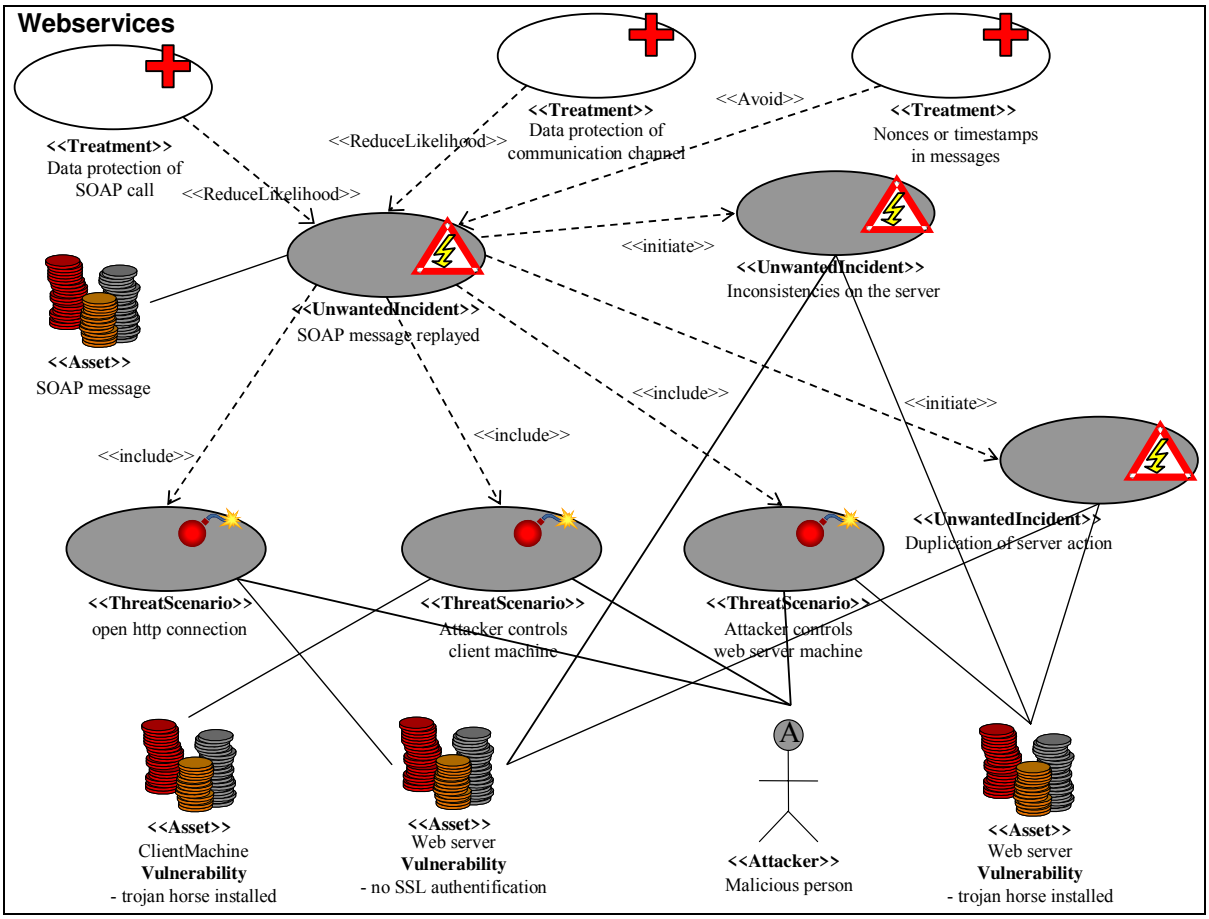


Figure 16 Web services

Set 1 – Web services

A. *Please record the current time (hh:mm)*
B. *What does a stapled arrow tagged with <<include>> mean?(use an example from the diagram above)*
C. *What does a solid line mean? (use an example from the diagram above)*
D. *Below each icon there is some text, what is the difference between the text put in angel brackets (<<text>>) and the other text?*
E. *How should one read the relation between "Nonces or timestampes in messages" and "SOAP message replayed"?*

Set 2 – Web services

A. *Please record the current time (hh:mm)*
B. *Which unwanted incident(s) affects the asset "Web server" in*

C. *Figure* **16***?*

D. *Which unwanted incident(s) affects the asset "Client machine" in*

E. *Figure* **16***?*

F. *Which unwanted incidents, assets and threats are associated with the threat scenario "Attacker controls web server machine" in*

G. *Figure* **16***?*

H. *What is the effect of the treatment "Nonces or timestamps in messages" applied against the unwanted incident "SOAP message replayed" in*

I. *Figure* **16***?*

J. *How should one read the relation between "SOAP massage replayed" and "Inconsistencies on the server" in*

K. *Figure* **16***?*

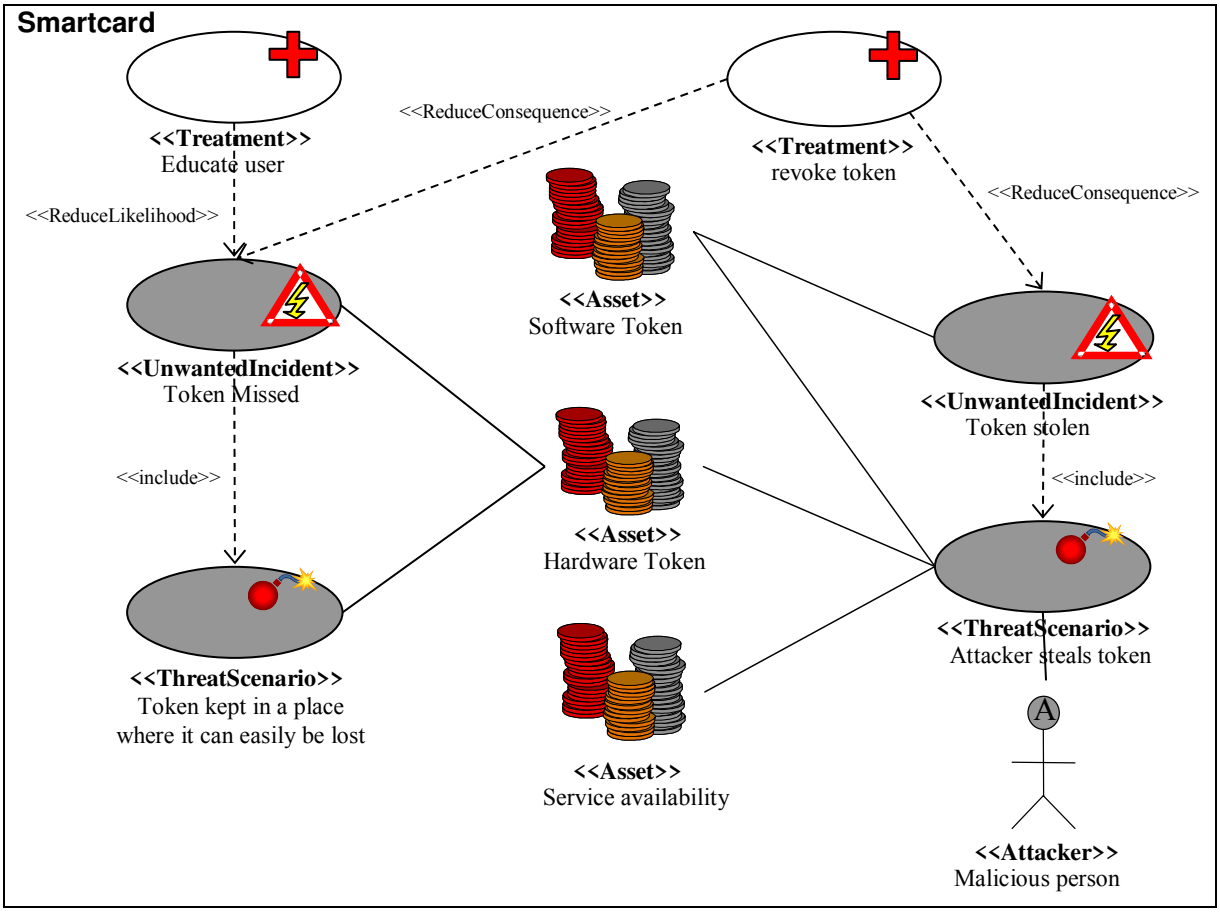


Figure 17 Smartcard

Set 3 – Smartcard

A. Please record the current time (hh:mm)

B. Can you explain the relation between: "Token missed", "Token kept in a place where it can be lost" and "Hardware token"?

C. If one removed the relation between "Hardware token" and "Token missed", how would this affect your explanation to B?

Set 4 - General

A. Please record the current time (hh:mm)

B. Explain why you think the icons you have seen in the diagrams have been selected.

C. *How would you have modeled the situation where to unwanted incidents exploits to separate vulnerabilities of one and the same asset?(draw a suggestion)*

A.2 Material without CORAS icons

Part 1

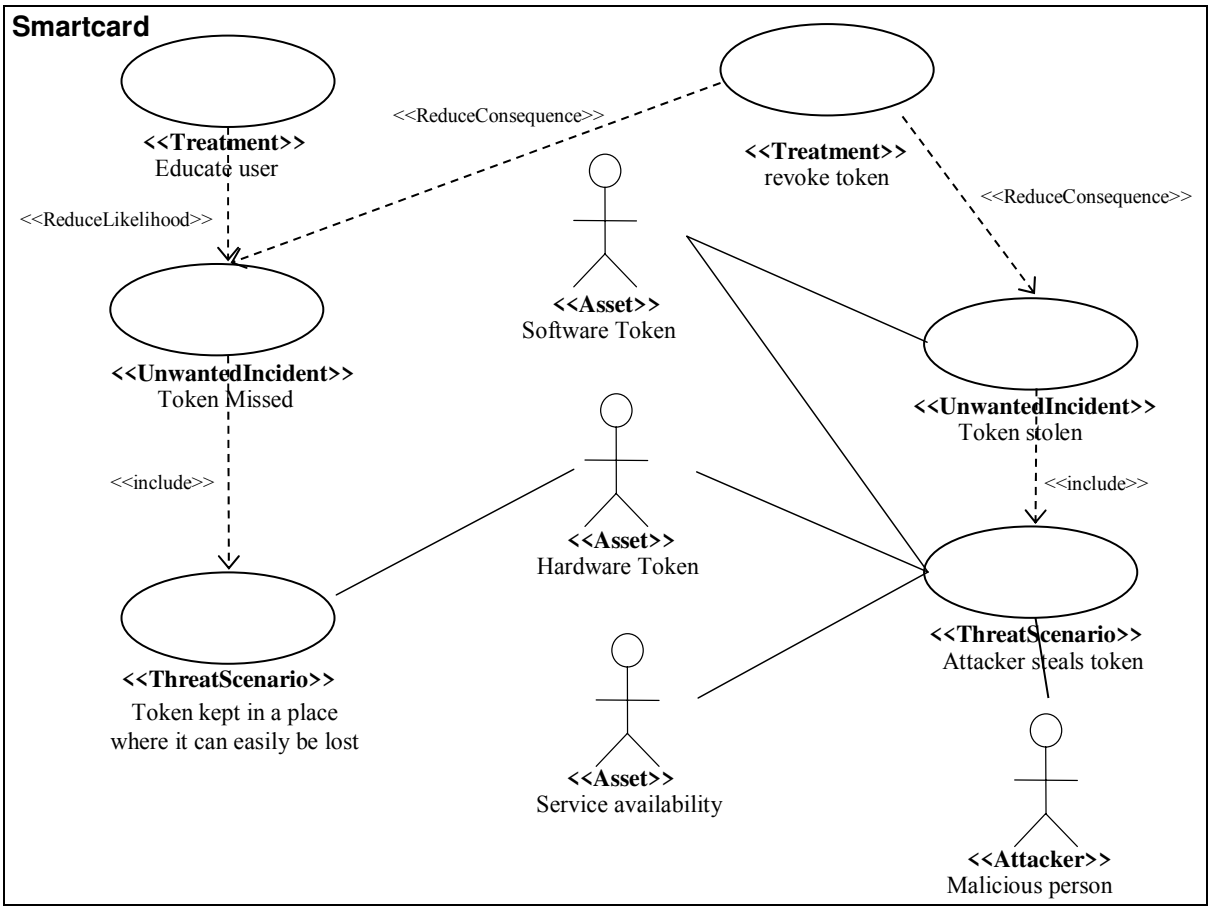


Figure 18 Smartcard

Set 1 – Smartcard

A. How many assets are explicitly modeled in the diagram above?

B. How many unwanted incidents are explicitly modeled in the diagram above?

C. How many threats are explicitly modeled in the diagram above?

D. How many risks are explicitly modeled in the diagram above?

E. How many threat scenarios are explicitly modeled in the diagram above?

F. How many treatments are explicitly modeled in the diagram above?

G. How many vulnerabilities are explicitly modeled in the diagram above?

Part 2

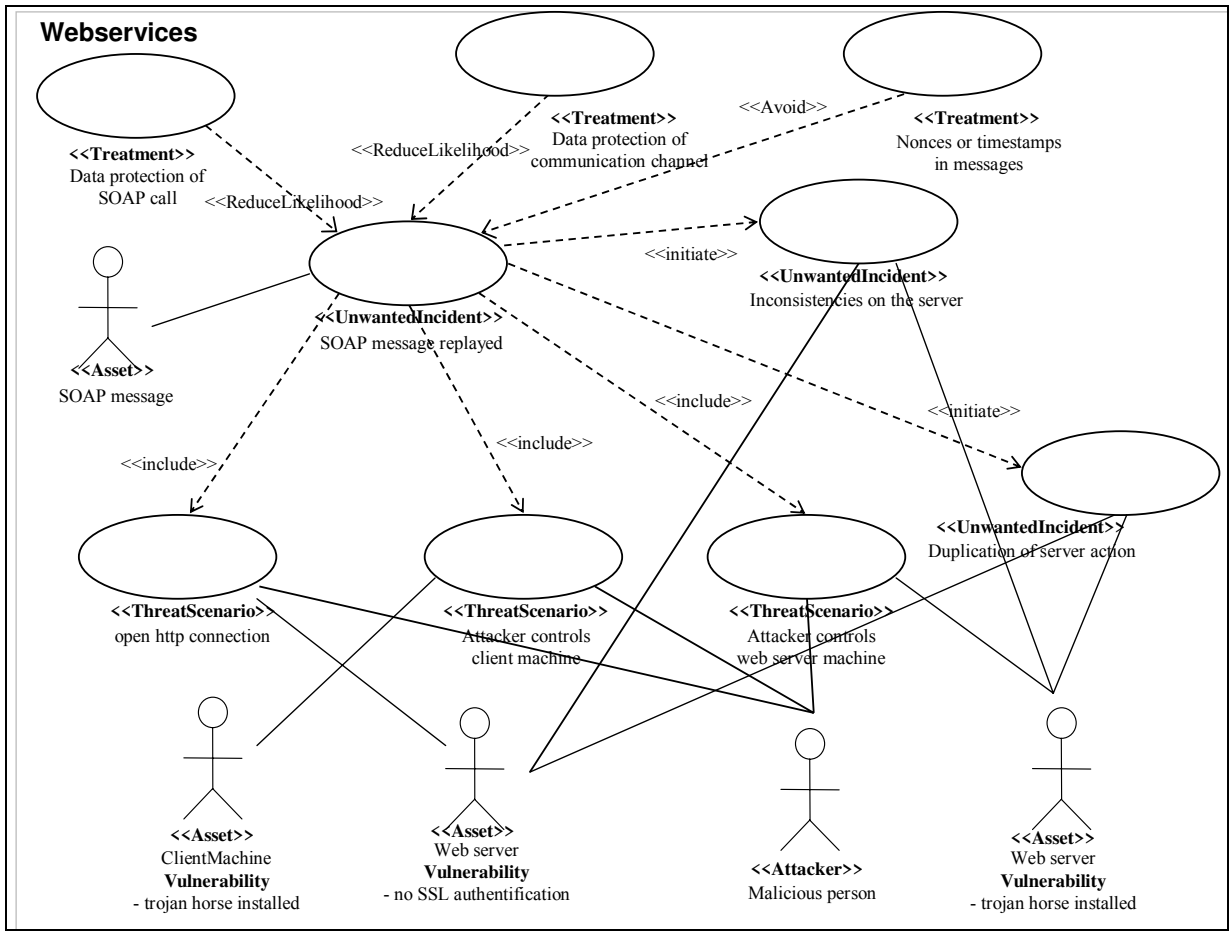


Figure 19 Web services

Set 1 – Web services

A. How many threat scenarios are explicitly modeled in the diagram above?

B. How many unwanted incidents are explicitly modeled in the diagram above?

C. How many risks are explicitly modeled in the diagram above?

D. How many types of elements are used in the diagram above?

E. Which assets are affected by the unwanted incident "SOAP message replayed"?

F. How many assets are explicitly modeled in the diagram above?

G. How many vulnerabilities are explicitly modeled in the diagram above?

H. Which threat scenarios affect the asset "Web server"?

I. How many threats are explicitly modeled in the diagram above?

J. How many treatments are explicitly modeled in the diagram above?

Part 3

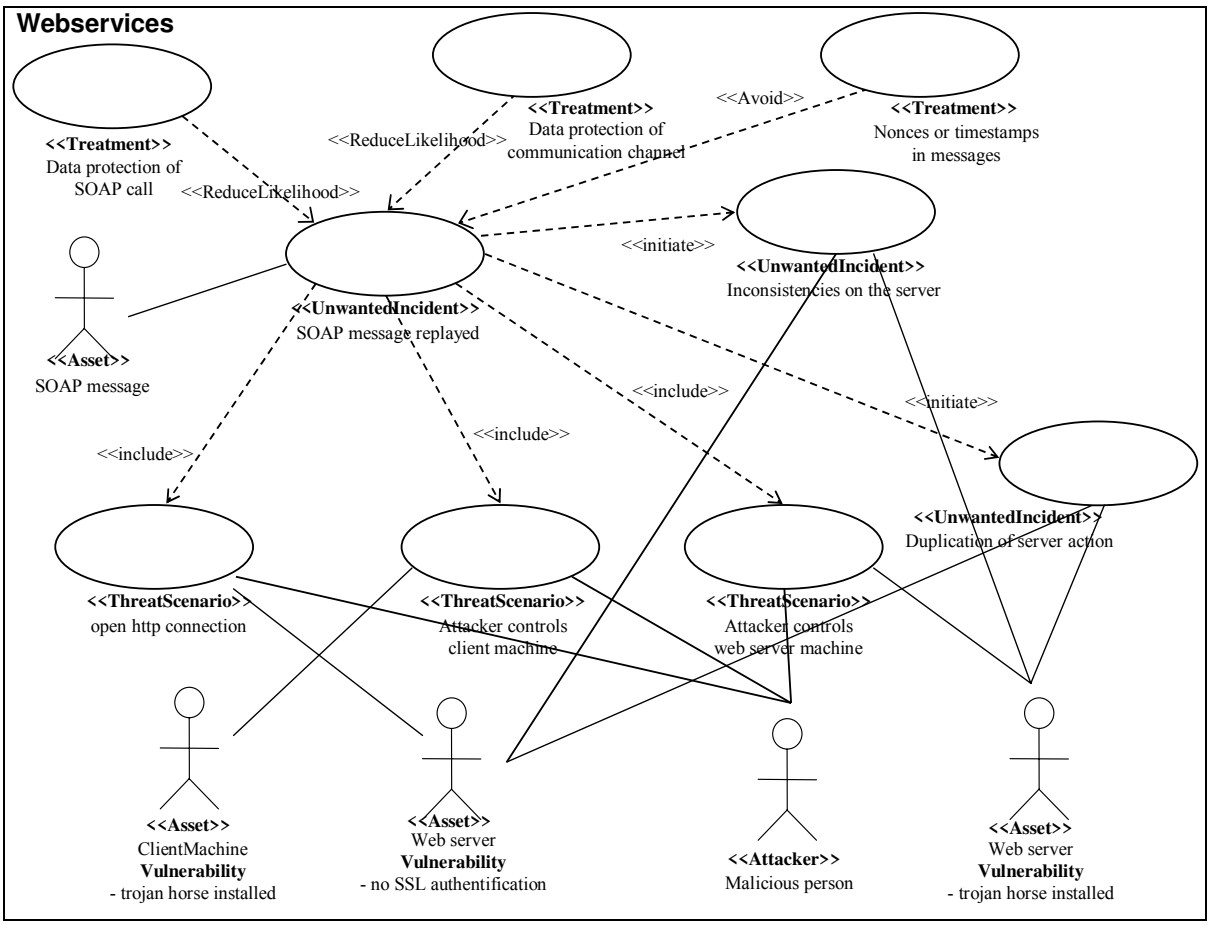


Figure 20 Web services

Set 1 – Web services

A. Please record the current time (hh:mm)

B. What does a stapled arrow tagged with <<include>> mean?(use an example from the diagram above)

C. What does a solid line mean? (use an example from the diagram above)

D. *Below each icon there is some text, what is the difference between the text put in angel brackets (<<text>>) and the other text?*
E. *How should one read the relation between "Nonces or timestampes in messages" and "SOAP message replayed"?*

Set 2 – Web services

A. *Please record the current time (hh:mm)*
B. *Which unwanted incident(s) affects the asset "Web server" in Figure 20?*
C. *Which unwanted incident(s) affects the asset "Client machine" in Figure 20?*
D. *Which unwanted incidents, assets and threats are associated with the threat scenario "Attacker controls web server machine" in Figure 20?*
E. *What is the effect of the treatment "Nonces or timestamps in messages" applied against the unwanted incident "SOAP message replayed" in Figure 20?*
F. *How should one read the relation between "SOAP massage replayed" and "Inconsistencies on the server" in Figure 20?*

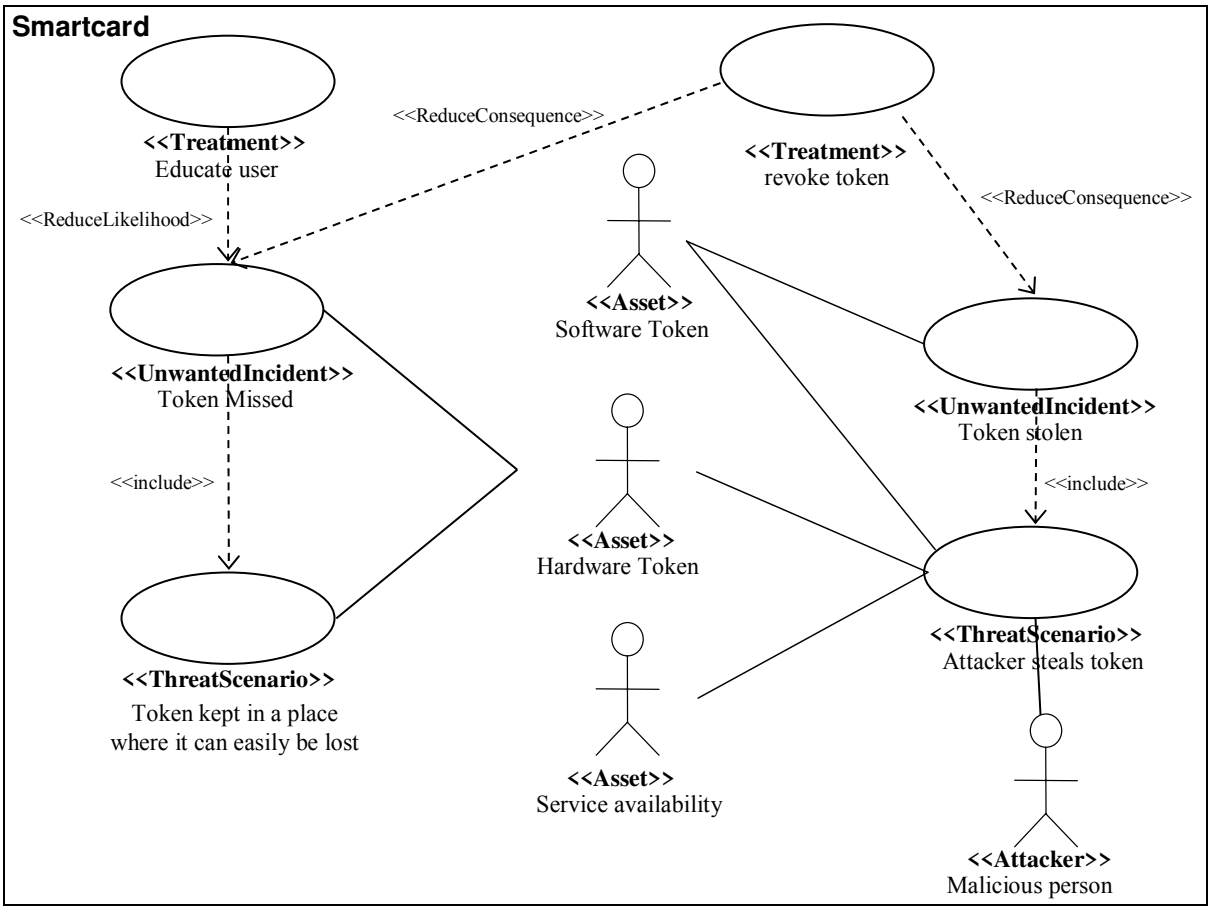


Figure 21 Smartcard

Set 3 – Smartcard

A. Please record the current time (hh:mm)

B. Can you explain the relation between: "Token missed", "Token kept in a place where it can be lost" and "Hardware token"?

C. If one removed the relation between "Hardware token" and "Token missed", how would this affect your explanation to B?

Set 4 – General

A. Please record the current time (hh:mm)

B. How would you have modeled the situation where to unwanted incidents exploits to separate vulnerabilities of one and the same asset?(draw a suggestion)

Appendix B Suggested Solutions for the Icon Experiment

Part 1

Set 1 – Smartcard

A. 3

B. 2

C. 1

D. 0, risks are not explicitly modeled in this type of diagram

E. 2

F. 2

G. 0

H. ”Hardware token” 2 stk.

Part *2*

Set 1 – Webservice3

A. 3

B. 0, risks are not explicitly modeled in this type of diagram

C. 5

D. All

E. 3

F. 3

G. ”Open http connection” and ”Attacker controls web server machines”

H. 1

I. 3

Part 3

Set 1 - Web services

A. -

B. An "include"- relation between an unwanted incident and a threat scenario means that the arrow from the unwanted incident points at the threat scenario(s) that is (are) included.

C. There is some sort of relation between them. An unwanted incident and a threat scenario may be related to an asset or a threat is an actor in a threat scenario.

D. The stereotype name in angle brackets (<< >>) has its origin from UML and denotes the type of element. The other text is the name or description of the element.
E. “If you apply nonces or timestamps in the messages you will avoid the problem with replay of SOAP messages”.
F. Yes all 3, (but it is possible to look upon these as “indirect” threat scenarios since they are included in the “predecessor” of this unwanted incident).

Set 2 – Web services

A. -
B. All three (NB: ”Web server” is modeled twice to emphasize that two different threat scenarios exploit two different vulnerabilities on the same asset. If both vulnerabilities had been modeled on the same asset-icon it would have been impossible to know which vulnerability is associated with each of the threat scenarios)
C. ”Soap message replayed”
D. All three unwanted incidents, two assets (”Web server”, ”Soap Message”), one threat.
E. By applying this treatment one avoids the unwanted incident.
F. The first initiates, or leads to the other.

Set 3 – Smartcard

A. -
B. The threat scenario "Token kept in.." is a part of the unwanted incident "Token missed" and affect the asset ”Hardware token”.
C. Unchanged interpretation since the threat scenario is included in the unwanted incident.

Set 4 - General

A. –
B. –
C. –

Appendix B

Risk Analysis Terminology for IT systems: Does it Match Intuition?

SINTEF REPORT

SINTEF ICT

Address: P.O.Box 124, Blindern
0314 Oslo NORWAY
Location: Forskningsveien 1
0373 Oslo
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

TITLE

Risk Analysis Terminology for IT systems: Does it match Intuition?

AUTHOR(S)

Ida Hogganvik, Ketil Stølen

CLIENT(S)

Norges Forskningsråd (NFR) / The Research Council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
STF90 A06034	Unrestricted	NFR152839/220	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
A	82-14-02806-4	40332800	30/3
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.) Ketil Stølen	CHECKED BY (NAME, SIGN.) Jan Heim
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.) Bjørn Skjellaug, Research Director	
	2006-09-24		

ABSTRACT

Many risk specific concepts like "threat", "consequence" and "risk" belong to the daily language. In a risk analysis one cannot be certain that the participants' interpretation of these terms is in accordance with risk analysis definitions. Risk analyses often use brainstorming techniques to identify risks based on the opinions and judgments of system experts. Such techniques employ risk specific terminology, and to avoid misunderstandings and uncertainty among the participants, it is important that the terminology does not conflict with the daily language understanding of these terms. We have developed a formal, conceptual model of IT system risk analysis terminology and investigated to what extent this model corresponds to the common, or intuitive, understanding of the concepts.

The mean score for all the subjects was 59% correct answers, but there are major differences according to which risk-related concept that is in question. Asset and vulnerability related questions receive the highest number of correct answers (72-65%), while the frequency related obtain the lowest score, only 45%.

This report presents the results from an empirical study of the understanding of risk related terminology conducted with 57 subjects, both professionals and students.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Risk analysis terminology	Riskioanalyseterminologi
GROUP 2	Empirical investigation	Empirisk undersøkelse
SELECTED BY AUTHOR	Conceptual model	Konseptuell modell

TABLE OF CONTENTS

1 INTRODUCTION 1

2 RISK ANALYSIS OF IT SYSTEMS 3

2.1 STRUCTURED BRAINSTORMING 3
2.2 RISK ANALYSIS TERMINOLOGY 3

3 STUDY DESIGN 7

3.1 SUBJECTS 7
3.2 MATERIAL 7
3.3 ANALYSIS MODEL 9

4 RESULTS AND DISCUSSION 11

4.1 THE DIFFERENCE BETWEEN THE QUESTION CATEGORIES 12
4.2 CONCERNING ASSET AND VULNERABILITY 13
4.3 CONCERNING RISK, UNWANTED INCIDENT, FREQUENCY-MEASURES AND CONSEQUENCE 14
4.4 CONCERNING TREATMENT 16
4.5 CONCERNING THREAT 16

5 THREATS TO VALIDITY 19

6 CONCLUSION 21

6.1 ASSET & VULNERABILITY 21
6.2 RISK 21
6.3 TREATMENT 22
6.4 THREAT 22
6.5 REVISING THE CONCEPTUAL MODEL 22
6.6 FURTHER WORK 23
ACKNOWLEDGEMENTS 23

REFERENCES 24

APPENDIX A - FREQUENCY TABLE (Q3-Q16) 25

APPENDIX B - DIFFERENCES BETWEEN QUESTION CATEGORIES 26

APPENDIX C - DIFFERENCES IN SCORE BETWEEN STUDENTS AND PROFESSIONALS 29

LIST OF FIGURES

FIGURE 1 – THE CONCEPTUAL MODEL THAT WAS USED AS BASIS FOR THE STUDY 5
FIGURE 2 – PERCEIVED ROLES 11
FIGURE 3 – FAMILIARITY WITH RISK ANALYSIS TERMS 11
FIGURE 4 – RESULTS FOR EACH OF THE ANSWER ALTERNATIVES 12
FIGURE 5 – PERCENTAGE OF CORRECT ANSWERS FOR EACH CATEGORY (MEAN) 12
FIGURE 6 – ASSETS IN A SECURITY ANALYSIS 13
FIGURE 7 – ONE UNWANTED INCIDENT: TWO RISKS 14
FIGURE 8 – UNWANTED INCIDENT AND RISK 15
FIGURE 9 – THE GOAL OF A TREATMENT 16
FIGURE 10 – POTENTIAL THREATS 17
FIGURE 11 – RISK COMPONENTS 22
FIGURE 12 – SPECIALIZING THREAT 22
FIGURE 13 – LIKELIHOOD COVERS THE OTHER TERMS 23
FIGURE 14 – THE REVISED CONCEPTUAL MODEL 23
FIGURE 15 – MEAN SCORE: STUDENTS VS. PROFESSIONALS 29

LIST OF TABLES

TABLE 1 – STATISTICAL DIFFERENCES 13
TABLE 2 – RESULTS Q3-Q16, (N=57) 25
TABLE 3 – THE RESULTS FROM WILCOXON'S TEST 26
TABLE 4 – RANKS 26
TABLE 5 – GROUP STATISTICS (GROUP 1=STUDENTS, 2= PROFS.) 29
TABLE 6 – INDEPENDENT SAMPLES TEST 29

1 Introduction

In software development it is common to conduct some form of analysis of the product to assess how secure it is. One may e.g. assess the system's ability to resist unauthorized access or whether the data is transmitted securely. The participants in a risk analysis face questions like: "What are the system threats and its vulnerabilities?" "What are the assets in your system which you want to protect?" "Are there unwanted incidents that can harm your assets?" "What is the frequency of this unwanted incident?" The questions make use of a specialized risk analysis terminology, but many of the terms are also commonly used in the daily language.
Standards for risk analysis of IT systems [1, 7, 10-12] are often inspired by standards developed for risk analysis in e.g. mechanical engineering or the process industry [8, 9]. Within these domains there is a long tradition for using risk analysis, and its terminology is quite well established, but is this also the case within IT systems development?

A previous study with students showed that there were diverging opinions on how to interpret concepts used in risk analysis [13, 14]. Concrete, tangible concepts like asset and threat were found easier to understand than more abstract terms like risk. The results from the study made us revise our conceptual model of risk analysis terminology, leading to the version we investigated in this study (Figure 1). To identify the concepts and relationships that are straightforward and those that are more problematic, we conducted a questionnaire among 57 Norwegian subjects representative for normal risk analysis participants. The investigation tested how risk analysis terms are interpreted by subjects (both students and professionals) with various competences on risk analysis. We assumed that many knew these terms from before, but how well did our conceptual model correspond with the subjects' interpretation?

The investigation revealed that the subjects' interpretations match our model quite closely, in average the subjects obtained almost 60% correct answers (here correct means in accordance with our model). 60% is an acceptable rate taking into consideration that many of the participants were more or less unfamiliar with risk analysis. We found significant difference according to which risk-related concept that was investigated. Asset and vulnerability seems to be the easiest understood with a higher number of correct answers than e.g. frequency related concepts which scored lowest. There was a statistical significant difference between software engineering students' score and the professionals', which indicates that people's background affects how well IT system risk analysis terminology is understood.

This report is structured as follows: Section 2 provides background information on risk analysis, structured brainstorming and risk analysis terminology. Section 3 describes the study design, and Section 4 provides the results and is also where we discuss our findings. Section 5 reports the threats to the validity of our results, and finally Section 6 presents the main conclusions.

2 Risk Analysis of IT systems

This section provides an introduction to risk analysis of IT systems and its terminology.

2.1 Structured brainstorming

Risk management of IT systems is the total process of identifying, controlling, and eliminating or minimizing uncertain events that may affect IT system resources [10]. Risk analysis is part of the management process and often make use of structured brainstorming techniques (i.e. what [15] calls HazOp-analysis). A structured brainstorming for identification of risks is a methodical "walk-through" of the target of analysis. Experts on different aspects of the target system are involved to identify system risks, threats and vulnerabilities, and sometimes suggest possible treatments for the risks. The participants in a brainstorming (in addition to the analysis team) can potentially be anyone associated with the target system. The authors of [15] recommend these roles to be represented: (1) *user* or *intended user* of the system assessed, (2) *experts* on one or more aspects of the system, (3) *system designer*, (4) *analysis leader* and (5) *analysis secretary*. The latter two are part of what we call the analysis team, which is responsible for guiding the analysis. Often the only form of preparation the participants receive is an introduction to brainstorming at the beginning of the session. Experts may in some cases only contribute in the part of the analysis where their expertises are needed. During the brainstorming session, the group of experts assesses system descriptions in a stepwise and structured manner, supported by guidewords that help them identify vulnerabilities and threats associated with risks. The findings are documented by the analysis secretary.

2.2 Risk analysis terminology

The participants in a structured brainstorming need a clear understanding of the terminology used. The individual interpretation of terms like frequency, probability, vulnerability and asset may differ depending on the participant's background and experience. Several of the concepts are employed in daily language, but not with a unique, context independent interpretation. Normally misunderstandings will be cleared up as the analysis progress, but our goal is to minimize them by using an intuitive language.

We have developed a conceptual model inspired by the model used in an IT system risk analysis method called CORAS (http://coras.sourceforge.net). CORAS was developed in a 4 year long EU project and gathers well known risk analysis methods into an integrated risk analysis framework based on several international standards [1, 3-7, 10-12]. The terminology definitions are mainly taken from ISO/IEC13335: Information technology – Guidelines for the management of IT Security [10] and AS/NZS4360: Australian / New Zealand Standard for Risk Management [1]. The model is shown in Figure 1 using class diagram notation from the Unified Modeling Language (UML) [16]. The associations between the elements have cardinalities specifying the number of instances of one element that can be related to one instance of the other (example: "a stakeholder has at least one and maximum an infinite number of assets; and an asset belongs to only one stakeholder"). The black diamond symbolizes aggregation and the white composition. Elements connected with a composition can also be part of other compositions, while aggregated elements only exist within the specific aggregation.

We explain Figure 1 as follows: the IT system or product evaluated during a risk analysis is called the target of evaluation [3], target system or target of analysis. Stakeholders are those people and organizations who may affect, be affected by, or perceive themselves to be

affected by, a decision or activity regarding the target system [1]. Different stakeholders often value the system differently; a system user who is dependent on the system may put a high value on it, while other stakeholders may not value it equally high. An asset is something to which a stakeholder directly assigns value and, hence, for which the stakeholder requires protection [4]. An asset is uniquely linked to a single stakeholder. The type of assets may differ according to the scope of the analysis. In a security-focused analysis the assets may be less obvious than in a safety-focused analysis. A safety analysis is often concerned about protecting people, the environment or equipment. In a security analysis assets can also be customer information, source code, company brand, critical system services etc. A stakeholder wants to protect his/her assets from being compromised. Target system stakeholders and their assets are normally identified early in process.

Figure 1 includes four important risk analysis concepts related to asset: vulnerability, unwanted incident, threat and risk. A vulnerability is a weakness of an asset or group of assets which can be exploited by one or more threats [10]. One may understand a vulnerability as something that is "missing", e.g. if a company network lacks a firewall it may be a vulnerability with respect to some assets in the network. An unwanted incident [10] is an event that may harm or reduce the value of assets and is something we want to prevent. An unwanted incident is the result of a threat exploiting a vulnerability. If a server in a company network is an asset, then an unwanted incident is unauthorized access to the server by intruders. A threat is a potential cause of an unwanted incident [10]. A risk is the chance of something happening that will have an impact upon objectives (assets)[1]. A risk consists of an unwanted incident, the chance of it happening and its consequence. If we continue the example about the company network, a threat may be a person who knows or discovers the network's vulnerability and wants to exploit it. The company may not recognize the potential risky situation because nobody outside the company is aware of the security hole, but if e.g. an employee is fired one suddenly has a risk of unauthorized network access by people familiar with the company's infrastructure. The level of risk is measured by a risk value [1] (e.g. low, medium, high or other scales) which is based upon the estimated frequency for the unwanted incident to happen and its consequence in terms of loss of asset value. Consequence is sometimes called impact [10]. Frequency is the exact number of occurrences within a time period, in addition to frequency there exist two other measures for how often a risk may occur: likelihood and probability. Likelihood is a qualitative description of probability or frequency [1] (e.g. "unlikely", "possible", "certain"), whereas probability is a number between 0-1 often represented as percentage (e.g. "a 50% chance for a risk to happen"). According to [10] probability should be used to characterize risk, while [1] suggests likelihood. If a risk is estimated to occur two times a year and the consequence is a loss of 200.000 dollars each time, the risk value could be "high" which means the risk should be treated. A treatment (in [10] called safeguard) is the selection and implementation of appropriate options for dealing with risk [1]. A treatment can either be directed towards the unwanted incident, the threat or the asset's vulnerability (or a combination of these). The desired effect of a treatment is reducing risk frequency and/or consequence (i.e. risk value) to an acceptable level.

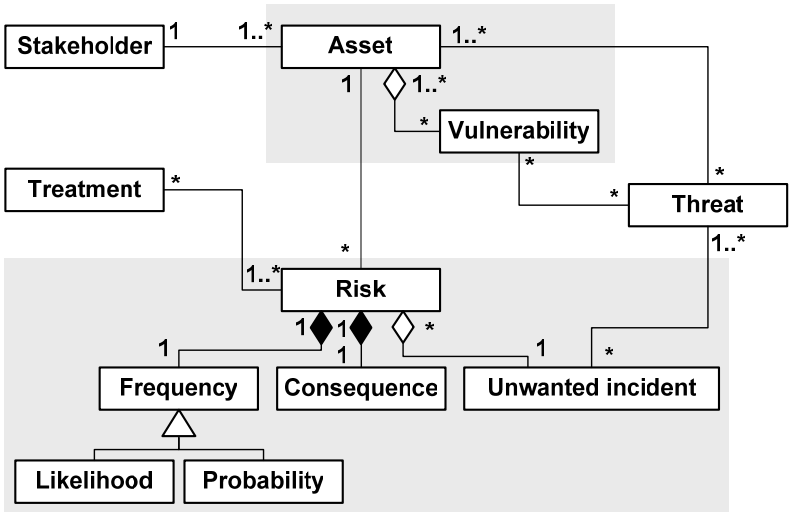


Figure 1 – The conceptual model that was used as basis for the study

.

3 Study Design

This section describes the subjects, the material and the analysis model.

3.1 Subjects

We aimed to select subjects that are representative for participants in a risk analysis of an IT system. In this type of risk analysis the participants may have various competences on risk analysis, but they are experts on one or more aspects of the IT system assessed. Our investigation did not consider a specific system, therefore we included subjects with general experiences from IT system development. The 57 subjects represent four groups:

- 12 professional researchers, including 2 industrial psychologists, from the Institute for Energy Technology (IFE), who had competence on developing safety critical systems for nuclear power plants (http://www.ife.no)
- 22 professional researchers, including 3 industrial psychologists, from SINTEF ICT (Information and Communication Technology), who had competence on software development, modeling, and usability studies (http://www.sintef.no)
- 14 master students in informatics who were attending a network security course at the University of Oslo (http://www.uio.no)
- 9 master students in informatics who were attending a safety critical software development course at Østfold University College (http://www.hiof.no)

The subjects from SINTEF ICT were paid for the time they spent on filling in the questionnaire, while the students and the IFE professionals participated voluntarely. The students ranged from some experience with risk analysis to none, while the professionals ranged from high competence on risk analysis to only superficial knowledge. The subjects received no training or introduction to our conceptual model in advance as we aimed to capture their intuitive understanding of the concepts.

3.2 Material

The subjects were given a multiple choice questionnaire with 16 questions or statements about risk analysis terminology (see frame below). The questions can be grouped into six categories, each concerning a specific risk concept. Question 3-5 (Q3-Q5) are asset-related, Q6-Q8 are treatment related, Q9-Q10 are frequency measure related, Q11-Q13 are riks related , Q14 relates to vulnerability and Q15-Q16 relate to threats. The material was distributed in English due to several reasons: the CORAS method is documented in English, it is also common for Norwegian companies to use English in reports and presentations, and higher education in software engineering is mainly based on English literature. This means that within software development good skills in English is essential.

The subjects had to judge each answer alternative as either false, partly false, partly true, true or uncertain. The questionnaire covers risk analysis concepts and their relations; and the subjects were given 20-25 minutes to complete it. Please note that even if the questionnaire occasionally uses the term “security analysis”, the risk analysis terminology tested is general and therefore not restricted to security-focused analyses. The true alternatives are marked with *.

1) Which roles could you have in a risk analysis?
2) How well do you know risk analysis terminology?
3) What can be considered as assets in a security analysis?

a. customer register *
b. a company brand *
c. critical system services *
d. human lives *
e. equipment *
f. source code *
g. a company's strategies *
h. employees' competence *

4) What is true about the relationship between risk and asset?
a. a risk can harm or reduce the value of the asset *
b. a presumption for a risk to arise is that there are vulnerabilities for someone to exploit *
c. a risk will harm or reduce the value of an asset
d. one risk may harm more than one asset

5) What is the difference between a risk and an incident scenario for an asset?
a. an incident scenario is an actual event that can harm the asset but without a frequency or consequence value *
b. a risk for an asset is an unwanted incident assigned a consequence and frequency value *
c. an unwanted incident cannot be part of more than one risk

6) The goal of a treatment can be:
a. to reduce risk *
b. to remove a threat *
c. to remove a vulnerability *
d. to reduce a threat *
e. to remove a risk *
f. to reduce a vulnerability *
g. to remove an unwanted incident *

7) What is true about treatment?
a. it is usually directed towards a vulnerability *
b. it is usually directed towards a threat *
c. it is usually directed towards an unwanted incident *
d. it is always directed towards a risk *
e. it can be directed towards both a vulnerability and threat *
f. it cannot be directed towards both a vulnerability and an unwanted incident at the same time
g. it cannot be directed towards both a vulnerability, a threat and an unwanted incident at the same time

8) When will a treatment be considered successful?
a. if it reduces risk frequency
b. if it reduces risk consequence
c. if it makes the risk value acceptable *
d. if it eliminates the risk *
e. if it reduces both frequency and consequence

9) What is the relationship between frequency, likelihood and probability?
a. the term frequency comprises both likelihood and probability *[1]
b. likelihood is measured qualitatively *
c. probability is measured quantitatively *

[1] In later revisions of the Australian/New Zealand risk management standard [2] likelihood has changed its definition and therefore made this alternative invalid, but in the analysis we use the old interpretation.

d. frequency is measured only quantitatively
e. likelihood is measured as a value between 0-1

10) What can be used to calculate risk value?
a. probability and frequency
b. likelihood and consequence *
c. consequence and frequency *
d. consequence and probability *
e. frequency and likelihood

11) What is a risk composed of?
a. a consequence, a frequency, a probability and an unwanted incident
b. a probability, an unwanted incident and a consequence *
c. a consequence and a frequency and an unwanted incident *
d. none of the alternatives above

12) What is the relationship between risk and unwanted incident?
a. a risk is part of an unwanted incident
b. a risk initiates the unwanted incident
c. an unwanted incident is part of a risk *
d. an unwanted incident can be a part of more than one risk *

13) When can one consider an unwanted incident a risk?
a. when it has a consequence, but not necessarily a frequency
b. when it has a frequency, but not necessarily a consequence
c. when it has both a frequency and a consequence *

14) What is true about vulnerability?
a. it can be a weakness or lack of the asset itself *
b. it can be a weakness or lack of the asset's surroundings *
c. a threat can exploit vulnerabilities *

15) What can be considered a threat?
a. hardware *
b. people *
c. software *
d. an event (initiated by a person)

16) What is true about threat?
a. a threat can initiate an unwanted incident *
b. a threat can constitute a risk even if there are no vulnerabilities to exploit
c. a threat can potentially reduce the value of an asset *
d. a risk is always associated with a threat *
e. a threat is not necessarily connected to a risk *

3.3 Analysis model

The data was analyzed with respect to three aspects: (1) to what extent do the subjects answer correctly without being properly trained? (2) Is there any difference between the students' score and the professionals'? (3) Which concepts and relations are particular difficult or easy to understand? The two categories partly true and true were considered as one positive category, partly false and false constitute one negative category, while uncertain was one category on its own. The reason for combining categories is that one can probably not expect to find one unique definition of a term that everyone agrees to, and as long as the participants in an analysis at least partly agree we consider this a sufficient basis for conducting a successful analysis. The difference between the two groups was investigated using a one tailed t-Test, with a significance level $\alpha=0.05$. The differences between the six question

categories were investigated using the Wilcoxon test [17], also with a significance level α=0.05. We used SPSS v13.0 on Windows XP for all the statistical tests.

4 Results and Discussion

The underlying assumption for this work is that many potential risk analysis participants know some risk analysis terminology from their daily language, but we do not know whether their interpretation is in accordance with standard IT risk analysis definitions. The 57 subjects had different backgrounds, but everyone was either studying software engineering or working in an IT system development environment. Their experience with risk analysis of IT systems ranged from beginners to experts on risk analysis.

The initial two questions ask the subjects which roles they believe they may have in a security analysis, and how familiar they are with risk analysis terminology. The questions were included to give an impression of the population. The results provide insight to whether the subjects are familiar with risk analysis or not.

Since the roles may overlap the subjects were allowed to choose more than one; the result is shown in Figure 2. Three students answered "Risk analysis leader" but these answers were seen as unrealistic. We see that the majority of the subjects has competence from system design, they are technical experts or system users.

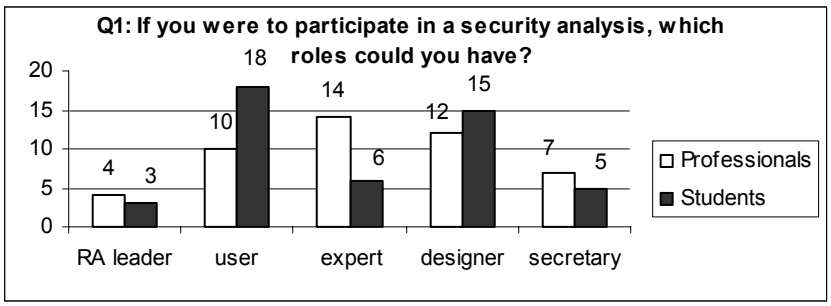


Figure 2 – Perceived roles

We investigated their familiarity of risk analysis terms to obtain an understanding of the background of the population (Figure 3).

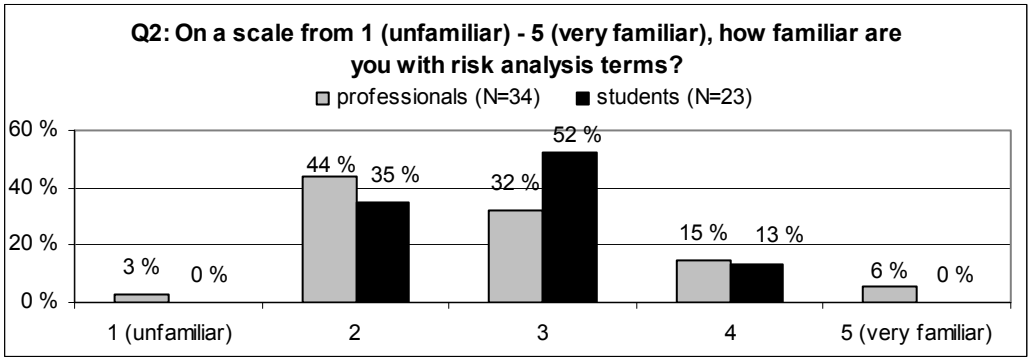


Figure 3 – Familiarity with risk analysis terms

The results from the study show that even without being trained in risk analysis, the subjects had a good understanding, or interpretation, of the most important risk analysis terms. In average the subjects obtained 58.6% correct answers, a fairly high number. Some concepts showed more difficult than others (e.g. risk, likelihood) and it would have been interesting to see whether similar results had been obtained with native English speakers.

The results for each answer alternative for question 3 to 16 is provided in Table 2, Appendix A. The mean score for the subjects is shown in Figure 4.

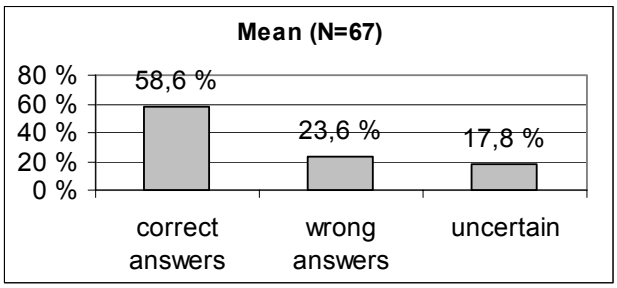


Figure 4 – Results for each of the answer alternatives

We checked the results for the professionals with the students and found a statistical difference in correct answers in favour of the professionals (Figure 15). This was expected since many of the professionals have worked with risk analysis-related topics. With respect to wrong and uncertain answers there is no significant difference between the students and the professionals. The details about the tests can be found in Appendix B.

More interesting than the difference between the groups is the investigation of the interpretation of the terminology. In the remaining of this section we discuss the findings related to concepts that were found particular difficult or very intuitive. Questions from the questionnaire are referred to using "Q" + their number, i.e. "Q5" means "question number 5".

4.1 The difference between the question categories

The questionnaire included 16 questions that can be divided into six categories which targets different risk related concepts. The groups were not equal with respect to the number of correct answers, and the difference in mean is shown in Figure 5.

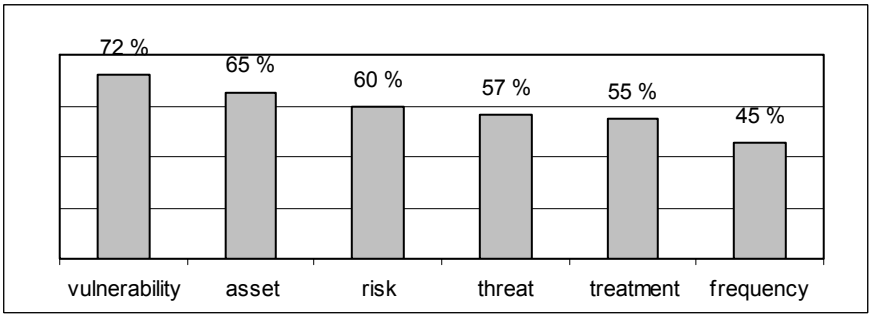


Figure 5 – Percentage of correct answers for each category (mean)

As we can see the vulnerability related questions obtained a higher number of correct answers than the frequency related category. To test whether the differences are significant we used the Wilcoxon test which gave these results:

- The vulnerability category and the asset category have significantly more correct answers than all the other categories, but are not significantly different from each other.
- The risk category, threat category and the treatment category are not significantly different from each other, but there they are all significantly different from the frequency measure category.

The results are presented with their p-value in Table 1where the categories are compared with each other, a value below 0,05 indicates a significant difference. For more detail about the statistical tests we refer to Appendix B.

Table 1 – Statistical differences

Categories	Asymp. Sig. (2-tailed)
treatment-asset	0,00
frequency-asset	0,00
risk-asset	0,05
vulnerability-asset	0,07
threat-asset	0,01
frequency-treatment	0,02
risk-treatment	0,24
vulnerability-treatment	0,00
threat-treatment	0,59
risk-frequency	0,00
vulnerability-frequency	0,00
threat-frequency	0,01
vulnerability-risk	0,03
threat-risk	0,48
threat-vulnerability	0,00

4.2 Concerning asset and vulnerability

These categories received the highest number of correct answers. The high score indicates that these topics can be considered easier to understand than the other ones. We asked the subjects what they consider as assets in a security analysis (Q3, Table 2) and the result is illustrated in Figure 6. Under half of the respondents think of "company brand" as an asset in a security analysis, even though brand name can be more expensive to repair/replace than "equipment". In fact over one third means that the company brand is not a relevant asset in a security analysis.

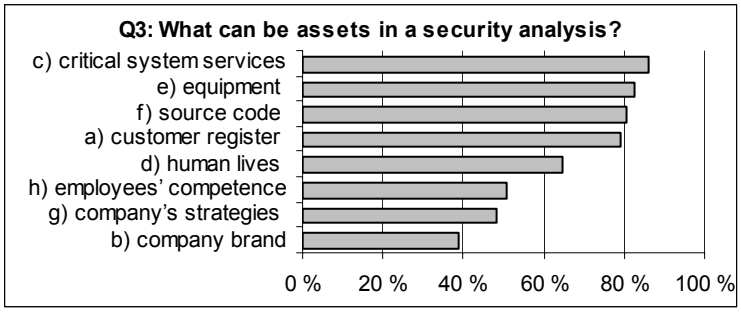


Figure 6 – Assets in a security analysis

We can see from these results that "things" are considered more important than "intangible" assets like company strategies. While security used to focus on protecting physical belongings or valuables, one is now concerned about securing information, reputation, strategies etc. as well. According to these results we can assume it to be challenging to make the participants in an analysis realize what their intangible assets are and also assign them appropriate values. Experience from our industrial field trials supports this assumption; often the participants find identification and valuation of assets particularly hard.

"Employee's competence" was included as an alternative seen from a company's perspective, meaning the risk of loosing core competence. Later we have realized that it is possible to

interpret this from an employee's point of view, i.e. the risk of an unwanted incident harming his/her competence, which was not the intention and therefore it will not be discussed.

A vulnerability is always associated with an asset. In practice a vulnerability can be either an aspect of the asset itself (a too simple password) or in its "surroundings" (the password stored in clear text). Many of the subjects think of a vulnerability as a weakness in the asset's surroundings rather than a part of the asset itself (Q14, Table 2). Both interpretations are covered by our conceptual model, and do not seem to create particular problems. The model also lets one vulnerability yield for more than one asset, which often is the case. A large group also agrees that there have to be vulnerabilities to exploit before a risk can arise (Q4, Table 2). In practice this means that it is possible to limit the scope of the analysis to only consider assets that have vulnerabilities.

4.3 Concerning risk, unwanted incident, frequency-measures and consequence

Risk is the most important term in a risk analysis. Nevertheless our studies have shown that it is one of the most difficult terms to fully understand. According to our definition, a risk can be understood as the relation between one unwanted incident and one asset, which means that one risk is only associated with one asset. Unfortunately the results show that a majority believe one risk may harm more than one asset (Q4, Table 2). The rationale for our definition is to cover the cases where an unwanted incident harms more than one asset and the consequences or frequencies are different. To specify this case they should be considered as two different risks (Figure 7) which also makes it possible to specify different and more specialized treatment alternatives.

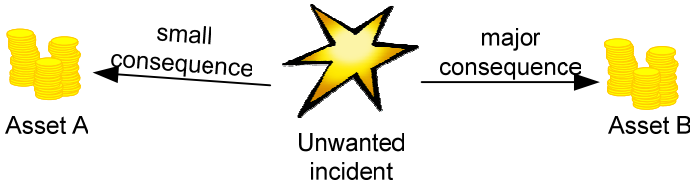


Figure 7 – One unwanted incident: two risks

We do not think the conceptual model should be changed to reflect the subjects' interpretation, but we will have to emphasize the special risk-asset relation when conducting risk analysis. Nevertheless, a majority agree that an unwanted incident is a part of a risk (Q12, Table 2) and also that it can be part of more than one risk (Q5c and Q12d, Table 2). This means that the relationship between risk and unwanted incident seems quite clear. When 46% think a risk can initiate an unwanted incident (Q12), we believe this is due to the misunderstanding that a risk is the same as a threat.
The subject seems quite sure that a risk *can* harm rather than *will* harm an asset (Q4a and c, Table 2) which means that risk is seen as something "potential", i.e. something that may or may not happen and this is in accordance with the definition of the term. In our previous study [14] we also found that people were reluctant to accept the statement "a risk is something that will reduce the value of an asset" due to the use of "will".

More than two thirds of the subjects believe a risk is an unwanted incident assigned a consequence and a frequency value (Q5, Table 2), a high number that contradicts the result for Q11 where few subjects were able to state what a risk is composed of. This gives us reasons to suspect that the subjects did not truly understand Q5. The fact that the subjects are inconsistent in their answers indicates that risk analysis concepts can be quite confusing.

The terms frequency, probability and likelihood create a large amount of uncertainty (Q9). Frequency measures were also a problem in our previous study [14]. This investigation shows that likelihood has not a unique interpretation as qualitative, nearly half of the subjects say they are uncertain whether it is measured qualitatively or quantitatively, and half say they are uncertain on whether it is measured as a value between 0-1 (Q9, Table 2). Even though the term likelihood is difficult, we need a frequency measure that is not quantitative. In risk analyses of IT systems, one seldom has the appropriate frequency statistics and one has to rely on expert judgments. The definition of likelihood covers both probability and frequency and in our revised conceptual model likelihood will be used. The term itself will still be subject to further investigations, e.g. will native English speakers find it complicated as well? One of the disadvantages of using qualitative measures is that it makes it more difficult to calculate risk severity using a traditional risk analysis technique like Fault Tree Analysis (FTA) [6] which requires quantitative input.

Risk value is a term that people without risk analysis background will find hard to understand. Nevertheless over half of the subjects managed to select the correct alternatives for calculating risk value; "consequence & probability" (d) and "consequence & frequency" (c). As a contrast, less than half chose the alternative "consequence & likelihood" (b) (Q10, Table 2). We believe that a large group of the subjects guessed, or deduced what a risk value is and chose the statements in the question that contained "consequence" plus a frequency measure, i.e. recognized that probability, likelihood, and frequency are the same kind of measure.

Probability is without doubt seen as a quantitative measure among the subjects and nearly two third think frequencies are measured only quantitatively (Q9c and d, Table 2). All alternatives containing likelihood obtained a higher uncertainty rate than the others. It is up to the risk analysis team to decide on which measure to use in an analysis, but probability seems to be the one that is easiest understood.
When it comes to identifying which components a risk consists of, the subjects get quite confused. Over half of the subjects believe it is one of the alternatives presented, but is unable to choose the correct one (Q11, Table 2). Some even rejects all of the alternatives even though they according to the definition are all correct (alternative "a" includes more information than strictly necessary to characterize a risk).

Figure 8 shows that a majority will consider an unwanted incident as a risk if it has both frequency and consequence (Q13c, Table 2). Interestingly, less than half think an unwanted incident with consequence is a risk (Q13a, Table 2), while only a few consider an unwanted incident with frequency a risk (Q13b, Table 2). Again this shows that frequency is a difficult concept and that consequence is more easily associated with risk.

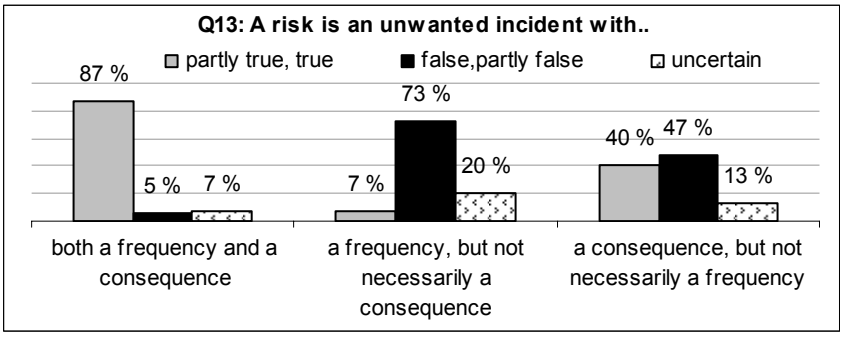


Figure 8 – Unwanted incident and risk

4.4 Concerning treatment

We asked the subjects what the goal of a treatment is, and where one usually applies it (Q6, Q7). We found that it is more common to consider "reduce risk" as the goal than "remove risk" (Figure 9 and Table 2). The subjects are reluctant to choose alternatives that include "remove", an option which probably seems more unlikely than "reduce". When it comes to where to apply the treatment, our model says that a treatment is directed towards a risk, without specifying exactly how, and many of the subjects agrees in this (Q7, Table 2) A treatment can be applied to anything that contributes to the occurrence of the risk. A majority of the subjects believe a treatment is something one applies to a vulnerability (Q6, Table 2), which is similar to the results from our previous study [14]. It is least usual to think of treatments directed towards threats and unwanted incidents (Q7b, c). In field trials we have experienced that the participants are more likely to come up with treatments that help reduce frequency (avoid incidents from happen) than treatments applied to reduce the consequences after incidents have happened.

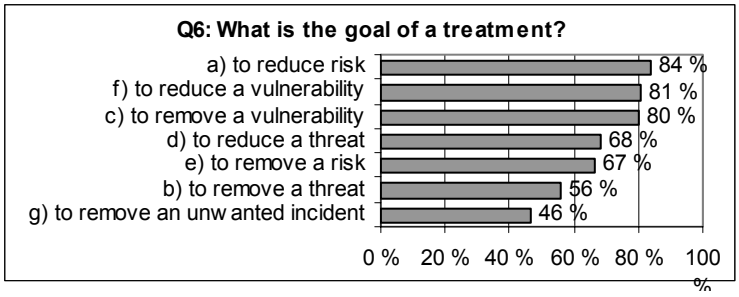


Figure 9 – The goal of a treatment

With respect to whether it is allowed to apply a treatment to more than one entity (e.g. treating both vulnerability and unwanted incident at the same time) the uncertainty rate increases.

We asked when a treatment is to be considered successful, but have later found the intention behind this question a bit tricky to "get" (Q8). Nevertheless, the essence of the answers clearly indicates that the subjects believe a successful treatment should reduce frequency and/or consequence. A large group considers a treatment successful if it eliminates a risk, which implies reducing frequency and/or consequence to zero (Q8d, Table 2).

4.5 Concerning threat

We distinguish between what a threat is and what it can do. When we asked what a threat is almost everyone considered people and events initiated by people as threats (Q15b,d, Table 2). Next, the subjects ranged software and hardware as potential threats (Q15c, a, Table 2) The result is shown in Figure 10 and means that an event is seen as a threat, even if there is a person behind it. This contradicts our definition where we consider the entity (human or non-human) that initiates the event as the real threat. Neither an explosion nor a security breach starts by itself, and the first step to solve the problem is to identify the source or initiator.

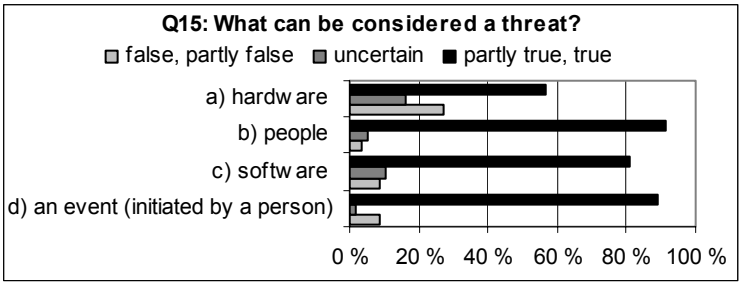


Figure 10 – Potential threats

We were surprised to find that nearly one third of the subjects think that hardware cannot be a threat, even though hardware failure most certainly can initiate unwanted incidents that may affect an IT system (Q15a, Table 2). In [10] a threat is specified to be human or environmental, while [1] simply speaks of identifying the "cause of event" and [12] categorizes threats as either accidental or intentional. In field trials we have experienced that the participants are well aware of the latter two, but less aware of non-human threats. To emphasize that a threat can also be non-human we will include human threat and non-human threat as specializations of threat in our revised model.

The majority believe that a threat has the ability to exploit vulnerabilities (Q14c, Table 2). They also think that a threat can initiate an unwanted incident and that it can potentially reduce the value of an asset (Q16a,c, Table 2). The subjects seem to be more uncertain whether or not a threat can constitute a risk even if there are no vulnerabilities to exploit (two third is either incorrect or uncertain) (Q16b, Table 2). Our model says that there have to be a vulnerability to exploit before one can speak of a risk. We define a risk to be associated with at least one threat, i.e. one initiator, and over half of the subjects agree (Q16d). On the other hand, we do not say that a threat has to be associated with a risk, but many disagree or are uncertain of that (Q16e). This relation is obviously quite difficult to comprehend. We believe that if one has identified a risk, meaning identified an unwanted incident with frequency and consequence, then it is likely that one also has identified what or who initiates the unwanted incident, i.e. the threat. A threat in itself is not associated with a risk unless there is a possibility of it initiating an unwanted incident. Therefore a risk is always related to one or more threats, but a threat is not necessarily related to a risk.

5 Threats to Validity

We have developed a conceptual model inspired by the one from CORAS which includes the most important terms from the CORAS risk analysis method. The terms' definitions are taken from the international standards [1, 3, 4, 10]. We do not attempt to cover all possible risk analysis terms.

The subjects were not native English speakers and the interpretation of the statements could have been affected by this. The study should be replicated with native English speakers to compare the results. Questionnaires formulated in natural language will always have room for misinterpretations. We did notice some minor vagueness in the text after the first run, but since the questionnaire was to be replicated we did not change the material. The conclusions drawn from the material has taken into account the potential misleading questions.

Participation was voluntarily and we do not know whether this affected the results. The subjects had various competences in one or more of the following domains (1) system modelling and design, (2) risk analysis and system development in the safety domain and (3) system development and security. The system modelling and design group had all heard a little about CORAS, but only a few had used it themselves. The professionals from the safety domain all knew something about software development and risk analysis, but less about security-focused risk analysis. The safety students knew some conventional risk analysis, and the security students were unfamiliar with both conventional risk analysis and security-focused risk analysis. A risk analysis team may very well be composed of people with similar knowledge of risk analysis as our subjects.

Even though we used master students attending a security course, it would have strengthened our results further if we had a group of professionals from the security domain like we had from the safety domain.

When comparing the categories like in this case, it is important to note than the number of question alternatives in each category is unequal. E.g. the vulnerability category has substantially fewer alternatives than the treatment category. Therefore the answers to each of the vulnerability questions will have more effect on the mean value than answers in larger categories.

The higher number of comparisons one makes, the more likely it is to find differences that are significant. In our work we have tested six categories, which give a total of 15 different comparisons. However, the high number of significant differences and the low p-values we obtained supports our conclusions that the differences found are in fact real.

6 Conclusion

This report presents the results from a study concerning the understanding of risk analysis terminology. The mean score for all the subjects was 58.6% correct answers, but there are major differences according to which risk-related concept that is in question. Asset and vulnerability related questions receive the highest number of correct answers (72-65%), while the frequency related obtains the lowest number (45%). We see that many of the terms we have in our conceptual model are well understood even by people without training in risk analysis. Being more experienced in risk analysis, the professionals obtained a higher number of correct answers than the students.

6.1 Asset & vulnerability

Assets are something of value to the system stakeholder, but getting the participants to fully understand the notion of asset appears challenging. Traditionally assets are considered physical entities, but in IT systems assets are often intangible objects like information, company reputation/brand etc. These can be both hard to identify and value appropriately. During our field trials we have experienced that the participants find it very hard to value an intangible asset in terms of money.

Assets are subject to weaknesses that make them vulnerable to threats. In our conceptual model a vulnerability is associated with one or more assets, but it is not a part of the asset. In this investigation we found that it is most common to think of a vulnerability as something in the asset's surroundings, meaning that our model does not conflict with the subjects' interpretation.

6.2 Risk

We define a risk to consist of an unwanted incident, a frequency and a consequence (Figure 11), and the subjects appear to have a clear understanding of an unwanted incident being a part of risk. On the other hand, it seems more complicated to know when an unwanted incident can be characterized as a risk. It looks like the subjects believe it is sufficient to establish the potential effect of the unwanted incident, but not how often it will occur. When the subjects were to judge whether an unwanted incident is a risk, it appears that a serious consequence is more important than a high frequency. This belief is reflected in the difficulties the subjects experienced with frequency measures, a concept that is less used in the daily language and associated with more uncertainty. In a real system, an unwanted incident that happens every day but with low consequence may over time cost just as much as a rare event with a serious consequence.

It is common to confuse how risk and unwanted incident relate to asset. In our definition an unwanted incident may harm more than one asset, but a risk is only associated with one asset. This enables us to describe an unwanted incident that has different consequence and/or frequency for a set of assets. The challenge is to make the participants understand that a risk is an "abstract" concept, while the unwanted incident is the "real" event.
Most of the subjects seem to have the correct interpretation of risk as something potential, meaning it *can happen*, not that it *will happen*, but still it should be emphasized that the risk is not only the incident (Figure 11).

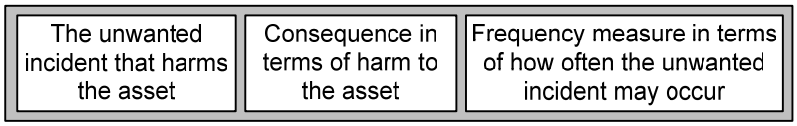


Figure 11 – Risk components

Our previous study [14] tested the understanding of probability and later included this as a frequency measure because it was a well understood concept. In this study we also included likelihood, which is a qualitative frequency measure. Likelihood was found to be a difficult and unfamiliar term without a unique interpretation. The problem may be that likelihood often is associated with mathematics and therefore people react to its definition as qualitative. In an analysis one seldom has appropriate quantitative data available, and one needs a qualitative measure similar to likelihood that can be estimated from experience or other relevant information. In lack of a better term, likelihood is included because of its suitable definition. Since it is more general than frequency (and probability) it will replace frequency in the revision of our conceptual model.

6.3 Treatment

Most subjects consider the goal of a treatment to be to reduce risk or reduce/remove vulnerabilities. The subjects consider a treatment as successful if it reduces the frequency and consequence of a risk, or if it eliminates risk. More subjects think a treatment is successful if it reduces consequence than if it reduces frequency. In our definition, a treatment is successful if it reduces consequence and/or frequency to an acceptable risk level. On the question of what it is most common to treat, most subjects gave priority to vulnerability and thereafter risk. This does not contradict our model where we define a treatment to be directed towards a risk, without specifying whether this means treating a vulnerability, a threat or an unwanted incident or a combination of these.

6.4 Threat

Human beings are most commonly viewed as threats, followed by events even if they are initiated by a person. According to our definition the initiator of the unwanted incident is regarded as the real threat and one needs to emphasize this. Surprisingly many do not consider hardware a threat. The concepts that most think of as threats are those involving humans which means that the term threat is interpreted as one with ability to do something. To emphasize that a threat can be human or non-human, this specialization will be included in our revised model.

6.5 Revising the conceptual model

The conceptual model of risk analysis terminology used as basis for our questionnaire, intended to incorporate the common risk analysis terms from CORAS and the interpretations of these. The results from this study show that it successfully reflects the subjects' common understanding of the risk analysis terminology. Still we like to make some minor adjustments based on the findings from this investigation. To increase the awareness of non-human threats the model should specify threat as human threat and non-human threat (Figure 12).

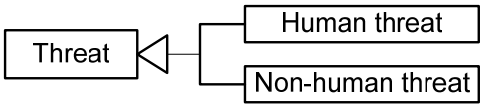


Figure 12 – Specializing threat

The current specialization of frequency into probability and likelihood is not satisfactory. Both frequency and probability are measures that can be covered by likelihood [2].

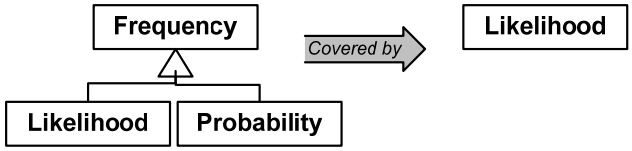


Figure 13 – Likelihood covers the other terms

Even though likelihood covers the other terms, it has shown as one of the most difficult concepts to understand. We will therefore accept any kind of frequency measure, including probability, frequency and likelihood. The final model is shown in Figure 14, the shaded areas are used to highlight concepts that are particularly close related to each other.

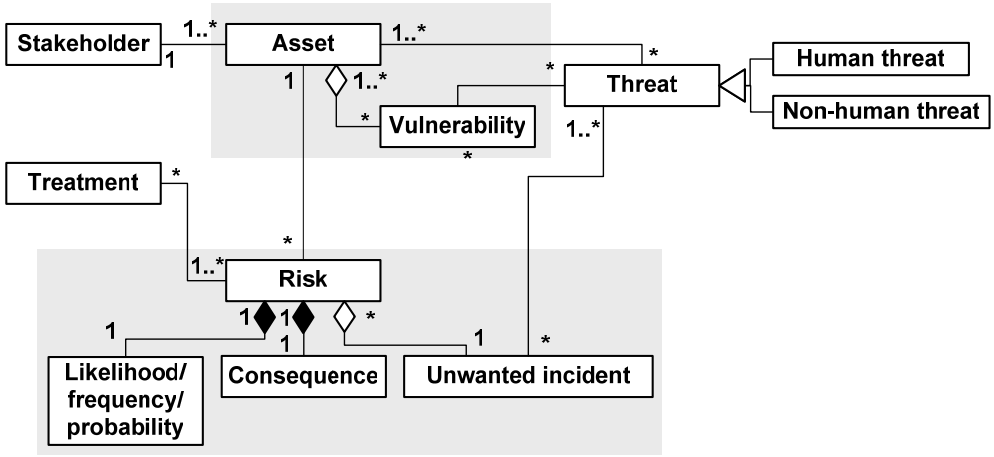


Figure 14 – The revised conceptual model

6.6 Further work

The findings from this study will be used in the development of a graphical modelling language for risk scenarios. The abstract syntax of the language will match our final conceptual model. These investigations will contribute to make the language as intuitive and easy to understand as possible. Our further hypothesis is that the relations that were found easy to understand in this investigation will also be most suitable and appropriate to model in a simple and straight forward manner, while the difficult concepts will need a more complex representation. This will be investigated in our further work.

Acknowledgements

The research on which this report builds on has been funded by the Research Council of Norway project SECURIS (152839/220). The authors thank Monica Kristensen (IFE), Lasse Øverlier (Norwegian Defence Research Establishment) and Jan Heim (SINTEF ICT) for valuable assistance and input. We are also grateful to the subjects for their participation.

References

[1] Australian/New Zealand Standard for Risk Management (AS/NZS 4360): Standards Association of Australia, 1999.

[2] Australian/New Zealand Standard for Risk Management (AS/NZS 4360): Standards Association of Australia 2004.

[3] British Standard BS4778: Quality vocabulary: Availability, reliability and maintainability terms. Glossary of international terms: British Standards Institute, 1991.

[4] HB 231:2000 Information security risk management guidelines, 2000.

[5] IEC 812 (IEC60812): Analysis techniques for systems reliability - Procedures for Failure Mode and Effects Analysis (FMEA), 1985.

[6] IEC 61025: Fault Tree Analysis (FTA): IEC, 1990.

[7] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems: IEC, 1998-2000.

[8] IEC 61511: Functional safety - Safety instrumented systems for the process industry, 2004.

[9] ISO 14121: Safety of machinery - Principles for risk assessment, 1999.

[10] ISO/IEC 13335: Information technology - Guidelines for management of IT Security: ISO, 1996-2000.

[11] ISO/IEC 17799: Information technology - Code of practice for information security management, 2000.

[12] ITU-T X.800: Security architecture for open system interconnection for CCITT applications. (Technically aligned with ISO 7498-2), 1991.

[13] I. Hogganvik and K. Stølen, "Empirical Investigations of the CORAS Language for Structured Brainstorming," SINTEF ICT, Technical report STF90 A05041, January 2005.

[14] I. Hogganvik and K. Stølen, "On the Comprehension of Security Risk Scenarios," presented at International Workshop on Program Comprehension (IWPC), 2005.

[15] F. Redmill, M. Chudleigh, and J. Catmur, *HAZOP and Software HAZOP*: Wiley, 1999.

[16] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*: Addison Wesley, 1998.

[17] R. E. Walpole, R. H. Myers, and S. L. Myers, *Probability and Statistics for Engineers and Scientists*, Sixth ed: Prentice Hall Int., 1998.

Appendix A - Frequency table (Q3-Q16)

Table 2 provides the results from the questionnaire showing the number of answers for each of the answer alternatives for question 3 to question16. The true alternatives are marked with *, and the number of non respondents is indicated in the n/a column.

Table 2 – Results Q3-Q16, (N=57)

	correct	incorrect	uncertain	n/a			correct	incorrect	uncertain	n/a
3a*	78,9 %	10,5 %	10,5 %	0,0 %		9a*	39,3 %	30,4 %	30,4 %	1,8 %
b*	38,6 %	35,1 %	26,3 %	0,0 %		b*	53,6 %	7,1 %	39,3 %	1,8 %
c*	86,0 %	5,3 %	8,8 %	0,0 %		c*	76,4 %	5,5 %	18,2 %	3,5 %
d*	64,9 %	21,1 %	14,0 %	0,0 %		d	19,6 %	62,5 %	17,9 %	1,8 %
e*	82,5 %	7,0 %	10,5 %	0,0 %		e	28,6 %	23,2 %	48,2 %	1,8 %
f*	80,7 %	7,0 %	12,3 %	0,0 %		10a	39,3 %	41,1 %	19,6 %	1,8 %
g*	48,2 %	30,4 %	21,4 %	1,8 %		b*	36,8 %	21,1 %	42,1 %	0,0 %
h*	50,9 %	35,1 %	14,0 %	0,0 %		c*	61,4 %	14,0 %	24,6 %	0,0 %
4a*	93,0 %	0,0 %	7,0 %	0,0 %		d*	63,2 %	10,5 %	26,3 %	0,0 %
b*	75,4 %	19,3 %	5,3 %	0,0 %		e	42,1 %	22,8 %	35,1 %	0,0 %
c	70,2 %	17,5 %	12,3 %	0,0 %		11a	37,5 %	37,5 %	25,0 %	1,8 %
d	1,8 %	87,7 %	10,5 %	0,0 %		b*	45,6 %	24,6 %	29,8 %	0,0 %
5a*	66,7 %	12,3 %	21,1 %	0,0 %		c*	42,9 %	30,4 %	26,8 %	1,8 %
b*	71,9 %	8,8 %	19,3 %	0,0 %		d	70,4 %	16,7 %	13,0 %	5,3 %
c	64,9 %	8,8 %	26,3 %	0,0 %		12a	60,0 %	23,6 %	16,4 %	3,5 %
6a*	83,9 %	7,1 %	8,9 %	1,8 %		b	46,4 %	42,9 %	10,7 %	1,8 %
b*	56,1 %	21,1 %	22,8 %	0,0 %		c*	80,4 %	12,5 %	7,1 %	1,8 %
c*	80,4 %	7,1 %	12,5 %	1,8 %		d*	87,3 %	5,5 %	7,3 %	3,5 %
d*	68,4 %	17,5 %	14,0 %	0,0 %		13a	47,3 %	40,0 %	12,7 %	3,5 %
e*	66,7 %	21,1 %	12,3 %	0,0 %		b	72,7 %	7,3 %	20,0 %	3,5 %
f*	80,7 %	10,5 %	8,8 %	0,0 %		c*	87,3 %	5,5 %	7,3 %	3,5 %
g*	46,4 %	33,9 %	19,6 %	1,8 %		14a*	63,6 %	20,0 %	16,4 %	3,5 %
7a*	74,5 %	12,7 %	12,7 %	3,5 %		b*	70,4 %	13,0 %	16,7 %	5,3 %
b*	46,3 %	25,9 %	27,8 %	5,3 %		c*	87,3 %	9,1 %	3,6 %	3,5 %
c*	53,7 %	18,5 %	27,8 %	5,3 %		15a*	56,4 %	27,3 %	16,4 %	3,5 %
d*	60,0 %	21,8 %	18,2 %	3,5 %		b*	91,1 %	3,6 %	5,4 %	1,8 %
e*	66,1 %	16,1 %	17,9 %	1,8 %		c*	80,7 %	8,8 %	10,5 %	0,0 %
f	47,3 %	12,7 %	40,0 %	3,5 %		d	8,9 %	89,3 %	1,8 %	1,8 %
g	41,8 %	20,0 %	38,2 %	3,5 %		16a*	83,6 %	14,5 %	1,8 %	3,5 %
8a	10,9 %	81,8 %	7,3 %	3,5 %		b	41,8 %	30,9 %	27,3 %	3,5 %
b	7,3 %	83,6 %	9,1 %	3,5 %		c*	81,8 %	3,6 %	14,5 %	3,5 %
c*	82,1 %	7,1 %	10,7 %	1,8 %		d*	55,6 %	24,1 %	20,4 %	5,3 %
d*	83,9 %	12,5 %	3,6 %	1,8 %		e*	24,1 %	35,2 %	40,7 %	5,3 %
e	8,9 %	83,9 %	7,1 %	1,8 %						

Appendix B - Differences between question categories

The questions can be divided into six groups or categories that focus on a particular risk-related concept:

- Category 1: Q3-Q4, asset related (asset)
- Category 2: Q6-Q8, treatment related (treatment)
- Category 3: Q9-Q10, frequency measures related (frequency m.)
- Category 4: Q11-Q13, risk related (risk)
- Category 5: Q14, vulnerability related (vulnerability)
- Category 6: Q15-Q16, threat related (threat)

All the categories were compared to find statistical significant differences (P value below 0,05) and the results are provided in Table 3.

Table 3 – The results from Wilcoxon's test

	Z	Asymp. Sig. (2-tailed)
treatment - asset	-3,445(a)	0,001
frequency m. - asset	-4,757(a)	0,000
risk - asset	-1,991(a)	0,047
vulnerability - asset	-1,815(b)	0,070
threat - asset	-2,448(a)	0,014
frequency m. - treatment	-2,443(a)	0,015
risk - treatment	-1,180(b)	0,238
vulnerability - treatment	-3,819(b)	0,000
threat - treatment	-,533(b)	0,594
risk - frequency m.	-3,891(b)	0,000
vulnerability - frequency m.	-4,180(b)	0,000
threat - frequency m.	-2,567(b)	0,010
vulnerability - risk	-2,218(b)	0,027
threat - risk	-,699(a)	0,484
threat - vulnerability	-3,046(a)	0,002

Table 4 – Ranks

		N	Mean Rank	Sum of Ranks
treatment - asset	Negative Ranks	40(a)	31,50	1260,00
	Positive Ranks	17(b)	23,12	393,00
	Ties	0(c)		
	Total	57		
frequency m. - asset	Negative Ranks	45(d)	27,83	1252,50
	Positive Ranks	8(e)	22,31	178,50
	Ties	4(f)		
	Total	57		
risk - asset	Negative Ranks	40(g)	26,93	1077,00
	Positive Ranks	17(h)	33,88	576,00
	Ties	0(i)		
	Total	57		
vulnerability - asset	Negative Ranks	22(j)	24,18	532,00
	Positive Ranks	32(k)	29,78	953,00
	Ties	2(l)		
	Total	56		

threat - asset	Negative Ranks	37(m)	29,68	1098,00
	Positive Ranks	19(n)	26,21	498,00
	Ties	1(o)		
	Total	57		
frequency m. - treatment	Negative Ranks	38(p)	29,84	1134,00
	Positive Ranks	19(q)	27,32	519,00
	Ties	0(r)		
	Total	57		
risk - treatment	Negative Ranks	24(s)	28,25	678,00
	Positive Ranks	33(t)	29,55	975,00
	Ties	0(u)		
	Total	57		
vulnerability - treatment	Negative Ranks	14(v)	23,57	330,00
	Positive Ranks	42(w)	30,14	1266,00
	Ties	0(x)		
	Total	56		
threat - treatment	Negative Ranks	28(y)	27,13	759,50
	Positive Ranks	29(z)	30,81	893,50
	Ties	0(aa)		
	Total	57		
risk - frequency m.	Negative Ranks	14(bb)	24,07	337,00
	Positive Ranks	43(cc)	30,60	1316,00
	Ties	0(dd)		
	Total	57		
vulnerability - frequency m.	Negative Ranks	14(ee)	20,43	286,00
	Positive Ranks	42(ff)	31,19	1310,00
	Ties	0(gg)		
	Total	56		
threat - frequency m.	Negative Ranks	21(hh)	23,98	503,50
	Positive Ranks	36(ii)	31,93	1149,50
	Ties	0(jj)		
	Total	57		
vulnerability - risk	Negative Ranks	20(kk)	25,28	505,50
	Positive Ranks	35(ll)	29,56	1034,50
	Ties	1(mm)		
	Total	56		
threat - risk	Negative Ranks	31(nn)	29,50	914,50
	Positive Ranks	26(oo)	28,40	738,50
	Ties	0(pp)		
	Total	57		
threat - vulnerability	Negative Ranks	35(qq)	29,21	1022,50
	Positive Ranks	17(rr)	20,91	355,50
	Ties	4(ss)		
	Total	56		

a treatment < asset
b treatment > asset
c treatment = asset
d frequency m. < asset
e frequency m. > asset
f frequency m. = asset
g risk < asset
h risk > asset
i risk = asset
j vulnerability < asset
k vulnerability > asset
l vulnerability = asset
m threat < asset
n threat > asset
o threat = asset
p frequency m. < treatment
q frequency m. > treatment
r frequency m. = treatment

s risk < treatment
t risk > treatment
u risk = treatment
v vulnerability < treatment
w vulnerability > treatment
x vulnerability = treatment
y threat < treatment
z threat > treatment
aa threat = treatment

bb risk < frequency m.
cc risk > frequency m.
dd risk = frequency m.
ee vulnerability < frequency m.
ff vulnerability > frequency m.
gg vulnerability = frequency m.
hh threat < frequency m.
ii threat > frequency m.
jj threat = frequency m.

kk vulnerability < risk
ll vulnerability > risk
mm vulnerability = risk
nn threat < risk
oo threat > risk
pp threat = risk
qq threat < vulnerability
rr threat > vulnerability
ss threat = vulnerability

Appendix C - Differences in score between students and professionals

We tested the difference in correct, uncertain and wrong answers between the students and professionals to see if there was a statistical significant difference between the two groups.

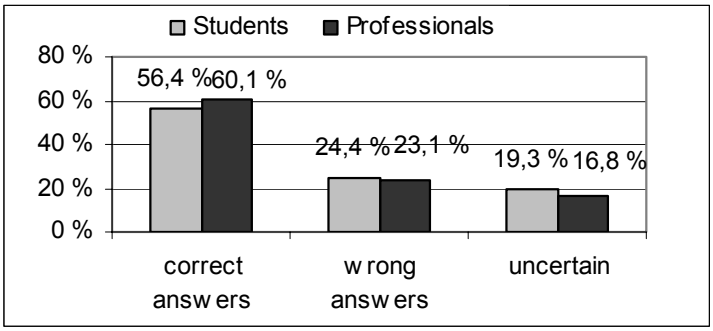


Figure 15 – Mean score: students vs. professionals

Table 5 – Group Statistics (group 1=students, 2= profs.)

	group	N	Mean	Std. Deviation	Std. Error Mean
correct	1	23	,5406	,11948	,02491
	2	34	,5979	,09961	,01708
incorrect	1	23	,2336	,07209	,01503
	2	34	,2305	,05847	,01003
uncertain	1	23	,1849	,13017	,02714
	2	34	,1673	,10874	,01865

The test shows that there is a statistical difference in mean score between the students and professionals for correct answers only (shaded area in Table 6). The test is two-tailed and therefore the p-value (Sig.) must be divided in two giving p = 0,27 and 0,32 which is below 0,05 significance level.

Table 6 – Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
correct	Equal variances assumed	,693	,409	-1,966	55	,054	-,05733	,02916	-,11577	,00110
	Equal variances not assumed			-1,898	41,443	,065	-,05733	,03021	-,11832	,00365
incorrect	Equal variances assumed	,193	,662	,182	55	,857	,00315	,01735	-,03162	,03792
	Equal variances not assumed			,174	40,578	,863	,00315	,01807	-,03335	,03965
uncertain	Equal variances assumed	,583	,448	,556	55	,580	,01769	,03180	-,04603	,08142

	Equal variances not assumed			,537	41,506	,594	,01769	,03293	-,04879	,08418

Appendix C

A Graphical Approach to Risk Modeling, Motivated by Empirical Investigations

A Graphical Approach to Risk Identification, Motivated by Empirical Investigations

Ida Hogganvik and Ketil Stølen

SINTEF ICT and Department of Informatics, University of Oslo
{iho, kst}@sintef.no

Abstract. We propose a graphical approach to identify, explain and document security threats and risk scenarios. Security risk analysis can be time consuming and expensive, hence, it is of great importance that involved parties quickly understand the risk picture. Risk analysis methods often make use of brainstorming sessions to identify risks, threats and vulnerabilities. These sessions involve system users, developers and decision makers. They typically often have completely different backgrounds and view the system from different perspectives. To facilitate communication and understanding among them, we have developed a graphical approach to document and explain the overall security risk picture. The development of the language and the guidelines for its use have been based on a combination of empirical investigations and experiences gathered from utilizing the approach in large scale industrial field trials. The investigations involved both professionals and students, and each field trial was in the order of 250 person hours.

1 Introduction

We have developed a graphical approach supporting the identification, communication and documentation of security threats and risk scenarios. The approach has been applied in several large industrial field trials and the major decisions regarding its underlying foundation, notation and guidelines are supported by empirical investigations. Our modeling approach originates from a UML [17] profile [15, 16], developed as a part of the EU funded research project CORAS (IST-2000-25031) [24] (http://coras.sourceforge.net). As a result of our work to satisfy the modeling needs in a security risk analysis, the language and its guidelines have evolved into a more specialized and refined approach. The language is meant to be used by the analyst during the security risk analysis, and has different purposes in each phase of the analysis. A normal risk analysis process often includes five phases: (1) context establishment, (2) risk identification, (3) risk estimation, (4) risk evaluation and (5) treatment identification [2]. In the following we refer to security risk analysis as security analysis. In the context establishment our language is used to specify the stakeholder(s) of the security analysis and their assets in *asset diagrams*. The purpose is to obtain a precise definition of what the valuable aspects of the target of analysis are, and which ones that are more important than others. From empirical investigations [5] and field trials we know that asset identification and valuation is very difficult, and that mistakes or inaccuracies made there may jeopardize the value of the whole security analysis.

O. Nierstrasz et al. (Eds.): MoDELS 2006, LNCS 4199, pp. 574 – 588, 2006.
© Springer-Verlag Berlin Heidelberg 2006

During risk identification we use *threat diagrams* to identify and document how vulnerabilities make it possible for threats to initiate unwanted incidents and which assets they affect. The threat diagrams give a clear and easily understandable overview of the risk picture and make it easier to see who or what the threat is, how the threat works (threat scenarios) and which vulnerabilities and assets that are involved.

The threat diagrams are used as input to the risk estimation phase, where unwanted incidents are assigned likelihood estimates and possible consequences. The likelihood estimation is often a difficult task, but illustrating the unwanted incidents in the correct context has proved very helpful in practice.

After the risk estimation the magnitude of each risk can be calculated on the basis of its likelihood and consequence, and modeled in *risk diagrams.* The risk diagrams specify which treats that initiate the different risks and exactly which assets they may harm. This risk representation is then compared to predefined risk tolerance levels to decide which ones that need treatments.

In the treatment identification, the threat diagrams that contain the non-tolerated risks are used as basis for *treatment diagrams*. In this phase the appropriate treatments are identified and modeled in treatment diagrams, where they point to the particular place where they should be implemented (e.g. pointing to a vulnerability). The resulting treatment diagrams can be seen as a plan for how to deal with the identified risks.

The contribution of our work is a revised, specialized, graphical language supporting risk identification based on structured brainstorming, and a comprehensive guideline for its use. Moreover, and in particular, we provide empirical support for the language's underlying, conceptual foundation and the main design decisions.

This paper is structured as follows: Sect. 2 introduces structured brainstorming in risk analysis. Sect. 3 explains the language's underlying conceptual foundation and Sect. 4 gives an example-driven introduction to our approach, including guidelines for modeling. Sect. 5 describes the major design decisions made during the development and the empirical investigations supporting these. Sect. 6 discusses the threats to validity for the empirical results, Sect. 7 presents related work and summarizes the main conclusions.

2 Structured Brainstorming for Risk Identification

A frequently used technique in security analysis, and in particular in risk identification, is so-called structured brainstorming (HazOp-analysis [18] is a kind of structured brainstorming). It may be understood as a structured "walk-through" of the target of analysis. The technique is not limited to a specific type of target, but can be used to assess anything from simple IT systems to large computerized process control systems or manual maintenance procedures in e.g. nuclear power plants. This does not mean that exactly the same technique can be applied directly to all kinds of targets, it has to be adapted to fit the target domain. The main idea of structured brainstorming is that a group of people with different competences and backgrounds will view the target from different perspectives and therefore identify more, and possibly other, risks than individuals or a more heterogeneous group. The input to a brainstorming session is various kinds of target models (e.g. UML models). The models are assessed in a stepwise and structured manner under the guidance of the security analysis leader. The identified risks are documented by an analysis secretary.

Deciding which roles should be present is part of tailoring the brainstorming technique towards the target in question. In some cases an expert is only participating in the part of the analysis where his or her expertise is needed. This makes it essential that the information is simple to communicate and comprehend.

3 The Conceptual Foundation

The first step in developing the language was constructing a conceptual model based on standardized security risk analysis terminology. We aimed to use the most intuitive and common interpretations. The conceptual model can be seen as a kind of abstract syntax for the language, and is shown in Fig. 1 using UML class diagram notation.

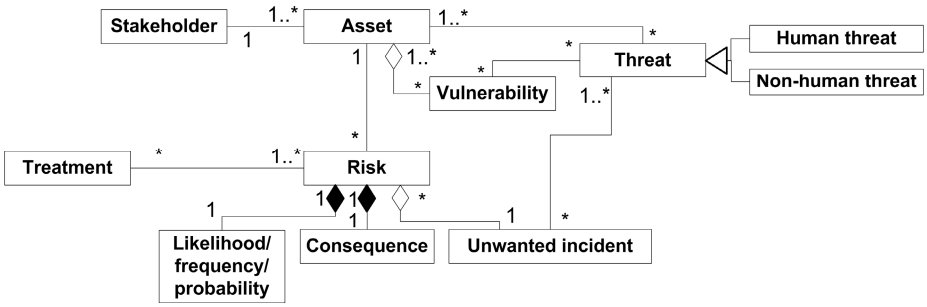


Fig. 1. The conceptual model

The conceptual model may be explained as follows: **stakeholders** are those people and organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity regarding the target of analysis [2]. An **asset** is something to which a stakeholder directly assigns value, and hence for which the stakeholder requires protection [3]. Assets are subject to **vulnerabilities,** which are weaknesses which can be exploited by one or more threats [9]. A **threat** is a potential cause of an unwanted incident [9]. An **unwanted incident** [9] is an event that may harm or reduce the value of assets and is something we want to prevent. A **risk** is the chance of something happening that will have an impact upon objectives (assets) [2]. Our model captures this interpretation by defining a risk to consist of an unwanted incident, a likelihood measure and a consequence. The abstract concept "risk", the more concrete "unwanted incident", and their respective relationships to "asset" require some explanation. In our definition, an unwanted incident that harms more than one asset gives rise to one unique risk for each asset it harms. This enables us to keep the consequences for different stakeholders separate, since an asset is always defined with respect to a single stakeholder. The level of risk is measured by a **risk value** [2] (e.g. low, medium, high or other scales) which is based upon the estimated **likelihood** (a general description of frequency or probability [2]) for the unwanted incident to happen and its **consequence** in terms of damage to an asset. A **treatment** is the selection and implementation of appropriate options for dealing with risk [2].

To validate our model and investigate the understanding of security analysis terminology we conducted an empirical study [5]. The 57 subjects included both students who had little or no knowledge of security analysis and more experienced professionals. We found that many of the terms as used in our conceptual model are well understood, even by people without training in security analysis.

The study showed that *human beings* are most commonly viewed as threats, followed by *events* (even if they are in fact initiated by a human). To increase the awareness of non-human threats, we decided to specify threat as either *human threat* and *non-human threat.* Our original specialization of frequency into probability and likelihood was not satisfactory. Both frequency and probability are measures that can be covered by likelihood. Likelihood was on the other hand found to be one of the least understood terms and we therefore decided to include all three concepts. For simplicity we only use "frequency" throughout the rest of the paper. On the question of what it is most common to treat, the subjects gave priority to vulnerability and thereafter risk. To make our model suitable for all treatment strategies we associate treatment with risk. This means that treatments can be directed towards vulnerabilities, threats, unwanted incidents or combinations of these.

4 Graphical Risk Modeling – An Example Driven Introduction

This section provides an example of a security analysis with guidelines for how and where the diagrams are made during the process. Fig. 2 gives the syntactical representation of the concepts from Sect.0, plus *initiate* (arrow), *treatment direction* (dashed arrow), *relationship* (line), *logical gates* (and & or) and *region* (used to structure the models). The symbols are a revised version of the ones used in Sect. 5.

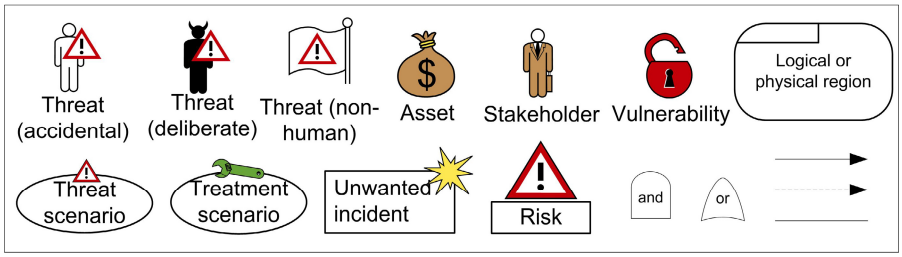


Fig. 2. The graphical representation of the main concepts

The target of analysis in our example is a web-based application which communicates confidential information between an insurance company and its customers. The development project is expensive and prestigious to the company, but the governmental data inspectorate is concerned about the level of privacy of the data provided by the service.

1 - Context establishment: The purpose of the context establishment is to characterize the target of the analysis and its environment. The web application is represented as a logical region (inspired by [20]) with two independent stakeholders (Fig. 3). The

stakeholders value assets differently: the governmental data inspectorate has "GDI1.data privacy" as its main asset, while the company management has identified four assets, ordered according to value as follows: "CM1.company brand & reputation", "CM2.data privacy", "CM3.application availability" and finally "CM4.application interface usability".

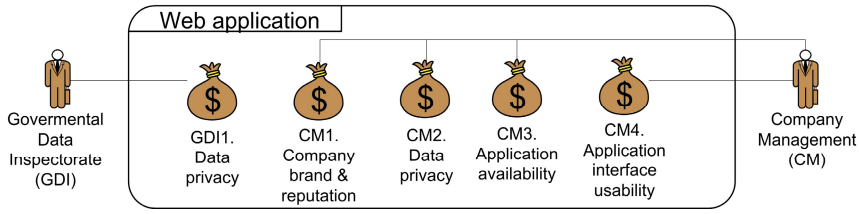


Fig. 3. Asset diagram

Modeling guideline for asset diagrams:

1. Draw a region that logically or physically represents the target of analysis.
2. Place the assets within the region, numbered according to its importance to the stakeholder, and with a reference to its stakeholder.
3. Associate the stakeholders with their assets.

2 - Risk identification: In the risk identification phase the security analysis leader and the brainstorming participants must find answers to questions like: *what are you most concerned about with respect to your assets?* (threat scenarios and unwanted incidents), *who/what initiates these*? (threats), *what makes this possible*? (vulnerabilities). This information is modeled in *threat diagrams*. Consider the threat diagram in Fig. 4. This diagram focuses on network related threat scenarios. The asset "CM4.application interface usability" is not harmed by this kind of incidents and therefore left out. The stakeholders are concerned about the unwanted incidents: "disclosure of data", "corruption of data" and "unavailability of application". Both threats are considered to cause incidents accidentally and are therefore modeled in the same diagram. We now explain one chain of events from the initiation caused by a threat to the left, to the impact on an asset to the right[1]: "IT-infrastructure" may first use "hardware failure" to make the server crash, and second use the "poor backup solution" to corrupt data and make the application unavailable.

Modeling guideline for threat diagrams:

1. Use the region from the asset diagram and add more regions if useful.
2. Model different kinds of threats in separate diagrams. E.g. deliberate sabotage in one diagram, mistakes in an other, environmental in a third etc. (classification from [9]). This makes it easier to generalize over the risks, e.g. "these risks are caused by deliberate intruders" or "these risks are caused by human errors".
3. Threats are placed to the left in the diagram.

[1] Disregard the frequency and consequence information, this is added later in risk estimation.

4. Assets are listed to the right, outside the region.
5. Unwanted incidents are placed within the region with relations to assets they impact.
6. Assets that are not harmed by any incidents are removed from the diagram.
7. Add threat scenarios between the threats and the unwanted incidents in the same order as they occur in real time (i.e. in a logical sequence).
8. Insert the vulnerabilities before the threat scenario or unwanted incident they lead to. E.g.: "poor backup solution" is placed before the threat scenario "application database fails to switch to backup solution".

3-Risk estimation: The threat diagrams are input to the risk estimation where threat scenarios and unwanted incidents are assigned frequencies and consequences. In Fig. 4 the final frequency for "corruption of data" is based on the frequencies of the two threat scenarios "application servers malfunctioning" and "application database fails to switch to backup solution". We here use the consequence scale: large (L), medium (M) and small (S). In a full security analysis the scale would be mapped to what the client considers to be large, medium and small reductions of asset value (e.g. 10% customer loss, 2 hours unavailability, 10.000$ etc.).

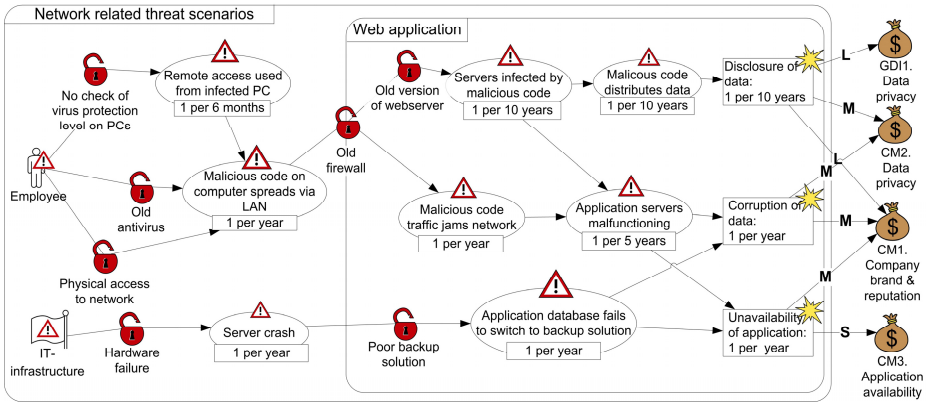


Fig. 4. Threat diagram (after risk estimation)

Modeling guideline for risk estimation:

1. Add frequency estimates to the threat scenarios.
2. Add frequency estimates to the unwanted incidents, based on the threat scenarios.
3. Annotate the unwanted incident-asset relations with consequences.

4 - Risk evaluation: On the basis of the risk estimation we model the resulting risks with their associated risk values in a risk diagram. Risk diagrams help the stakeholders to get an overview and evaluate which risks that need treatments. The example in Fig. 5 presents the threats and the risks they represent against the two most important assets in our example ("GDI1.data privacy" and "CM1.company brand & reputation"). We use a short hand notation where "Disclosure of data" in reality represents two risks with value = "major" (shown with associations to two assets).

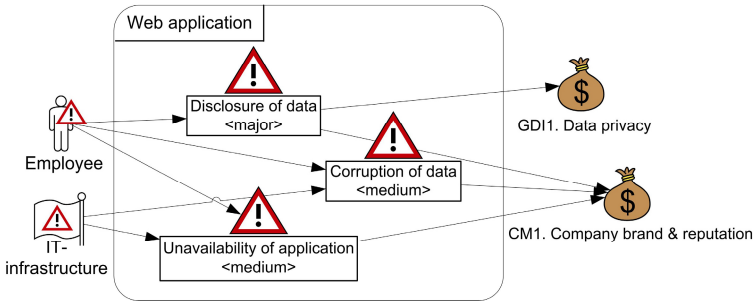


Fig. 5. Risk diagram (for the two most important assets)

Modeling guideline for risk diagrams:

1. Use the threat diagram and annotate all relations between unwanted incidents and assets with the risk symbol, showing a short risk description and the risk value (similar risks may be grouped as in the short hand notation shown in Fig. 5).
2. Split the risk diagrams into several diagrams according to risk value or asset importance (show all intolerable risks, all risks for specific assets etc.).
3. Remove threat scenarios and vulnerabilities, but keep the relations between the threats and the unwanted incidents.

5 - Treatment identification: The threat diagrams are also used as basis for treatment diagrams and extended with treatment options. The treatments must then be assessed to see whether they bring the risk value to an acceptable level. This makes it possible to optimize the treatment strategy according to a cost/benefit analysis. Fig. 6 shows the risks for the two most important assets. The proposed treatments in our example, "upgrade server", "upgrade backup solution" and "limit access to the network", are directed toward three vulnerabilities. Since treatments often address vulnerabilities,

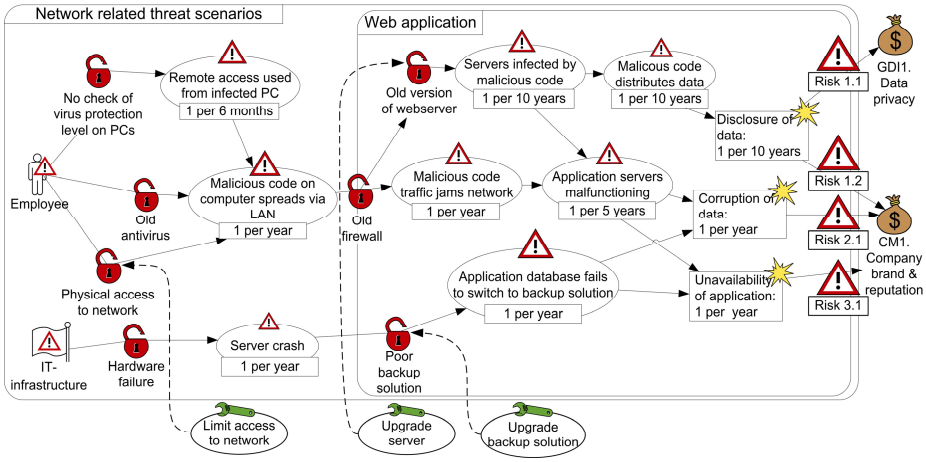


Fig. 6. Treatment diagram for: GCI1.data privacy and CM1.company brand & reputation

one may use the analogy “to close the padlock”, - meaning closing the specific pathway through the graph.

Modeling guideline for treatment diagrams:
1. Use the threat diagrams as a basis and annotate all arrows from unwanted incidents to assets with risk icons.
2. If the diagram becomes complex, split it like the risk diagrams.
3. Annotate the diagram with treatments, pointing to where they will be applied.

5 Design Decisions

Designing a language and its guidelines requires many design decisions. In the following we describe our design choices based on experiences from four major field trials and the results from two empirical investigations.

5.1 Field Trial Experiences

The field trials were carried out within the setting of the research project SECURIS (152839/220). Each security analysis required about 250 person hours from the analysis team and 50-100 hours from the stakeholder. The clients and scopes for the analyses were:

- Vessel classification company: A web based information sharing service between customer and service provider.
- Telecom company: Mobile access for employees to e-mail, calendar and contacts.
- Energy company: A control and supervisory system for power grid lines.
- Metal production company: A web based control and supervisory system for metal production.

The participants (the analysis team as well as the representatives of the stakeholders) were requested to evaluate the use of graphical risk modeling after each session and their feedback was:

- the use of the graphical models during the analysis made it easier to actively involve the participants and helped ensure an effective communication between the analysis team and the participants.
- the participants found the notation itself easy to understand and remember. It was considered to be a good way of visualizing threat scenarios and very suitable for presentations. According to one of the participants this type of visualization emphasizes the “message” or the purpose of the analysis.
- one of the main benefits was how the language helped specifying the relations between threats and the chain of events they may cause, the various states of the target, and potential incidents. The modeling method made the participants more conscious of the target of analysis and its risks by representing threats and vulnerabilities more explicitly than “just talking” about them.
- the language provides an opportunity to document cause-consequence relations in a precise and detailed manner.

The participants emphasized that they need a proper introduction to the notation and sufficient time to understand the information that is presented to them. One should limit the amount of information in one diagram, and rather split it according to threat type, scenario type or asset type. It is also important to strive towards correctness from the very beginning. This means that the participants must be involved early, enabling them to adjust the diagram as they find appropriate.

The diagrams capture information gathered during brainstorming sessions and one of the main concerns of the participants was whether or not the diagrams would be complete. To help overcome this we recommend utilizing check lists to ensure that all important aspects are covered. Any changes to the diagrams suggested during the brainstorming session represent important information. Their rationale should be captured by the analysis secretary, and the diagrams should be updated real time.

5.2 The Graphical Icons Experiment

Graphical icons are often seen as mere "decoration" just to make diagrams look nicer, but a major hypothesis in our work is that people unfamiliar with system modeling, may considerably benefit from carefully designed icons. In particular they may help the participants in a structured brainstorming to arrive at a common understanding of the security risk scenarios without wasting too much time. The original UML profile [15, 16] is characterized by its use of special graphical icons. To validate the effect of this we conducted an experiment which compared the usefulness of the UML profile icons compared to standard UML use case icons (Fig. 7) [5].

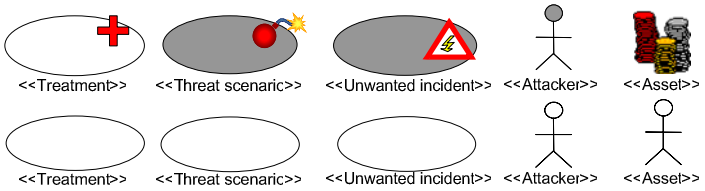


Fig. 7. The notation with and without special icons

In most of the tasks the group receiving normal UML icons had similar mean score to the group using profile icons, but when the time pressure increased the subjects with profile icons managed to complete more tasks than the other group. The positive effects of using graphical icons in models is also supported by [13]. The icons did on the other hand not significantly affect the correctness of interpretation of risk scenarios. Nevertheless, we decided to use special graphical icons to help the participants in a structured brainstorming to quickly get an overview of the risk scenarios.

5.3 Modeling Preferences Experiment

In the already mentioned empirical study of the conceptual foundation we found that some concepts were difficult to understand. Our hypothesis was that an explicit representation of these concepts in the models would mitigate this problem. To select the best representation we conducted an experiment involving 33 professionals. We used

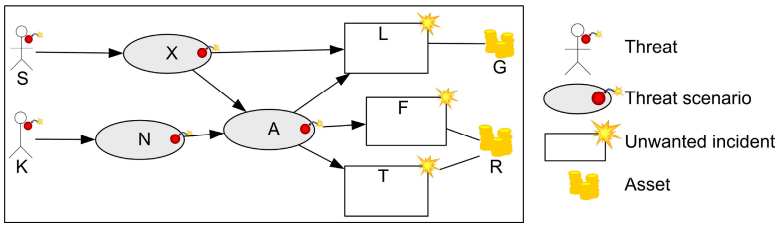


Fig. 8. The basic diagram (using the symbols from the previous version of the language)

variations of Fig. 8, which is a simplified version of a real threat diagram, to illustrate the various options. For a full description of the study we refer to [4].

Representing frequency: To help visualizing which paths that are most likely to be chosen by a threat, we investigated the effect of using different line types (**Fig. 9**). Line type (thick, thin, dashed etc.) has been suggested by [23] to represent aspects of associations between elements in a graph notation.

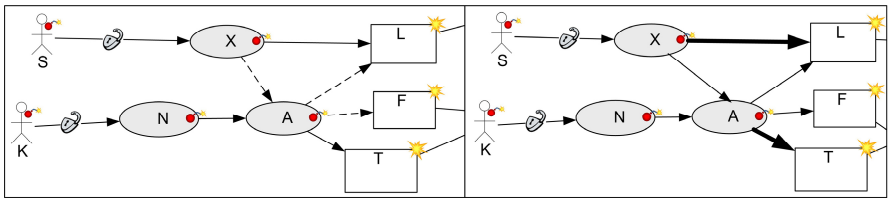


Fig. 9. Paths more likely than others in the graph

The result showed that neither of the line types is preferred for this purpose, possibly because a thick or dashed line does not convey a unique interpretation. During field trials we have found it more helpful to annotate threat scenarios with frequency estimates. The unwanted incident frequency is then estimated on the basis of the frequency estimates of the threat scenarios that cause the incident. This has reduced the need for assigning frequency information to the graph pathways.

Representing risk: The concept "risk" often leads to some confusion due to its abstract nature. We tried to make the risk notion less abstract by decorating the arrows between the unwanted incidents and the assets they harm with risk representations.

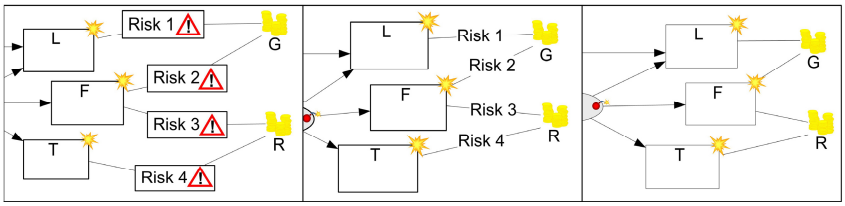


Fig. 10. "F": one incident, but two risks

The alternatives we tested included “text label and icon in box”, “text label only” and “association only” (Fig. 10).

The investigation showed that neither of the suggested representations is significantly preferred. Based on this we decided to specify risk in separate risk diagrams. In treatment diagrams, where we need to represent risks in their context, we use a text label and a risk icon. This alternative received the highest score, although the difference was not sufficiently large to be statistically significant from the two others.

In risk evaluation and treatment identification it may be useful to specify the magnitude of risks or unwanted incidents. We tested color, size and text label (Fig. 11), where the two first, according to [23], can be used to visualize magnitude. The result was clear, the participants preferred the text label version over the two other alternatives. A possible explanation is that a text label has unique interpretation, while size or color may have many interpretations.

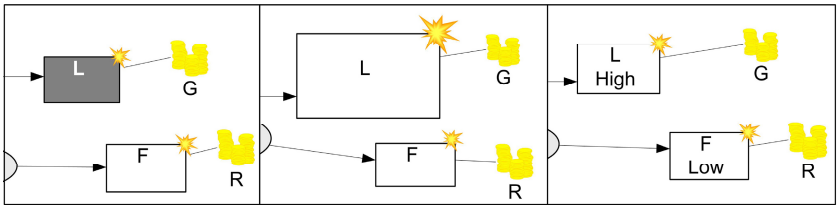


Fig. 11. The magnitude of an unwanted incident

Representing vulnerability: In the original UML profile [15, 16], vulnerabilities were only represented as attributes of the assets they were associated with. During field trials we experienced a definite need for representing vulnerabilities explicitly in the diagrams. We tested the UML profile representation against one that uses a vulnerability symbol (Fig. 12). The result showed that the alternative using chain locks as a vulnerability symbols was significantly preferred.

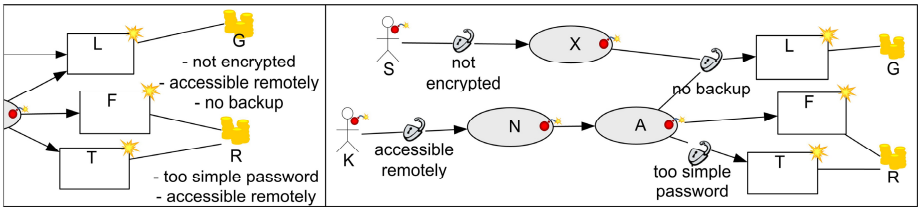


Fig. 12. Modeling vulnerabilities

The vulnerability symbols are helpful in describing threat scenarios and an excellent support in treatment identification. One of the participants in a field trial actually pointed to a vulnerability and claimed “*we’ve fixed this one now, so you should close the chain lock*”.

6 The Validity of the Empirical Results

The main threats to validity of the empirical results on which our approach builds, are summarized in this section. For full details we refer to [4-6].

The investigation of the underlying conceptual foundation was organized as a survey and the subjects included both master students in informatics and professionals. Despite their heterogeneous backgrounds, we believe that the results give a pretty good picture of the opinions of people working with system development in a general sense. The survey was formulated in natural language which always gives room for misinterpretations; some questions were discarded from statistical analysis due to this.

The four major field trials have given us a genuine opportunity to make the modeling guidelines appropriate for actual and realistic modeling situations. One of the main challenges we faced was how to optimize the structure of the diagrams. In some cases it was feasible to structure according to the kind of threat or incident, while in others we structured according to work processes in the target of analysis. This problem must be addressed early in the risk analysis process when the client specifies the acceptable risk level. The other main issue was how to capture changes and comments that arise during the brainstorming sessions. The solution seems to be modeling "on-the-fly". Updating and modifying diagrams real time can be challenging, but with the appropriate tool-support it can be overcome.

The subjects that participated in the icon experiment were master students in informatics. They had been introduced to the UML profile icons in a lesson focusing on the principles of risk modeling. This gave the subjects that received profile icons an advantage, but we believe it was too small to make a real difference.

In the modeling preferences investigation, the various alternatives were tested on a population with mainly technical background. This kind of background is similar to the background of most of the participants in our field trials and therefore considered to be a representative population.

The diagrams we used were a compromise between realistic diagrams and naive notation-only tests. This is a weakness that is difficult to avoid, and we will therefore validate our findings by testing them in real threat diagrams in our next field trial.

In general, humans who get involved in a security analysis are rarely familiar with the concepts or the process used. Concepts that students find intuitive to understand and model are probably intuitive also to professionals. We therefore believe that our use of students in two of the three investigations had little effect on the validity of the results.

7 Conclusions and Related Work

Misuse cases [1, 21, 22] was an important source of inspiration in the development of the above mentioned UML profile. A misuse case is a kind of UML use case [10] which characterizes functionality that the system should *not* allow. The use case notation is often employed for high level specification of systems and considered to be one of the easiest understandable notations in UML. A misuse case can be defined as "a completed sequence of actions which results in loss for the organization or some specific stakeholder" [22]. To obtain more detailed specifications of the system, the

misuse cases can be further specialized into e.g. sequence- or activity diagrams. There are a number of security oriented extensions of UML, e.g. UMLSec [11] and SecureUML [14]. These and related approaches have however all been designed to capture security properties and security aspects at a more detailed level than our language. Moreover, their focus is not on brainstorming sessions as in our case.

Fault tree is a tree-notation used in fault tree analysis (FTA) [8]. The top node represents an unwanted incident, or failure, and the different events that may lead to the top event are modeled as branches of nodes, with the leaf node as the causing event. The probability of the top node can be calculated on the basis of the probabilities of the leaf nodes and the logical gates "and" and "or". Our threat diagrams often look a bit like fault trees, but may have more than one top node. Computing the top node probability of a fault tree requires precise quantitative input, which rarely is available for incidents caused by human errors or common cause failures. Our approach can model fault trees, but we also allow qualitative likelihood values as input.

Event tree analysis (ETA) [7] focuses on illustrating the consequences of an event and the probabilities of these. It is often used as an extension of fault trees to create "cause-consequence" diagrams. Event trees can to a large extent also be simulated in our notation.

Attack trees [19] aim to provide a formal and methodical way of describing the security of a system based on the attacks it may be exposed to. The notation uses a tree structure similar to fault trees, with the attack goal as the top node and different ways of achieving the goal as leaf nodes. Our approach supports this way of modeling, but facilitates in addition the specification of the attack initiators (threats) and the harm caused by the attack (damage to assets).

The Riskit method [12] includes a risk modeling technique that makes it possible to specify factors that may influence a software development project. It has similarities to our approach, but targets project risks and therefore utilizes different concepts and definitions and has not the special focus on security risks.

In security risk analysis various types of brainstorming techniques are widely used to identify risks. The participants use their competence to discover relevant threats, vulnerabilities and unwanted incidents that the analysis object may be subject to. The overall idea of brainstorming is that the brainstorming group will identify more risks together than the same individuals working separately. The participants have extensive knowledge of the target of analysis, but often within different areas. The challenge is to make them understand and discuss the overall risk picture without being hampered by misunderstandings and communication problems. Since a brainstorming session may be exhausting, we need techniques and guidelines that make the session as efficient and effective as possible. The findings must also be documented in a precise and comprehensive manner.

We have developed a graphical approach to risk identification to meet these requirements. It captures the complete risk picture, including assets, stakeholders and how vulnerabilities make security threats able to harm the assets. The graphical notation is both simple and expressive, and does not require extensive knowledge of modeling. The language originates from a UML profile for risk assessment [15, 16]. Through application in field trials and empirical studies it has evolved towards a more refined and specialized language. To our knowledge our approach is the only one that

combines system modeling techniques with risk modeling techniques for use in structured brainstorming within the security domain.

A major presumption for an intuitive and understandable language is the underlying conceptual foundation, which defines the various concepts in the language and how they relate to each other. We have selected our definitions from well recognized standards within security and risk analysis [2, 3, 9]. To ensure that the best definitions of concepts and relations were chosen, we investigated the alternatives empirically. We found that risk analysis terminology in general is quite well understood. In general, those concepts that caused most uncertainty were the concepts that are least used in daily language. As an example, frequency is less used in the daily language than consequence and was one of the terms many of the respondents had trouble understanding. The subjects tend to care less about the frequency than the consequence of a risk, even though a high frequency risk with low consequence over time may cost the same as a large consequence risk.

The language has been utilized in four large industrial field trials. We have experienced that the diagrams facilitate active involvement of the participants in the brainstorming sessions, and they are very helpful in visualizing the risk picture. According to the participants, the diagrams explicitly illustrate the threats and vulnerabilities in a way that makes it easy to see the relations and precisely define the risk consequences.

The notational design decisions are supported by two empirical studies. The first one investigated the usefulness of special graphical icons contra traditional UML use case icons. The result showed that the group that received diagrams with special icons was able to conclude faster. This and similar supporting studies convinced us to use graphical icons in our language. The second study tested different modeling approaches on a group of professionals. From this we conclude amongst others that text labels are preferable over visualization mechanisms like size, color or line type. For our language this means that we mark a major risk with the text label “major”, rather than representing it with a large risk icon.

The final language description will include a formal semantics and a more detailed modeling guideline. The language is currently being implemented in the CORAS’ risk analysis tool (v3.0) which is planned for release autumn 2006.

Acknowledgments. The research on which this paper reports has partly been funded by the Research Council of Norway project SECURIS (152839/220). The authors especially thank Mass Soldal Lund for his valuable input. We also thank the SECURIS analysis team: Fredrik Seehusen, Bjørnar Solhaug, Iselin Engan, Gyrd Brændeland and Fredrik Vraalsen.

References

1. Alexander, I., *Misuse cases: Use cases with hostile intent.* IEEE Software, 2003. **20**(1), pp. 58-66.
2. AS/NZS4360, *Australian/New Zealand Standard for Risk Management*. 2004, Standards Australia/Standards New Zealand.
3. HB231, *Information security risk management guidelines*. 2004, Standards Australia/Standards New Zealand.

4. Hogganvik, I. and K. Stølen, *Investigating Preferences in Graphical Risk Modeling* (Tech. report SINTEF A57). SINTEF ICT, 2006. http://heim.ifi.uio.no/~ketils/securis/the-securis-dissemination.htm.
5. Hogganvik, I. and K. Stølen. *On the Comprehension of Security Risk Scenarios*. *In Proc. of 13th Int. Workshop on Program Comprehension (IWPC'05)*. 2005, pp. 115-124.
6. Hogganvik, I. and K. Stølen. *Risk Analysis Terminology for IT-systems: does it match intuition? In Proc. of Int. Symposium on Empirical Software Engineering (ISESE'05)*. 2005, pp. 13-23.
7. IEC60300-3-9, *Event Tree Analysis in Dependability management - Part 3: Application guide - Section 9: Risk analysis of technological systems.* 1995.
8. IEC61025, *Fault Tree Analysis (FTA)*. 1990.
9. ISO/IEC13335, *Information technology - Guidelines for management of IT Security*. 1996-2000.
10. Jacobson, I., et al., *Object-Oriented Software Engineering: A Use Case Driven Approach.* 1992: Addison-Wesley.
11. Jürjens, J., *Secure Systems Development with UML*. 2005: Springer.
12. Kontio, J., *Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation*. PhD thesis, Dept. of Computer Science and Engineering, Helsinki University of Technology, 2001.
13. Kuzniarz, L., M. Staron, and C. Wohlin. *An Empirical Study on Using Stereotypes to Improve Understanding of UML Models*. *In Proc. of 12th Int. Workshop on Program Comprehension (IWPC'04)*. 2004, pp. 14-23.
14. Lodderstedt, T., D. Basin, and J. Doser. *SecureUML: A UML-Based Modeling Language for Model-Driven Security*. *In Proc. of UML'02*. 2002, pp. 426-441.
15. Lund, M.S., et al., *UML profile for security assessment* (Tech. report STF40 A03066). SINTEF ICT, 2003.
16. OMG, *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*. 2005, Object Management Group.
17. OMG, *The Unified Modeling Language (UML) 2.0*. 2004.
18. Redmill, F., M. Chudleigh, and J. Catmur, *HAZOP and Software HAZOP*. 1999: Wiley.
19. Schneier, B., *Attack trees: Modeling security threats.* Dr. Dobb's Journal, 1999. **24**(12), pp. 21-29.
20. Seehusen, F. and K. Stølen. *Graphical specification of dynamic network structure. In Proc. of 7th Int. Conference on Enterprise Information Systems (ICEIS'05)*. 2005, pp. p.203-209.
21. Sindre, G. and A.L. Opdahl. *Eliciting Security Requirements by Misuse Cases*. *In Proc. of TOOLS-PACIFIC*. 2000, pp. 120-131.
22. Sindre, G. and A.L. Opdahl. *Templates for Misuse Case Description*. *In Proc. of Workshop of Requirements Engineering: Foundation of Software Quality (REFSQ'01)*. 2001, pp. 125-136.
23. Ware, C., *Information Visualization: Perception for Design*. 2 ed. 2004: Elsevier.
24. Aagedal, J.Ø., et al. *Model-based risk assessment to improve enterprise security*. *In Proc. of Enterprise Distributed Object Communication (EDOC'02)*. 2002, pp. 51-64.

Appendix D

Structured Semantics for the CORAS Security Risk Modeling Language

SINTEF

SINTEF ICT

Address: P.O.Box 124, Blindern
0314 Oslo NORWAY
Location: Forskningsveien 1
0373 Oslo
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

Structured Semantics for the CORAS Security Risk Modelling Language

AUTHOR(S)

Heidi E. I. Dahl, Ida Hogganvik and Ketil Stølen

CLIENT(S)

n/a

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
A970	Unrestricted	n/a	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
Unrestricted	978-8214040494	SECURIS(152839/220) S3MS(IST-2006-027004) DIGIT (180052/S10)	34
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.)	CHECKED BY (NAME, SIGN.)
CORAS Semantics.doc		Ketil Stølen	Bjørnar Solhaug
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.)	
	2007-09-24	Bjørn Skjellaug, Forskningssjef	

ABSTRACT

The CORAS security risk modelling language is a graphical language customized for communication, documentation and analysis of security threat and risk scenarios. We present a textual syntax and a structured semantics for each of the five different types of CORAS diagrams, together with step-by-step instructions on how to translate a graphical diagram via the textual syntax into a readable paragraph of English prose, using the structured semantics.

This enables users of the CORAS language to easily extract the precise meaning of a given diagram. The semantics is modular in the sense that the semantics of a diagram can be deduced from the semantics of its constituents.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Security risk analysis	Sikkerhetsanalyse
GROUP 2	Modelling language	Modelleringsspråk
SELECTED BY AUTHOR	Semantics	Semantikk

Structured Semantics for the CORAS Security Risk Modelling Language

Heidi E. I. Dahl, Ida Hogganvik and Ketil Stølen

September 24, 2007

Abstract

The CORAS security risk modelling language is a graphical language customized for communication, documentation and analysis of security threat and risk scenarios. We present a textual syntax and a structured semantics for each of the five different types of CORAS diagrams, together with step-by-step instructions on how to translate a graphical diagram via the textual syntax into a readable paragraph of English prose, using the structured semantics. This enables users of the CORAS language to easily extract the precise meaning of a given diagram. The semantics is modular in the sense that the semantics of a diagram can be deduced from the semantics of its constituents.

Contents

1 Introduction **3**

2 The CORAS Language **4**

3 The Structured Semantics **5**

4 Asset Diagrams **6**
- 4.1 Textual Syntax of Asset Diagrams . . . 7
- 4.2 Translation from Graphical to Textual Syntax . . . 7
- 4.3 Translation from the Textual Syntax to English . . . 8
- 4.4 Example . . . 9

5 Threat Diagrams **10**
- 5.1 Textual Syntax of Threat Diagrams . . . 11
- 5.2 Translation from Graphical to Textual Syntax . . . 12
- 5.3 Translation from the Textual Syntax to English . . . 13
- 5.4 Example . . . 15

6 Risk Diagrams **18**
- 6.1 Textual Syntax of Risk Diagrams . . . 19
- 6.2 Translation from Graphical to Textual Syntax . . . 19
- 6.3 Translation from the Textual Syntax to English . . . 20
- 6.4 Example . . . 21

7 Treatment Diagrams **23**
7.1 Textual Syntax of Treatment Diagrams 24
7.2 Translation from Graphical to Textual Syntax 25
7.3 Translation from the Textual Syntax to English 26
7.4 Example 27

8 Treatment Overview Diagrams **28**
8.1 Textual Syntax of Treatment Overview Diagrams 29
8.2 Translation from Graphical to Textual Syntax 30
8.3 Translation from the Textual Syntax to English 31
8.4 Example 31

9 Conclusion **32**

References **34**

1 Introduction

CORAS is a method for security risk analysis [4]. It comes with a specialized language for communication, documentation and analysis of security threat and risk scenarios. The language was originally defined as a UML [15] profile [14, 16], and has later been customized and refined in several aspects, based on experiences from industrial case studies, and by empirical investigations documented in [5], [6] and [7].

The CORAS language is in particular intended to support brainstorming sessions used to identify and estimate security risks. Such brainstorming sessions are characterized by the involvement of people with thorough knowledge of specific, but only partly overlapping aspects of the target of analysis. Typical participants are the intended users of the target, its designers, developers, and relevant decision makers. These people normally have quite different backgrounds and it may be difficult for the analysts to make them work well together as a group. Our experiences indicate that the CORAS language improves both the efficiency of the analysis process and the quality of the results.

We claim that our graphical approach to security risk modelling contributes to solving three issues related to security analysis:

- *How to facilitate communication in a group consisting of people with different backgrounds and competences:* Our aim has been to provide the participants with a means of communication that covers both technical and more high-level information, without being too complicated to understand. Offering a common basis for communication will hopefully reduce misunderstandings and thereby give a more correct risk picture.
- *How to estimate the likelihoods and consequences of identified risks:* In practice, reliable data on which this can be based is often not available. The participants must use their expert knowledge, experience and familiarity with the domain to estimate both the likelihoods and the consequences of incidents that might not have happened yet. Our aim has been to offer a structured, graphical risk picture to make the complexity more manageable. A graphical representation may illustrate who or what caused the incidents and the weaknesses in the system that made them possible.
- *How to document the security analysis in a comprehensible manner:* The findings of a security analysis constitute vital information not only to the participants in the analysis, but also to the organization as a whole. Our aim has been to define a documentation method that should be more or less self-explanatory, and not rely on extensive training to be understood.

Although we have aimed at making a language that is easily understandable, situations in which the intended meaning of a construct or an expression needs further explanation are bound to arise. The main contribution of this report is the definition of a structured semantics aiming to fulfil this need. The semantics takes an arbitrary CORAS diagram and delivers its intended meaning as a readable paragraph of English. It is structured in the sense that it comes with step-by-step instructions allowing the translation to be conducted automatically. The semantics has been developed to meet the following success criteria:

1. *The translation from CORAS diagrams to English should be modular.* If we add new relations and/or elements to a diagram we have already translated, the translation of the modified diagram is the union of the translation of the original diagram with the translation of the new relations and/or elements.
2. *The resulting paragraph should be understandable English.* The purpose of the translation is to provide a description, in English, of a CORAS diagram, in order to communicate the meaning of the diagram to those not familiar with the intended meaning of the various elements and relations of the CORAS language.

3. *The translation should be easy to perform.* Anyone, even someone unacquainted with CORAS diagrams, should be able to translate a CORAS diagram into English.

4. *The translation should be possible to automate.* Automatic translation is a feature that will be implemented in the CORAS tool in the future (see `http://coras.sourceforge.net` for downloads and documentation).

5. *It should be possible to translate inconsistent diagrams, and the translation should enable the user to identify inconsistencies.* Inconsistent diagrams should still be possible to translate, and the resulting paragraph in English should be sufficiently clear to allow the user to identify the cause of the inconsistency.

The next section introduces the different kinds of CORAS diagrams. Section 3 presents the notation and naming conventions used in the textual syntax and structured semantics. Sections 4 - 8 present the syntax and structured semantics of each of the five types of CORAS diagrams. Each of these sections is self-contained and includes instructions on how to translate an arbitrary syntactically correct diagram into English. An example diagram is given as a presentation of the graphical syntax of each diagram, which contains all possible relations. However, in order to keep the examples to a manageable size, not all permutations of optional elements such as likelihoods and treatment categories have been included. In the cases where one CORAS diagram type is based on another, we still give a complete definition of its textual syntax, translation rules from graphical to textual syntax, and from the textual syntax into English. At the end of each section we present an example of a translation.

2 The CORAS Language

The CORAS language originates from a UML profile developed as a part of the EU funded research project CORAS (IST-2000-25031) [1] (`http://coras.sourceforge.net`). As a result of our work to satisfy the modelling needs in a security risk analysis, the language and its guidelines have evolved into a more specialized and refined approach. The language is meant to support the analyst during the security risk analysis, and serves different purposes in each phase of the analysis. A security risk analysis is normally structured into five phases: (1) context establishment, (2) risk identification, (3) risk estimation, (4) risk evaluation and (5) treatment identification [3].

In the context establishment we employ *assets overview diagrams* to specify the parties of the security analysis and their assets. The purpose is to obtain a precise definition of what the valuable aspects of the target of analysis are, and which are the most important. From empirical investigations [5] and field trials we know that asset identification and valuation is very difficult, and that mistakes or inaccuracies made there may jeopardize the value of the whole security analysis.

During risk identification we use *threat diagrams* to identify and document how vulnerabilities may be exploited by threats to initiate unwanted incidents, and which assets the unwanted incidents affect. The threat diagrams give a clear and easily understandable overview of the risk picture and make it easier to see who or what the threat is, how the threat works (threat scenarios) and which vulnerabilities and assets they involve.

The threat diagrams are used as input to the risk estimation phase, where unwanted incidents are assigned likelihood estimates and possible consequences. The likelihood estimation is often a difficult task, but illustrating the unwanted incidents in the correct context has proved very helpful in practice.

After the risk estimation, the magnitude of each risk can be calculated on the basis of its likelihood and consequence, and modelled in *risk diagrams.* The risk diagrams specify which threats initiate the different risks, and exactly which assets they may harm. This risk representation is then compared to predefined risk tolerance levels to decide which ones need treatments.

In the treatment identification, the threat diagrams containing the risks that cannot be tolerated are used as basis for treatment identification. In this phase the appropriate treatments are identified and modelled in *treatment diagrams*. The resulting treatment diagrams can be seen as a plan for how to deal with the identified risks.

Communicating the results of an analysis in such a way that they are well understood by decision makers can be challenging. The CORAS language supports this by offering *treatment overview diagrams*. Treatment overview diagrams may for example be used to provide a high level summary when presenting the main findings from an analysis.

To summarize, the CORAS language consists of five different kinds of diagrams: asset diagrams, threat diagrams, risk diagrams, treatment diagrams and treatment overview diagrams. Their basic building blocks are presented in Figure 1.

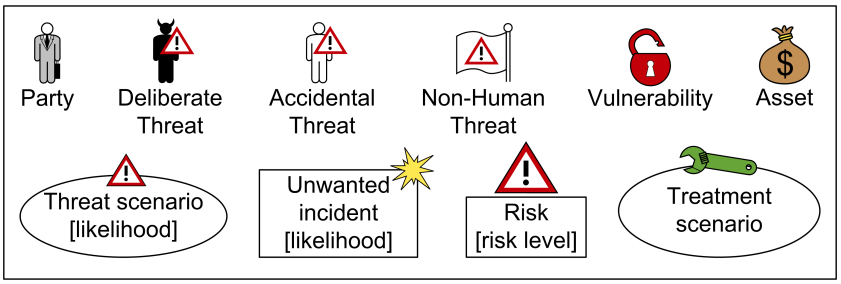


Figure 1: Basic building blocks of the CORAS diagrams

3 The Structured Semantics

The structured semantics for the CORAS language is divided into two separate steps:

(A) The translation of a diagram into its textual syntax, and

(B) The translation of the textual syntax into its meaning as a paragraph in English.

Hence, the semantics enables the user of CORAS to extract the meaning of an arbitrary CORAS diagram by applying first (A), then (B).

Both these steps, and therefore the structured semantics, are modular: a diagram is translated relation by relation, vertex by vertex.

To characterize the textual syntax for the CORAS language, we use ISO's standardized Extended BNF notation [10]. The vertical bar _|_ represents options, braces {_} (respectively {_}$^-$) means an ordered sequence of zero (respectively one) or more repetitions of the enclosed element, and square brackets [_] denotes optional features. The terminal operators are surrounded by quotes: '_'.

To improve readability, we do not resolve all the EBNF definitions completely into terminal operators, but stop at the level of *identifier*, *linguistic term* and *numerical value*. An identifier is a name or a natural language description of an element, i.e. a finite sequence of upper- and lower-case letters, numbers, spaces and punctuation marks. Its precise definition in EBNF is

$$\begin{aligned} \textit{identifier} &= \{\textit{upper-case letter}|\textit{lower-case letter}|\textit{digit}|\text{`\ '}|\textit{punctuation mark}\}^- ; \\ \textit{uppercase letter} &= \text{`A'}|\text{`B'}|\text{`C'}|\ldots \\ \textit{lowercase letter} &= \text{`a'}|\text{`b'}|\text{`c'}|\ldots \\ \textit{punctuation mark} &= \text{`.'}|\text{`,'}|\text{`:'}|\ldots \\ &\ldots \end{aligned}$$

Linguistic terms are identifiers which are members of a totally ordered set, for example likelihoods given as 'high', 'medium' and 'low'. Numerical values are just that: numbers allowing precise calculations of likelihoods, consequences and risk values.

In the translation rules, we use $\Rightarrow$ to denote the translation from graphical to textual syntax.

The translation rules for the vertices and annotations of the diagrams are given by the naming conventions in Table 1.

Vertex	**Instance**
party	p
asset	a
deliberate threat	dt
accidental threat	at
non-human threat	nht
threat scenario	ts
unwanted incident	ui
risk	r
treatment scenario	trs

Annotation	**Instance**
vulnerability	$v = \{v\} = V_1$
vulnerability set	$V_n = \{v_1, \ldots, v_n\}$
likelihood	l
consequence	c
risk value	rv
risk function	rf

Table 1: Naming conventions

When there are more than one instance of a type, we use subscripts, e.g. dt_1, dt_2 for two deliberate threats. The translation from the textual syntax into English is defined by a function $[\![_]\!]$ on textual expressions. We use _ := _ to denote "defined as".

4 Asset Diagrams

Asset diagrams are used to give an overview of the assets within the scope of the security risk analysis, and to show how harm to one asset may affect other assets. They also present the parties involved in the analysis, and their interest in the assets. The asset diagram has two basic building blocks, parties and assets, shown in Figure 2. We refer to the parties and assets of an asset diagram

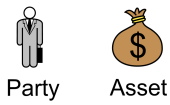


Figure 2: Basic building blocks of asset diagrams

as its *vertices*. The vertices may be connected by two kinds of ordered relations: the *protect* and the *indirect harm* relations. The protect relation displays the relationship inherent in the definition of an asset; i.e. that the asset is something whose value a party wants to protect. An important part of this is the level of risk that a party is willing to accept with respect to the asset in question, therefore we have the option to annotate the relation with the maximum acceptable risk level. The indirect harm relation captures dependency between two assets, in the sense that harm to the first asset may indirectly harm the other.

To summarize, asset diagrams may contain protect relations between parties and assets, and indirect harm relations between assets. Figure 3 shows the graphical syntax of asset diagrams.

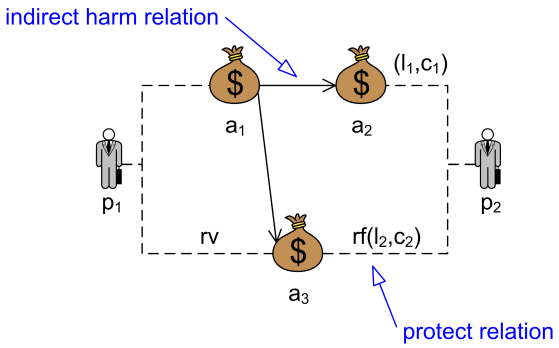


Figure 3: Graphical syntax of asset diagrams

4.1 Textual Syntax of Asset Diagrams

$$
\begin{aligned}
\mathit{diagram} &= (\{\mathit{vertex}\}^{-}, \{\mathit{relation}\});\\
\mathit{vertex} &= \mathit{party} \mid \mathit{asset};\\
\mathit{relation} &= \mathit{protect} \mid \mathit{indirect\ harm};\\
\mathit{protect} &= \mathit{party} \overset{[\mathit{risk\ level}]}{\cdots} \mathit{asset};\\
\mathit{indirect\ harm} &= \mathit{asset} \rightarrow \mathit{asset};\\
\mathit{party} &= \mathit{identifier};\\
\mathit{asset} &= \mathit{identifier};\\
\mathit{risk\ level} &= \mathit{risk\ value} \mid \mathit{risk\ function}(\mathit{likelihood}, \mathit{consequence}) \mid\\
&\quad (\mathit{likelihood}, \mathit{consequence});\\
\mathit{risk\ value} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{risk\ function} &= \mathit{identifier};\\
\mathit{likelihood} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{consequence} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};
\end{aligned}
$$

4.2 Translation from Graphical to Textual Syntax

We translate an asset diagram into the textual syntax schematically, vertex by vertex and relation by relation. The set of textual expressions we obtain is the translation of the complete diagram.

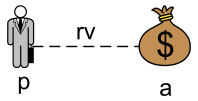
4.2.1 Vertices

The translation of a vertex in the textual syntax is the label of the icon representing it in the graphical syntax. For example, the translation of the set of vertices in Figure 3 is the set

$$\{p_1, p_2, a_1, a_2, a_3\}.$$

Note that in this translation, the type of each node is preserved through the naming conventions from Table 1. When the naming convention is not used we need to type each vertex, for example in a type table, or decompose the diagram into three sets: sets of parties, sets of assets and sets of relations.

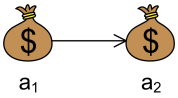
4.2.2 Protect relation



$\Rrightarrow\ p \overset{rv}{\cdots} a$

The translation is similar when the relation is annotated with a risk function or a likelihood and consequence pair instead of a risk value. In either case, replace rv with the appropriate annotation. If the protect relation is not annotated, remove rv altogether.

4.2.3 Indirect harm relation



$\Rrightarrow\ a_1 \rightarrow a_2$

4.3 Translation from the Textual Syntax to English

4.3.1 Diagram

A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0, m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation.

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

4.3.2 Vertices

$$[\![p]\!] := \boldsymbol{p} \text{ is a party.}$$
$$[\![a]\!] := \boldsymbol{a} \text{ is an asset.}$$

4.3.3 Protect relation

$$[\![p \cdots a]\!] := \boldsymbol{p} \text{ wants to protect the value of } \boldsymbol{a}.$$
$$[\![p \overset{rl}{\cdots} a]\!] := \boldsymbol{p} \text{ wants to protect the value of } \boldsymbol{a}, \text{ but accepts } [\![rl]\!] \text{ or less.}$$

The second expression is the translation of a protect relation annotated with a maximum acceptable risk level. The rl for risk level is replaced by either rv, $rf(l, c)$ or (l, c) depending on how the risk level is formulated.

4.3.4 Indirect harm relation

$$[\![a_1 \rightarrow a_2]\!] := \boldsymbol{a_2} \text{ is harmed indirectly via } \boldsymbol{a_1}.$$

4.3.5 Annotations

$$
\begin{aligned}
[\![\, rv \,]\!] &:= \text{risk value } \boldsymbol{rv} \\
[\![\, rf(l,c) \,]\!] &:= \text{risk function } \boldsymbol{rf} \text{ of } [\![\, (l,c) \,]\!] \\
[\![\, (l,c) \,]\!] &:= [\![\, l \,]\!] \text{ and } [\![\, c \,]\!] \\
[\![\, l \,]\!] &:= \text{likelihood } \boldsymbol{l} \\
[\![\, c \,]\!] &:= \text{consequence } \boldsymbol{c}
\end{aligned}
$$

4.4 Example

We illustrate the translation of asset diagrams with the following example:

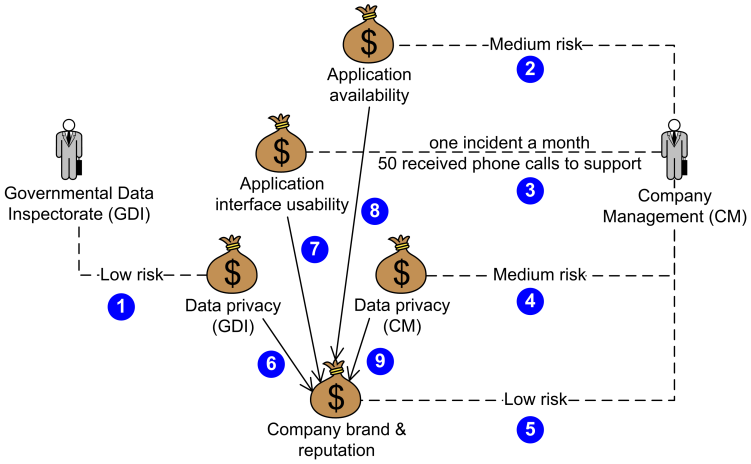


Figure 4: Asset diagram

First we translate the vertices of the diagram: the parties and assets.

1. ***Governmental Data Inspectorate (GDI)*** is a party.
2. ***Company Management (CM)*** is a party.
3. ***Application availability*** is an asset.
4. ***Application interface usability*** is an asset.
5. ***Data privacy (GDI)*** is an asset.
6. ***Data privacy (CM)*** is an asset.
7. ***Company brand & reputation*** is an asset.

We then translate the diagram relation by relation in the order suggested in Figure 4:

1. ***Governmental Data Inspectorate (GDI)*** wants to protect the value of ***Data privacy (GDI)***, but accepts risk value ***Low risk*** or less.
2. ***Company Management (CM)*** wants to protect the value of ***Application availability***, but accepts risk value ***Medium risk*** or less.

3. ***Company Management (CM)*** wants to protect the value of ***Application interface usability***, but accepts likelihood ***one incident a month*** and consequence ***50 received phone calls to support*** or less.
4. ***Company Management (CM)*** wants to protect the value of ***Data privacy (CM)***, but accepts risk value ***Medium risk*** or less.
5. ***Company Management (CM)*** wants to protect the value of ***Company brand & reputation***, but accepts risk value ***Low risk*** or less.
6. ***Company brand & reputation*** is harmed indirectly via ***Data privacy (GDI)***.
7. ***Company brand & reputation*** is harmed indirectly via ***Application interface usability***.
8. ***Company brand & reputation*** is harmed indirectly via ***Application availability***.
9. ***Company brand & reputation*** is harmed indirectly via ***Data privacy (CM)***.

We may choose to structure the sentences further to improve readability:

1. ***Governmental Data Inspectorate (GDI)*** wants to protect the value of ***Data privacy (GDI)***, but accepts risk value ***Low risk*** or less.

2. - 5. ***Company Management (CM)*** wants to protect the value of

- ***Application availability***, but accepts risk value ***Medium risk*** or less.
- ***Application interface usability***, but accepts likelihood ***one incident a month*** and consequence ***50 received phone calls to support*** or less.
- ***Data privacy (CM)***, but accepts risk value ***Medium risk*** or less.
- ***Company brand & reputation***, but accepts risk value ***Low risk*** or less.

6. - 9. ***Company brand & reputation*** is harmed indirectly via

- ***Data privacy (GDI)***.
- ***Application interface usability***.
- ***Application availability***.
- ***Data privacy (CM)***.

Note that the order of the sentences does not affect the translation. In other words, the sentences commute. The order we choose is merely a convention in order to provide a clear presentation.

5 Threat Diagrams

A threat diagram is a presentation of the chains of events initiated by threats, that have consequences for the assets. Its basic building blocks are the seven elements of Figure 5: deliberate, accidental and non-human threats, vulnerabilities, threat scenarios, unwanted incidents, and assets. Threat scenarios and unwanted incidents may be assigned a likelihood. We refer to the threats, threat scenarios, unwanted incidents and assets of a threat diagram as its *vertices*. Vulnerabilities and likelihoods are used as annotations. Threat diagrams have three different relations: *initiate*, *leads to* and *impact*. A threat exploits one or more vulnerabilities, initiating a threat scenario or an unwanted incident. This is captured by the binary initiate relation. The threat scenario or unwanted incident may then lead to new threat scenarios or unwanted incidents, either as direct consequences or by the threat exploiting other vulnerabilities. This is captured by the binary leads



Figure 5: Basic building blocks of threat diagrams

to relation. Either of these relations may be assigned a likelihood and/or one or more vulnerabilities. The impact relation relates an unwanted incident and an asset that is harmed as a consequence of this incident. Each impact relation may be assigned a consequence.

Figure 6 presents the graphical syntax of a threat diagram.

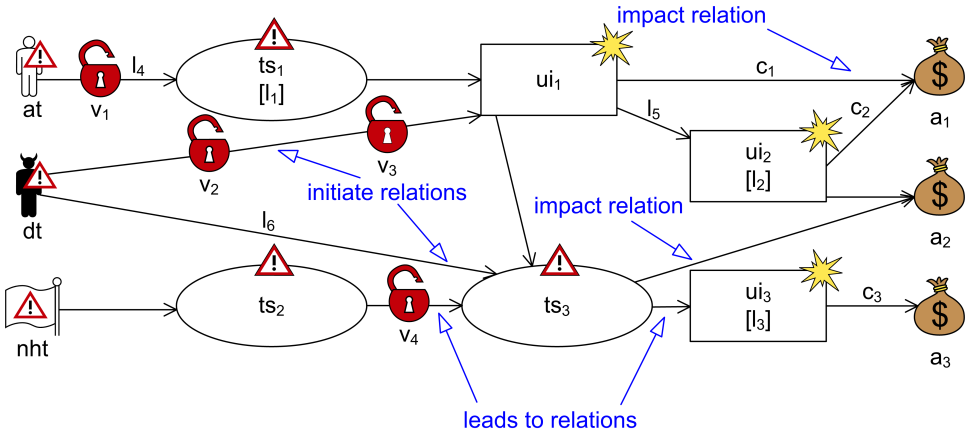


Figure 6: Graphical syntax of threat diagrams

5.1 Textual Syntax of Threat Diagrams

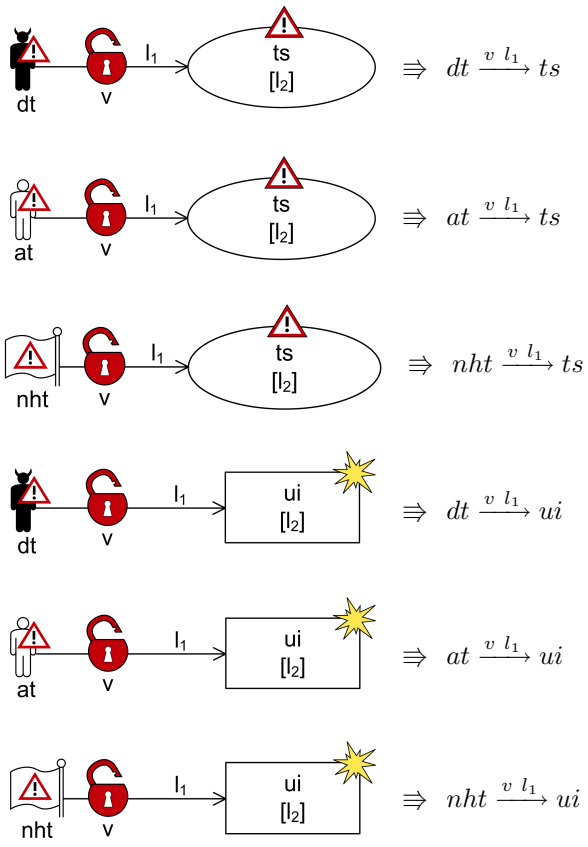
$$
\begin{aligned}
diagram &= (\{vertex\}^{-}, \{relation\});\\
vertex &= threat \mid threat\ scenario \mid unwanted\ incident \mid asset;\\
relation &= initiate \mid leads\ to \mid impact;\\
initiate &= threat \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\qquad threat \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident;\\
leads\ to &= threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\qquad threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident \mid\\
&\qquad unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\qquad unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident;\\
impact &= unwanted\ incident \xrightarrow{[consequence]} asset \mid\\
&\qquad threat\ scenario \rightarrow asset;\\
threat &= deliberate\ threat \mid accidental\ threat \mid non\text{-}human\ threat;\\
deliberate\ threat &= identifier;
\end{aligned}
$$

$$
\begin{aligned}
\textit{accidental threat} &= \textit{identifier};\\
\textit{non-human threat} &= \textit{identifier};\\
\textit{vulnerability set} &= \{\textit{vulnerability}\}^{-};\\
\textit{vulnerability} &= \textit{identifier};\\
\textit{threat scenario} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{unwanted incident} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{asset} &= \textit{identifier};\\
\textit{likelihood} &= \textit{linguistic term} \mid \textit{numerical value};\\
\textit{consequence} &= \textit{linguistic term} \mid \textit{numerical value};
\end{aligned}
$$

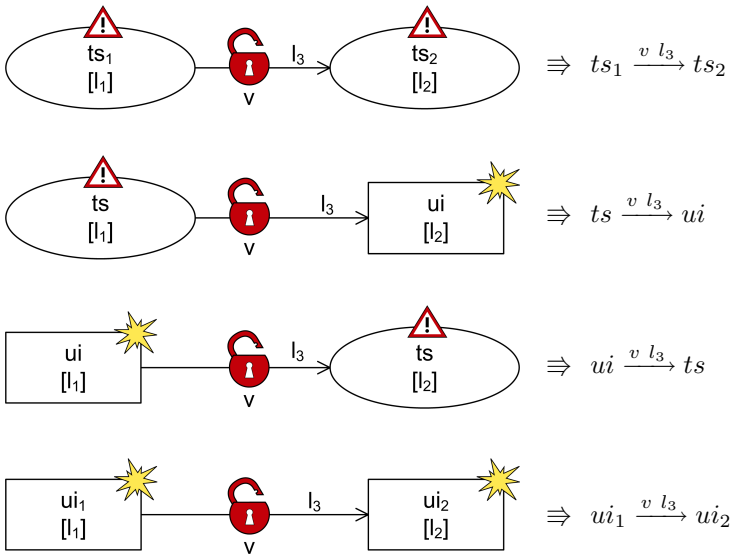
5.2 Translation from Graphical to Textual Syntax

See Section 4 for instructions on how to translate the vertices. Here we present the translation rules including all optional elements. We then explain how the rules may be modified if one or more of the optional elements are removed. To simplify the presentation of the translation rules, they contain only one vulnerability, as there is no icon for vulnerability sets. In order to translate a relation with more than one vulnerability, replace the v in the textual expression of the rule with the set of vulnerabilities.

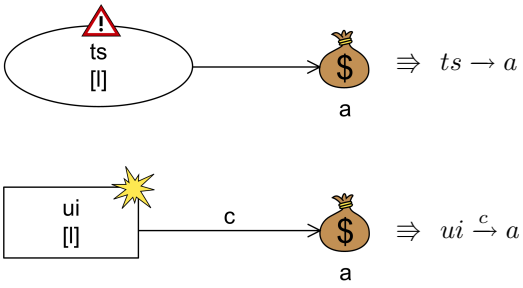
5.2.1 Initiate relation



5.2.2 Leads to relation



5.2.3 Impact relation



The optional elements of the threat diagram are vulnerability sets, likelihoods, and consequences. In the case where a vulnerability or consequence is absent in the graphical expression, simply remove the corresponding element from the textual expression (see Table 1 for an overview of naming conventions). The translation of the relation is the same whether or not likelihoods have been assigned to its vertices.

5.3 Translation from the Textual Syntax to English

5.3.1 Diagram

A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0, m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation.

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

5.3.2 Vertices

$$
\begin{aligned}
[\![\, dt \,]\!] &:= \textbf{\textit{dt}} \text{ is a deliberate threat.} \\
[\![\, at \,]\!] &:= \textbf{\textit{at}} \text{ is an accidental threat.} \\
[\![\, nht \,]\!] &:= \textbf{\textit{nht}} \text{ is a non-human threat.} \\
[\![\, a \,]\!] &:= \textbf{\textit{a}} \text{ is an asset.} \\
[\![\, ts \,]\!] &:= \text{Threat scenario } \textbf{\textit{ts}} \text{ occurs with undefined likelihood.} \\
[\![\, ts(l) \,]\!] &:= \text{Threat scenario } \textbf{\textit{ts}} \text{ occurs with } [\![\, l \,]\!]. \\
[\![\, ui \,]\!] &:= \text{Unwanted incident } \textbf{\textit{ui}} \text{ occurs with undefined likelihood.} \\
[\![\, ui(l) \,]\!] &:= \text{Unwanted incident } \textbf{\textit{ui}} \text{ occurs with } [\![\, l \,]\!].
\end{aligned}
$$

5.3.3 Initiate relation

The semantics for the initiate relation is independent of which kind of threat it originates from and whether it ends in a threat scenario or an unwanted incident. In the following definitions the threat t may be replaced by dt, at or nht, for deliberate, accidental and non-human threats respectively. The ts representing a threat scenario may be replaced with an unwanted incident ui.

$$
\begin{aligned}
[\![\, t \rightarrow ts \,]\!] &:= \textbf{\textit{t}} \text{ initiates } \textbf{\textit{ts}} \text{ with undefined likelihood.} \\
[\![\, t \xrightarrow{l} ts \,]\!] &:= \textbf{\textit{t}} \text{ initiates } \textbf{\textit{ts}} \text{ with } [\![\, l \,]\!]. \\
[\![\, t \xrightarrow{V_n} ts \,]\!] &:= \textbf{\textit{t}} \text{ exploits } [\![\, V_n \,]\!] \text{ to initiate } \textbf{\textit{ts}} \text{ with undefined likelihood.} \\
[\![\, t \xrightarrow{V_n\ l} ts \,]\!] &:= \textbf{\textit{t}} \text{ exploits } [\![\, V_n \,]\!] \text{ to initiate } \textbf{\textit{ts}} \text{ with } [\![\, l \,]\!].
\end{aligned}
$$

5.3.4 Leads to relation

As for the initiate relation, the semantics for the leads to relation is the same whether or not it originates and/or ends in a threat scenario or unwanted incident. The definitions below are given for a leads to relation between threat scenarios. For the other combinations replace one or more ts with *ui(l)* as before.

$$
\begin{aligned}
[\![\, ts_1 \rightarrow ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with undefined likelihood.} \\
[\![\, ts_1 \xrightarrow{l} ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with } [\![\, l \,]\!]. \\
[\![\, ts_1 \xrightarrow{V_n} ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with undefined likelihood, due to } [\![\, V_n \,]\!]. \\
[\![\, ts_1 \xrightarrow{V_n\ l} ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with } [\![\, l \,]\!], \text{ due to } [\![\, V_n \,]\!].
\end{aligned}
$$

5.3.5 Impact relation

As for the two previous relations, the semantics for the unannotated impact relation is the same independent of whether it originates in a threat scenario or an unwanted incident.

$$[\![\, ts \rightarrow a \,]\!] := \textbf{\textit{ts}} \text{ impacts } \textbf{\textit{a}} \text{ with undefined consequence.}$$

However, only impact relations originating in unwanted incidents may be annotated with consequences. This relation have the following semantics:

$$[\![\, ui \xrightarrow{c} a \,]\!] := \textbf{\textit{ui}} \text{ impacts } \textbf{\textit{a}} \text{ with } [\![\, c \,]\!].$$

5.3.6 Annotations

$$
\begin{aligned}
[\![v]\!] &:= \text{vulnerability } \boldsymbol{v} \\
[\![V_n]\!] &:= \text{vulnerabilities } \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \\
[\![l]\!] &:= \text{likelihood } \boldsymbol{l} \\
[\![c]\!] &:= \text{consequence } \boldsymbol{c}
\end{aligned}
$$

5.4 Example

We continue the example from the section on asset diagrams to illustrate the translation of threat diagrams. In the threat diagram in Figure 7, we have numbered the relations from 1 to 21 in the order in which they are translated.

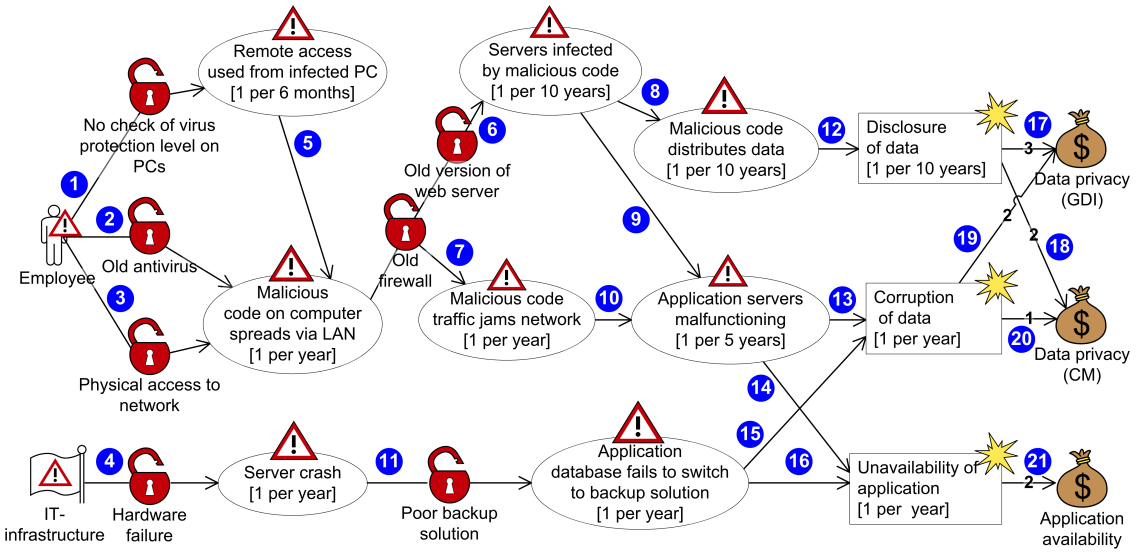


Figure 7: Threat diagram

We start by translating the vertices of the diagram, from the top left to the bottom right:

1. ***Employee*** is an accidental threat.
2. ***IT-infrastructure*** is a non-human threat.
3. Threat scenario ***Remote access used from infected PC*** occurs with likelihood ***1 per 6 months***.
4. Threat scenario ***Malicious code on computer spreads via LAN*** occurs with likelihood ***1 per year***.
5. Threat scenario ***Server crash*** occurs with likelihood ***1 per year***.
6. Threat scenario ***Servers infected by malicious code*** occurs with likelihood ***1 per 10 years***.
7. Threat scenario ***Malicious code traffic jams network*** occurs with likelihood ***1 per year***.
8. Threat scenario ***Malicious code distributes data*** occurs with likelihood ***1 per 10 years***.

9. Threat scenario ***Application servers malfunctioning*** occurs with likelihood ***1 per 5 years***.
10. Threat scenario ***Application database fails to switch to backup solution*** occurs with likelihood ***1 per year***.
11. Unwanted incident ***Disclosure of data*** occurs with likelihood ***1 per 10 years***.
12. Unwanted incident ***Corruption of data*** occurs with likelihood ***1 per year***.
13. Unwanted incident ***Unavailability of application*** occurs with likelihood ***1 per year***.
14. ***Data privacy (GDI)*** is an asset.
15. ***Data privacy (CM)*** is an asset.
16. ***Application availability*** is an asset.

Then we translate the relations of the diagram, in the order suggested by the numbering in Figure 7:

1. ***Employee*** exploits vulnerability ***No check of virus protection level on PCs*** to initiate ***Remote access used from infected PC*** with undefined likelihood.
2. ***Employee*** exploits vulnerability ***Old antivirus*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.
3. ***Employee*** exploits vulnerability ***Physical access to network*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.
4. ***IT-infrastructure*** exploits vulnerability ***Hardware failure*** to initiate ***Server crash*** with undefined likelihood.
5. ***Remote access used from infected PC*** leads to ***Malicious code on computer spreads via LAN*** with undefined likelihood.
6. ***Malicious code on computer spreads via LAN*** leads to ***Servers infected by malicious code*** with undefined likelihood, due to vulnerabilities ***Old firewall***, ***Old version of web server***.
7. ***Malicious code on computer spreads via LAN*** leads to ***Malicious code traffic jams network*** with undefined likelihood, due to vulnerability ***Old firewall***.
8. ***Servers infected by malicious code*** leads to ***Malicious code distributes data*** with undefined likelihood.
9. ***Servers infected by malicious code*** leads to ***Application servers malfunctioning*** with undefined likelihood.
10. ***Malicious code traffic jams network*** leads to ***Application servers malfunctioning*** with undefined likelihood.
11. ***Server crash*** leads to ***Application database fails to switch to backup solution*** with undefined likelihood, due to vulnerability ***Poor backup solution***.
12. ***Malicious code distributes data*** leads to ***Disclosure of data*** with undefined likelihood.
13. ***Application servers malfunctioning*** leads to ***Corruption of data*** with undefined likelihood.

14. ***Application servers malfunctioning*** leads to ***Unavailability of application*** with undefined likelihood.
15. ***Application database fails to switch to backup solution*** leads to ***Corruption of data*** with undefined likelihood.
16. ***Application database fails to switch to backup solution*** leads to ***Unavailability of application*** with undefined likelihood.
17. ***Disclosure of data*** impacts ***Data privacy (GDI)*** with consequence *3*.
18. ***Disclosure of data*** impacts ***Data privacy (CM)*** with consequence *2*.
19. ***Corruption of data*** impacts ***Data privacy (GDI)*** with consequence *2*.
20. ***Corruption of data*** impacts ***Data privacy (CM)*** with consequence *1*.
21. ***Unavailability of application*** impacts ***Application availability*** with consequence *2*.

As was the case with the asset diagrams, we may choose to structure the translation of the relations to reflect the structure of the diagrams, and to make the translation more readable:

1.-3. ***Employee*** exploits vulnerability

- ***No check of virus protection level on PCs*** to initiate ***Remote access used from infected PC*** with undefined likelihood.
- ***Old antivirus*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.
- ***Physical access to network*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.

4. ***IT-infrastructure*** exploits vulnerability ***Hardware failure*** to initiate ***Server crash*** with undefined likelihood.
5. ***Remote access used from infected PC*** leads to ***Malicious code on computer spreads via LAN*** with undefined likelihood.

6.-7. ***Malicious code on computer spreads via LAN*** leads to

- ***Servers infected by malicious code*** with undefined likelihood, due to vulnerabilities ***Old firewall***, ***Old version of web server***.
- ***Malicious code traffic jams network*** with undefined likelihood, due to vulnerability ***Old firewall***.

8.-9. ***Servers infected by malicious code*** leads to

- ***Malicious code distributes data*** with undefined likelihood.
- ***Application servers malfunctioning*** with undefined likelihood.

10. ***Malicious code traffic jams network*** leads to ***Application servers malfunctioning*** with undefined likelihood.
11. ***Server crash*** leads to ***Application database fails to switch to backup solution*** with undefined likelihood, due to vulnerability ***Poor backup solution***.
12. ***Malicious code distributes data*** leads to ***Disclosure of data*** with undefined likelihood.

13.-14. ***Application servers malfunctioning*** leads to

- ***Corruption of data*** with undefined likelihood.
- ***Unavailability of application*** with undefined likelihood.

15.-16. ***Application database fails to switch to backup solution***

- ***Corruption of data*** with undefined likelihood.
- ***Unavailability of application*** with undefined likelihood.

17.-18. ***Disclosure of data*** impacts

- ***Data privacy (GDI)*** with consequence *3*.
- ***Data privacy (CM)*** with consequence *2*.

19.-20. ***Corruption of data*** impacts

- ***Data privacy (GDI)*** with consequence *2*.
- ***Data privacy (CM)*** with consequence *1*.

21. ***Unavailability of application*** impacts ***CM3.*** ***Application availability*** with consequence *2*.

As we see from the example, the optimal order of the sentences depends on how you want to structure them.

6 Risk Diagrams

A risk diagram provides a summary of a threat diagram, displaying the risks posed to the assets by the threats. It has five basic elements: deliberate, accidental, and non-human threats, risks, and assets (see Figure 8). A risk may be assigned a risk value, risk function, or a likelihood and consequence pair, indicating its severity.



Figure 8: Basic building blocks of risk diagrams

All of the basic building blocks of the risk diagram are referred to as *vertices*. A risk diagram has three kinds of relations: an *initiate* relation from a threat to a risk, a *leads to* relation between risks, and an *impact* relation from a risk to an asset. The graphical syntax is displayed in Figure 9.

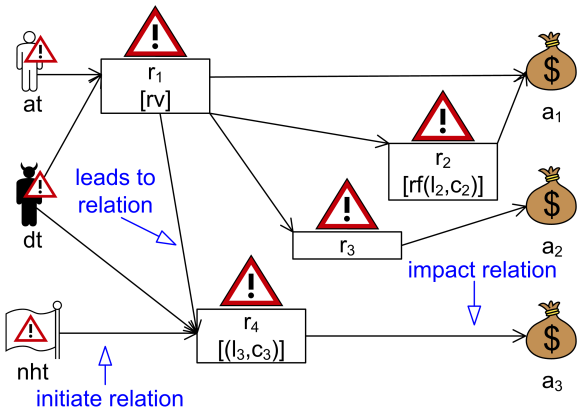


Figure 9: Graphical syntax of risk diagrams

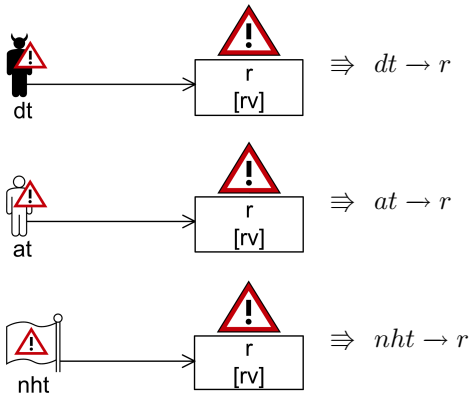
6.1 Textual Syntax of Risk Diagrams

$$
\begin{aligned}
\mathit{diagram} &= (\{\mathit{vertex}\}^{-}, \{\mathit{relation}\});\\
\mathit{vertex} &= \mathit{threat} \mid \mathit{risk} \mid \mathit{asset};\\
\mathit{relation} &= \mathit{initiate} \mid \mathit{leads\ to} \mid \mathit{impact};\\
\mathit{initiate} &= \mathit{threat} \rightarrow \mathit{risk};\\
\mathit{leads\ to} &= \mathit{risk} \rightarrow \mathit{risk};\\
\mathit{impact} &= \mathit{risk} \rightarrow \mathit{asset};\\
\mathit{threat} &= \mathit{deliberate\ threat} \mid \mathit{accidental\ threat} \mid \mathit{non\text{-}human\ threat};\\
\mathit{deliberate\ threat} &= \mathit{identifier};\\
\mathit{accidental\ threat} &= \mathit{identifier};\\
\mathit{non\text{-}human\ threat} &= \mathit{identifier};\\
\mathit{risk} &= \mathit{identifier}\,[(\mathit{risk\ level})]\,;\\
\mathit{asset} &= \mathit{identifier};\\
\mathit{risk\ level} &= \mathit{risk\ value} \mid \mathit{risk\ function}(\mathit{likelihood}, \mathit{consequence}) \mid\\
&\quad (\mathit{likelihood}, \mathit{consequence});\\
\mathit{risk\ value} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{risk\ function} &= \mathit{identifier};\\
\mathit{likelihood} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{consequence} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};
\end{aligned}
$$

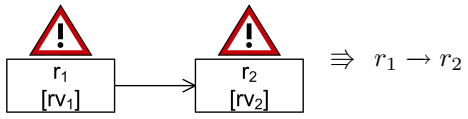
6.2 Translation from Graphical to Textual Syntax

See Section 4 for instructions on how to translate the vertices. To translate the relations from the graphical to the textual syntax we use the following rules:

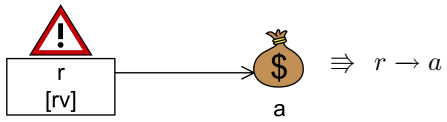
6.2.1 Initiate relation



6.2.2 Leads to relation



6.2.3 Impact relation



6.3 Translation from the Textual Syntax to English

6.3.1 Diagram

A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0, m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation.

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

6.3.2 Vertices

$$\begin{aligned}
[\![dt]\!] &:= \textbf{\textit{dt}} \text{ is a deliberate threat.} \\
[\![at]\!] &:= \textbf{\textit{at}} \text{ is an accidental threat.} \\
[\![nht]\!] &:= \textbf{\textit{nht}} \text{ is a non-human threat.} \\
[\![a]\!] &:= \textbf{\textit{a}} \text{ is an asset.} \\
[\![r]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with undefined risk level.} \\
[\![r(rv)]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with } [\![rv]\!]. \\
[\![r(rf(l,c))]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with } [\![rf(l,c)]\!]. \\
[\![r(l,c)]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with } [\![(l,c)]\!].
\end{aligned}$$

6.3.3 Initiate relation

As was the case for threat diagrams, the semantics of an initiate relation is independent of which kind of threat it originates from. In the following definition the threat t may be replaced by dt, at or nht, for deliberate, accidental and non-human threats respectively.

$$[\![t \rightarrow r]\!] := \textbf{\textit{t}} \text{ initiates } \textbf{\textit{r}}.$$

6.3.4 Leads to relation

$$[\![r_1 \rightarrow r_2]\!] := \textbf{\textit{r}}_1 \text{ leads to } \textbf{\textit{r}}_2.$$

6.3.5 Impact relation

$$[\![r \rightarrow a]\!] := \textbf{\textit{r}} \text{ impacts } \textbf{\textit{a}}.$$

6.3.6 Annotations

$$\begin{aligned} [\![rv]\!] &:= \text{ risk value } \textbf{\textit{rv}} \\ [\![rf(l,c)]\!] &:= \text{ risk function } \textbf{\textit{rf}} \text{ of } [\![(l,c)]\!] \\ [\![(l,c)]\!] &:= [\![l]\!] \text{ and } [\![c]\!] \\ [\![l]\!] &:= \text{ likelihood } \textbf{\textit{l}} \\ [\![c]\!] &:= \text{ consequence } \textbf{\textit{c}} \end{aligned}$$

6.4 Example

We continue the example from the last section to illustrate the translation of risk diagrams.

We start by translating the vertices of the risk diagram in Figure 10:

1. ***Employee*** is an accidental threat.
2. ***IT-infrastructure*** is a non-human threat.
3. Risk ***Disclosure of data 1*** occurs with risk value ***[medium]***.
4. Risk ***Disclosure of data 2*** occurs with risk value ***[low]***.
5. Risk ***Corruption of data 1*** occurs with risk value ***[medium]***.
6. Risk ***Corruption of data 2*** occurs with risk value ***[major]***.
7. Risk ***Unavailability of application*** occurs with risk value ***[medium]***.
8. ***Data privacy (GDI)*** is an asset.
9. ***Data privacy (CM)*** is an asset.
10. ***Application availability*** is an asset.

Translating the diagram relation by relation in the order suggested in Figure 10:

1. ***Employee*** initiates ***Disclosure of data 1***.
2. ***Employee*** initiates ***Disclosure of data 2***.

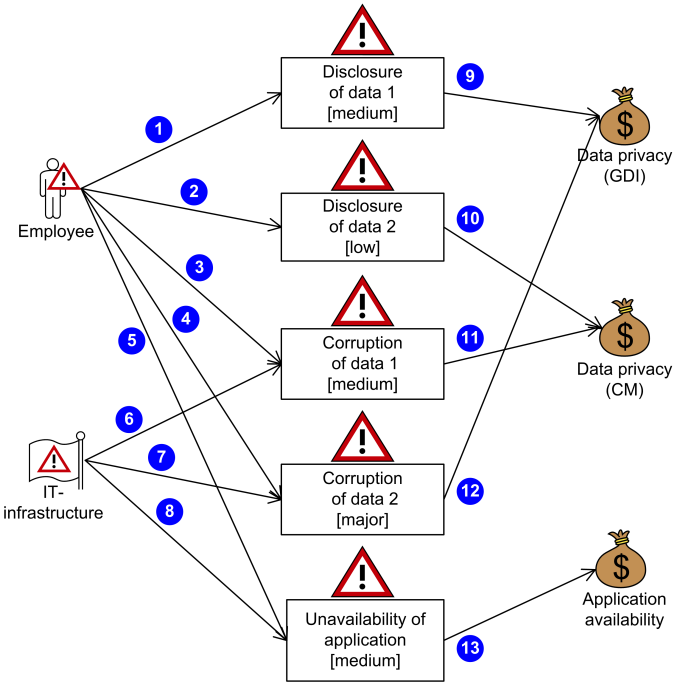


Figure 10: Risk diagram

3. ***Employee*** initiates ***Corruption of data 1***.
4. ***Employee*** initiates ***Corruption of data 2***.
5. ***Employee*** initiates ***Unavailability of application***.
6. ***IT-infrastructure*** initiates ***Corruption of data 1***.
7. ***IT-infrastructure*** initiates ***Corruption of data 2***.
8. ***IT-infrastructure*** initiates ***Unavailability of application***.
9. ***Disclosure of data 1*** impacts ***Data privacy (GDI)***.
10. ***Disclosure of data 2*** impacts ***Data privacy (CM)***.
11. ***Corruption of data 1*** impacts ***Data privacy (CM)***.
12. ***Corruption of data 2*** impacts ***Data privacy (GDI)***.
13. ***Unavailability of application*** impacts ***Application availability***.

Structuring the text as with the other diagrams:

1.-5. ***Employee*** initiates

- ***Disclosure of data 1***.
- ***Disclosure of data 2***.
- ***Corruption of data 1***.
- ***Corruption of data 2***.

- ***Unavailability of application***.

6.-8. ***IT-infrastructure*** initiates

- ***Corruption of data 1***.
- ***Corruption of data 2***.
- ***Unavailability of application***.

9. ***Disclosure of data 1*** impacts ***Data privacy (GDI)***.
10. ***Disclosure of data 2*** impacts ***Data privacy (CM)***.
11. ***Corruption of data 1*** impacts ***Data privacy (CM)***.
12. ***Corruption of data 2*** impacts ***Data privacy (GDI)***.
13. ***Unavailability of application*** impacts ***Application availability***.

7 Treatment Diagrams

A treatment diagram is a presentation of proposed treatments of elements of a threat diagram. It expands the threat diagram, inserting the risks from the risk diagram between the appropriate unwanted incidents and assets, and adds treatment scenarios. Its basic building blocks are shown in Figure 11.

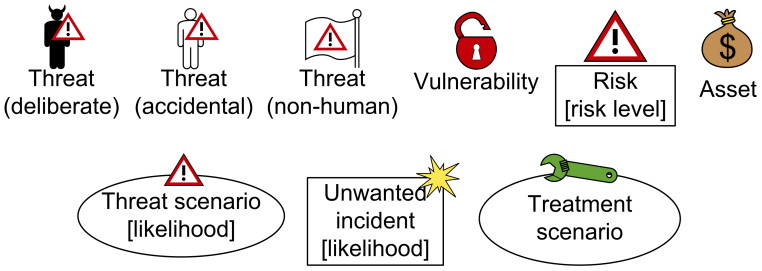


Figure 11: Basic building blocks of treatment diagrams

We refer to all the basic building blocks except vulnerabilities as *vertices*. The *initiate* and *leads to* relations are defined as for the threat diagram, adding the possibility of leads to relations from unwanted incidents to risks. This new relation is annotated with neither vulnerabilities nor likelihoods. As was the case for risk diagrams, the *impact* relation relates a risk to an asset.

Treatment Category	Abbreviation
avoid	*av*
decrease likelihood	*dl*
decrease consequence	*dc*
share	*sh*
retain	*re*

Table 2: Treatment category abbreviations

We also have an additional relation, *treat*, from a treatment scenario to any other kind of element. It may be annotated with a treatment category, cf. the Joint Australian/New Zealand Standard for Risk Management [3, p. 20]. The abbreviations in Table 2 are used to annotate the treat relations in the semantics for the treatment and treatment overview diagrams.

Figure 12 shows the graphical syntax of treatment diagrams.

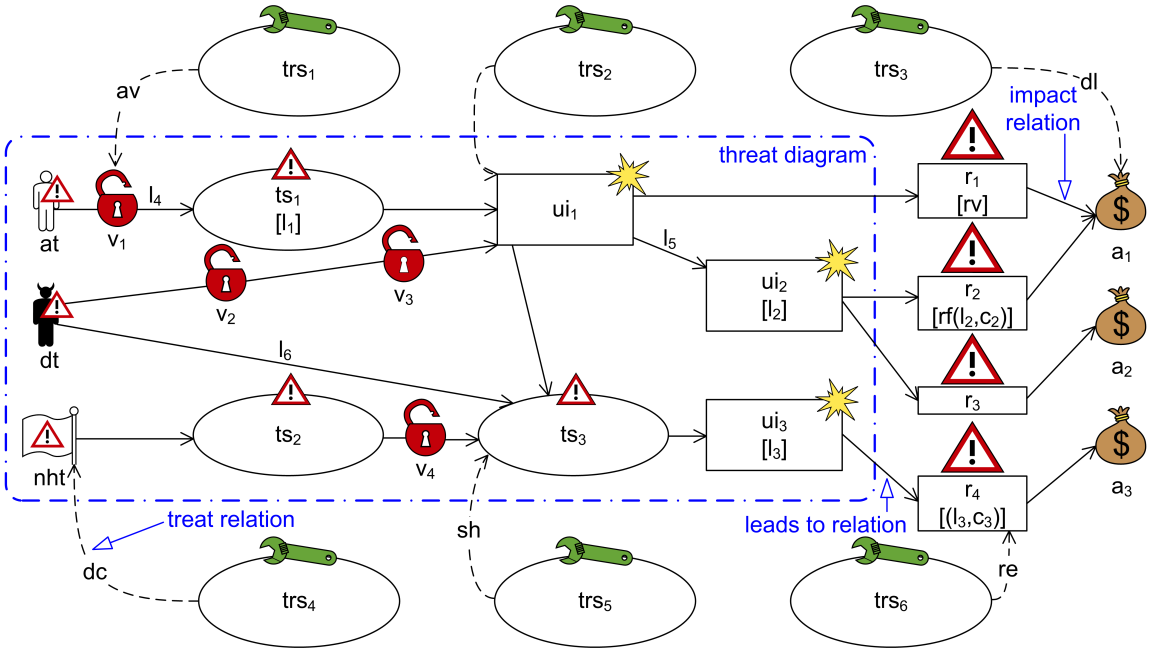


Figure 12: Graphical syntax of treatment diagrams

7.1 Textual Syntax of Treatment Diagrams

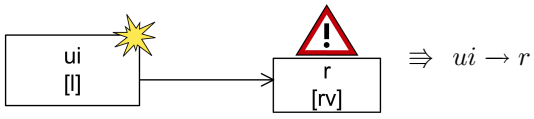
$$
\begin{aligned}
diagram &= (\{vertex\}^{-}, \{relation\});\\
vertex &= threat \mid threat\ scenario \mid unwanted\ incident \mid risk \mid asset \mid\\
&\quad\ treatment\ scenario;\\
relation &= initiate \mid leads\ to \mid impact \mid treat;\\
initiate &= threat \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\quad\ threat \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident \mid\\
&\quad\ threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\quad\ threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident \mid\\
&\quad\ unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\quad\ unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident;\\
leads\ to &= unwanted\ incident \rightarrow risk;\\
impact &= risk \rightarrow asset;\\
treat &= treatment\ scenario \xrightarrow{[treatment\ category]} threat \mid\\
&\quad\ treatment\ scenario \xrightarrow{[treatment\ category]} vulnerability \mid\\
&\quad\ treatment\ scenario \xrightarrow{[treatment\ category]} threat\ scenario \mid\\
&\quad\ treatment\ scenario \xrightarrow{[treatment\ category]} unwanted\ incident \mid
\end{aligned}
$$

$$
\begin{aligned}
&\textit{treatment scenario} \xrightarrow{[\textit{treatment category}]} \textit{risk} \mid \\
&\textit{treatment scenario} \xrightarrow{[\textit{treatment category}]} \textit{asset};\\
\textit{threat} &= \textit{deliberate threat} \mid \textit{accidental threat} \mid \textit{non-human threat};\\
\textit{deliberate threat} &= \textit{identifier};\\
\textit{accidental threat} &= \textit{identifier};\\
\textit{non-human threat} &= \textit{identifier};\\
\textit{vulnerability set} &= \{\textit{vulnerability}\}^{-};\\
\textit{vulnerability} &= \textit{identifier};\\
\textit{threat scenario} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{unwanted incident} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{risk} &= \textit{identifier}\,[(\textit{risk level})];\\
\textit{asset} &= \textit{identifier};\\
\textit{treatment scenario} &= \textit{identifier};\\
\textit{treatment category} &= \text{‘avoid’} \mid \text{‘decrease likelihood’} \mid \text{‘decrease consequence’} \mid \text{‘share’} \mid \text{‘retain’};\\
\textit{risk level} &= \textit{risk value} \mid \textit{risk function}(\textit{likelihood}, \textit{consequence}) \mid \\
&\quad (\textit{likelihood}, \textit{consequence});\\
\textit{risk value} &= \textit{linguistic term} \mid \textit{numerical value};\\
\textit{risk function} &= \textit{identifier};\\
\textit{likelihood} &= \textit{linguistic term} \mid \textit{numerical value};\\
\textit{consequence} &= \textit{linguistic term} \mid \textit{numerical value};
\end{aligned}
$$

7.2 Translation from Graphical to Textual Syntax

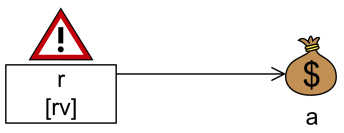
The part of the treatment diagram in Figure 12 within the blue threat diagram-box are translated in the section on threat diagrams. What remains to be translated are the leads to relation from an unwanted incident to a risk, the impact relation, and the treat relations. As mentioned in earlier sections, we include all optional elements in the translation rules.

7.2.1 Leads to relation



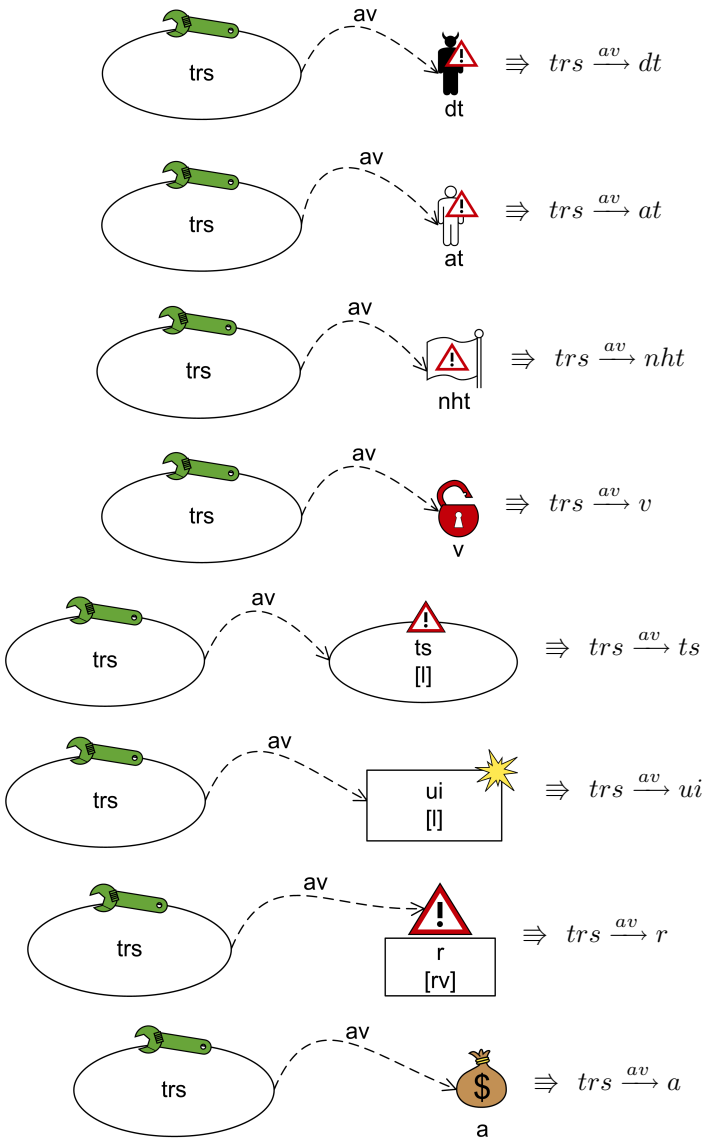
$\Rrightarrow\ ui \rightarrow r$

7.2.2 Impact relation



$\Rrightarrow\ r \rightarrow a$

7.2.3 Treat relation



To obtain the translation rule for a treat relation without treatment category, remove the *av*-annotation from the arrow in the textual expression. If the treat relation is annotated with a different treatment category, simply replace *av* with the appropriate abbreviation.

7.3 Translation from the Textual Syntax to English

The translation of the vertices and relations within the blue threat diagram-box in Figure 12 is given in Section 5.3. The translations of risks and assets, and of the impact relation, are given in Section 6.3. This leaves treatment scenarios, treat relations and the new leads to relation.

7.3.1 Vertices

$$[\![\, trs \,]\!] := \textbf{\textit{trs}} \text{ is a treatment scenario.}$$

7.3.2 Leads to relation

$$[\![\, ui \rightarrow r \,]\!] := \textbf{\textit{ui}} \text{ leads to } \textbf{\textit{r}}.$$

7.3.3 Treat relation

An unannotated treat relation has the same translation, independent of the object it treats:

$$[\![\, trs \rightarrow dt \,]\!] := \textbf{\textit{trs}} \text{ treats } \textbf{\textit{dt}}.$$

Here the dt for deliberate threat may be replaced by at, nht, v, ts, ui, r or a.

When the treat relation is annotated with a treatment category, the translation depends on the object being treated. In the following, the annotation av may be replaced by any of the other annotations. t for threat may be replaced by dt, at or nht.

$$\begin{aligned}
[\![\, trs \xrightarrow{av} t \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{t}} \text{ attacking the system.} \\
[\![\, trs \xrightarrow{av} v \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{v}} \text{ being exploited.} \\
[\![\, trs \xrightarrow{av} ts \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{ts}} \text{ being initiated.} \\
[\![\, trs \xrightarrow{av} ui \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{ui}} \text{ being initiated.} \\
[\![\, trs \xrightarrow{av} r \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{r}}. \\
[\![\, trs \xrightarrow{av} a \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{a}} \text{ being harmed.}
\end{aligned}$$

7.3.4 Annotations

The annotations left to translate are the treatment categories:

$$\begin{aligned}
[\![\, av \,]\!] &:= \text{avoids the risk} \\
[\![\, dl \,]\!] &:= \text{decreases the likelihood} \\
[\![\, dc \,]\!] &:= \text{decreases the consequences} \\
[\![\, sh \,]\!] &:= \text{shares the risk} \\
[\![\, re \,]\!] &:= \text{retains the risk}
\end{aligned}$$

7.4 Example

We continue the example from the last section to illustrate the translation of treatment diagrams:

Most of the diagram has already been translated in Sections 5.4 and 6.4. The only remaining vertices are the treatment scenarios:

1. ***Increase awareness of security risks*** is a treatment scenario.
2. ***Upgrade server*** is a treatment scenario.
3. ***Limit access to network*** is a treatment scenario.

We translate the remaining relations in the order suggested in Figure 13:

1. ***Increase awareness of security risks*** treats ***No check of virus protection level on PCs***.
2. ***Upgrade server*** treats ***Old version of webserver***.

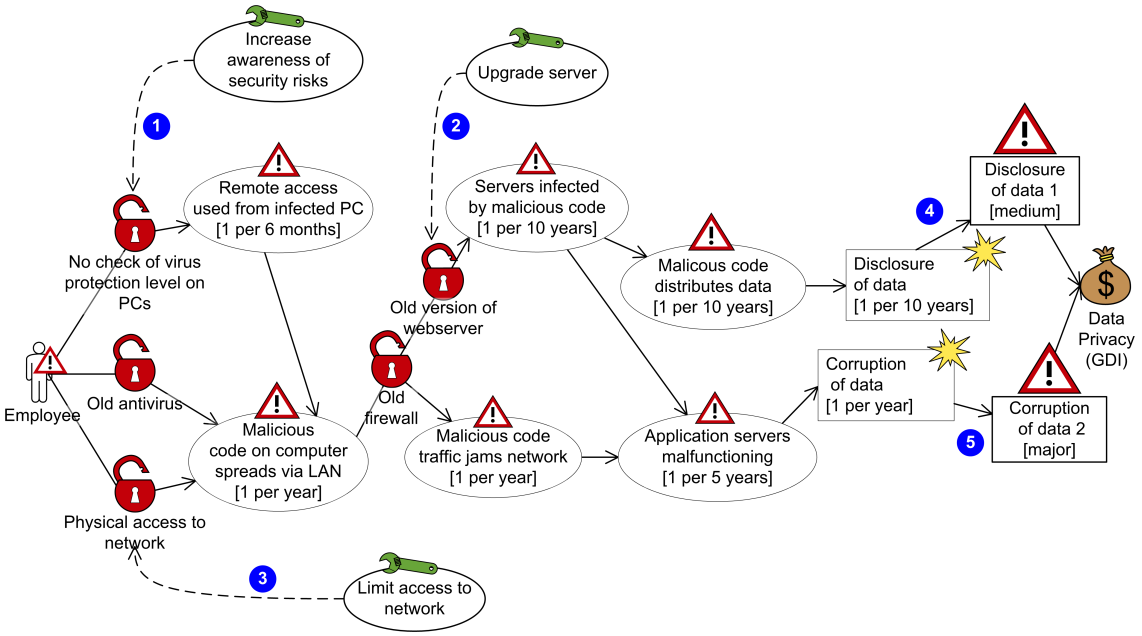


Figure 13: Treatment diagram

3. ***Limit access to network*** treats ***Physical access to network***.

4. ***Disclosure of data*** leads to ***Disclosure of data 1***.

5. ***Corruption of data*** leads to ***Corruption of data 2***.

8 Treatment Overview Diagrams

The last kind of CORAS diagram is the treatment overview diagram. Its basis is a risk diagram, to which the treatment scenarios of a corresponding treatment diagram are added. As in the treatment diagram, *treat* relations are used to indicate the elements affected by each treatment scenario. They may be annotated with a treatment category, cf. the Joint Australian/New Zealand Standard for Risk Management [3, p. 20]. See Table 2, p. 23, for the treatment category abbreviations. Figure 14 displays the basic component of treatment overview diagrams, and their graphical syntax is shown in Figure 15. We refer to all of the basic building blocks as *vertices*.

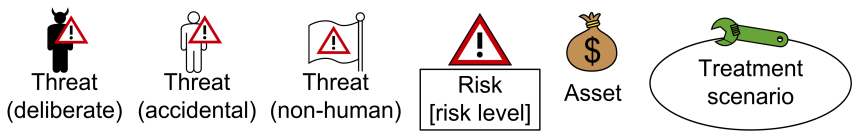


Figure 14: Basic building blocks of treatment overview diagrams

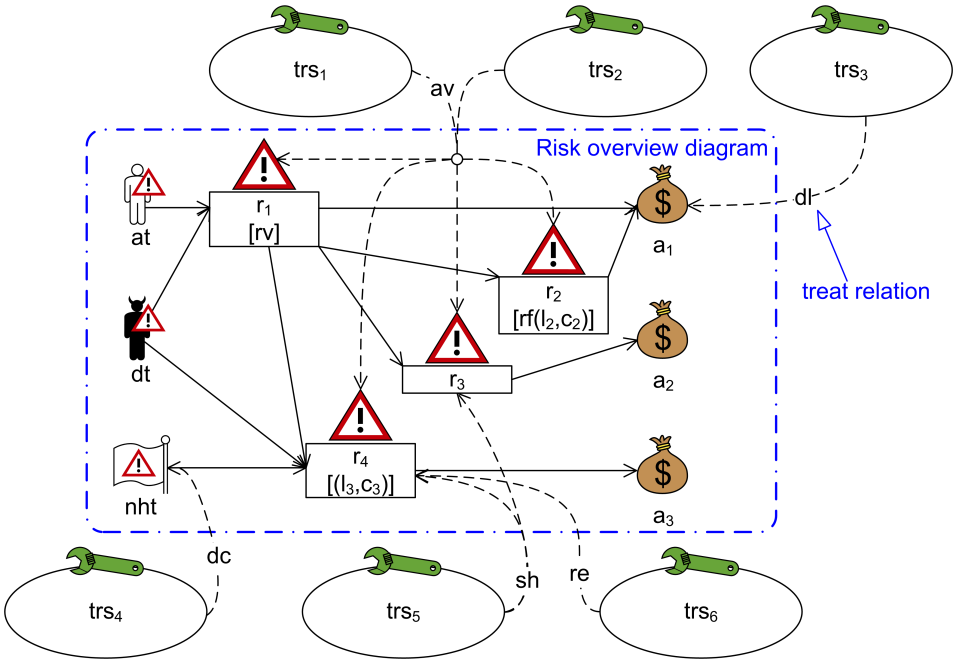


Figure 15: Graphical syntax of treatment overview diagrams

8.1 Textual Syntax of Treatment Overview Diagrams

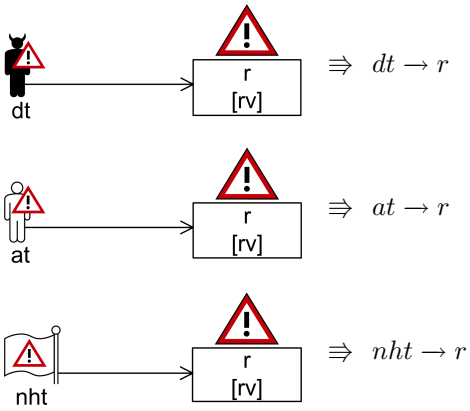
$$
\begin{aligned}
\mathit{diagram} &= (\{\mathit{vertex}\}^{-}, \{\mathit{relation}\});\\
\mathit{vertex} &= \mathit{threat} \mid \mathit{risk} \mid \mathit{asset} \mid \mathit{treatment\ scenario};\\
\mathit{relation} &= \mathit{initiate} \mid \mathit{leads\ to} \mid \mathit{impact} \mid \mathit{treat};\\
\mathit{initiate} &= \mathit{threat} \rightarrow \mathit{risk};\\
\mathit{leads\ to} &= \mathit{risk} \rightarrow \mathit{risk};\\
\mathit{impact} &= \mathit{risk} \rightarrow \mathit{asset};\\
\mathit{treat} &= \mathit{treatment\ scenario} \xrightarrow{[\mathit{treatment\ category}]} \mathit{threat} \mid\\
&\qquad \mathit{treatment\ scenario} \xrightarrow{[\mathit{treatment\ category}]} \mathit{risk} \mid\\
&\qquad \mathit{treatment\ scenario} \xrightarrow{[\mathit{treatment\ category}]} \mathit{asset};\\
\mathit{threat} &= \mathit{deliberate\ threat} \mid \mathit{accidental\ threat} \mid \mathit{non\text{-}human\ threat};\\
\mathit{deliberate\ threat} &= \mathit{identifier};\\
\mathit{accidental\ threat} &= \mathit{identifier};\\
\mathit{non\text{-}human\ threat} &= \mathit{identifier};\\
\mathit{risk} &= \mathit{identifier}\ [(\mathit{risk\ level})]\ ;\\
\mathit{asset} &= \mathit{identifier};\\
\mathit{treatment\ scenario} &= \mathit{identifier};\\
\mathit{treatment\ category} &= \text{‘avoid’} \mid \text{‘decrease likelihood’} \mid \text{‘decrease consequence’} \mid \text{‘share’} \mid \text{‘retain’};\\
\mathit{risk\ level} &= \mathit{risk\ value} \mid \mathit{risk\ function}(\mathit{likelihood}, \mathit{consequence}) \mid\\
&\qquad (\mathit{likelihood}, \mathit{consequence});
\end{aligned}
$$

$$\begin{aligned} \textit{risk value} &= \textit{linguistic term} \mid \textit{numerical value};\\ \textit{risk function} &= \textit{identifier};\\ \textit{likelihood} &= \textit{linguistic term} \mid \textit{numerical value};\\ \textit{consequence} &= \textit{linguistic term} \mid \textit{numerical value}; \end{aligned}$$

8.2 Translation from Graphical to Textual Syntax

Most of the treatment overview diagram was translated in Section 6.2, while treatment scenarios and the treat relation was translated in Section 7.2. The following are the relevant translation rules:

8.2.1 Initiate relation

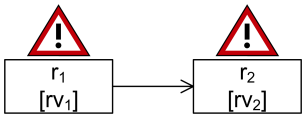


$\Rrightarrow dt \rightarrow r$

$\Rrightarrow at \rightarrow r$

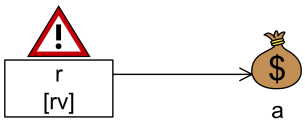
$\Rrightarrow nht \rightarrow r$

8.2.2 Leads to relation



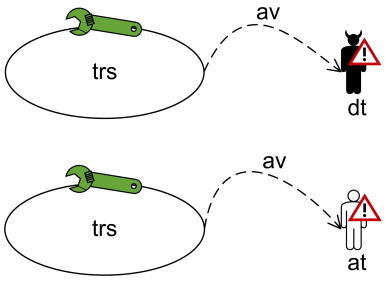
$\Rrightarrow r_1 \rightarrow r_2$

8.2.3 Impact relation



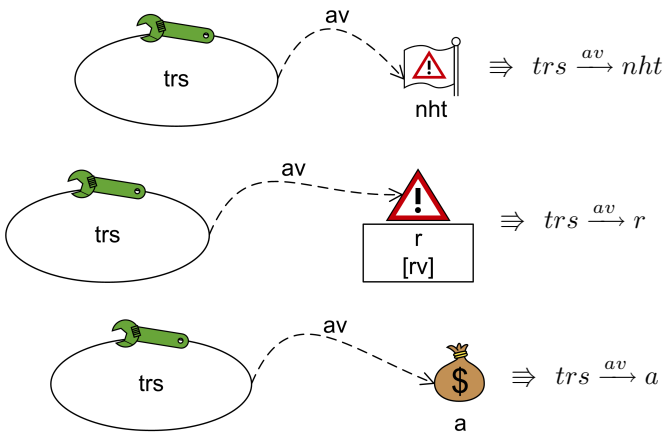
$\Rrightarrow r \rightarrow a$

8.2.4 Treat relation



$\Rrightarrow trs \xrightarrow{av} dt$

$\Rrightarrow trs \xrightarrow{av} at$



8.3 Translation from the Textual Syntax to English

The basic components, the treatment categories and the treat relations have the same structured semantics as in the treatment diagram. The rest of the semantics can be found in Section 6 on risk diagrams.

8.4 Example

We continue the example from the last section to illustrate the translation of treatment overview diagrams:

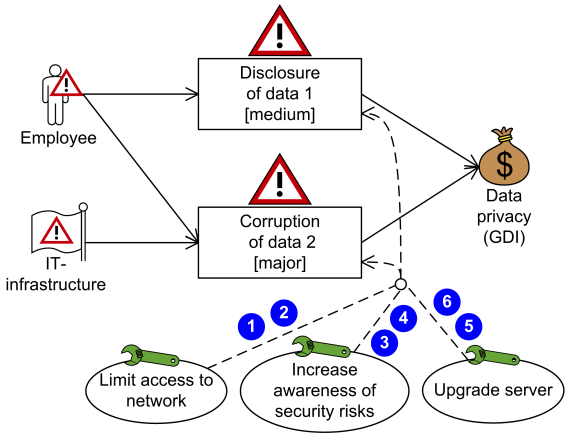


Figure 16: Treatment overview diagram

Everything but the treat relations have already been translated:

1. ***Limit access to network*** treats ***Disclosure of data 1***.
2. ***Limit access to network*** treats ***Corruption of data 2***.
3. ***Increase awareness of security risks*** treats ***Disclosure of data 1***.

4. ***Increase awareness of security risks*** treats ***Corruption of data 2***.
5. ***Upgrade server*** treats ***Disclosure of data 1***.
6. ***Upgrade server*** treats ***Corruption of data 2***.

Organising the results:

1.-2. ***Limit access to network*** treats

- ***Disclosure of data 1***.
- ***Corruption of data 2***.

3.-4. ***Increase awareness of security risks*** treats

- ***Disclosure of data 1***.
- ***Corruption of data 2***.

5.-6. ***Upgrade server*** treats

- ***Disclosure of data 1***.
- ***Corruption of data 2***.

9 Conclusion

The CORAS language has been designed to be easily understandable in order to aid communication in a security risk analysis context. Even so, situations in which there is a need to explain the intended meaning of a construct or expression are bound to arise. An example of such a situation is when the analysis results are distributed to parties, within the client company, which have not been part of the analysis process.

In order to fill this need, this paper has presented a structured semantics for the CORAS security risk modelling language. We have provided instructions on how to translate the two main CORAS diagrams, via the textual syntax, into a paragraph of English.

The report satisfies the success criteria stated at the end of Section 1:

1. *The translation from CORAS diagrams to English should be modular.* We divided the translation into two independent steps: (A) Graphical to textual syntax, and (B) Textual syntax to English. Both of these component translations are modular (the diagram and textual expressions are translated relation by relation) so the complete translation is modular.

2. *The resulting paragraph should be understandable English.* The wording of the English phrases in the structured semantics is based on the descriptions used by CORAS developers to explain the diagrams to non-specialists during a CORAS security risk analysis. This gives us a translation into phrases of clear, understandable English.

3. *The translation should be easy to perform.* The translation of a diagram is done by pattern matching, first by matching each relation to a translation rule and removing unwanted optional elements, then by matching the resulting textual expression to a rule in the structured semantics.

4. *The translation should be possible to automate.* The translation rules and the structured semantics are presented in such a way that the pattern matching may be done automatically. However, the structuring of the translation depends to a large degree on the structure of the original diagram. Thus it is difficult to give a general recommendation on how this is done.

This means that while it is possible to automatically structure the translation to reflect the branching nature of the CORAS diagrams, a more comprehensive structuring may require human intervention unless the structure of the diagram adheres to a predefined style.

5. *It should be possible to translate inconsistent diagrams, and the translation should enable the user to identify inconsistencies.* As a CORAS diagram is translated relation by relation and not from a more global perspective, it does not matter to the translation whether or not the diagram is inconsistent. However, the inconsistencies may not be conspicuous before the translation is appropriately structured.

Future Work

The work presented in this paper is the starting point for several research activities. The most immediate would be empirical testing of the translation process and the resulting sentences. The CORAS tool (see `http://coras.sourceforge.net`) will be updated to reflect the structure of the textual syntax and facilitate automatic translation.

The development of the CORAS method and language continues in several projects at SINTEF ICT, building on experiences from industrial case studies. There is also ongoing work aiming for an integrated approach to security and usability analysis.

Related Work

Misuse cases [2, 18, 19] was an important source of inspiration in the development of the UML profile mentioned in Sec. 1. A misuse case is a kind of UML use case [11] which characterizes functionality that the system should not allow. There are a number of security oriented extensions of UML, e.g. UMLSec [12] and SecureUML [13]. These and related approaches, however, have all been designed to capture security properties and security aspects at a more detailed level than our language. Moreover, their focus is not on brainstorming sessions as in our case.

Fault tree is a tree-notation used in fault tree analysis (FTA) [9]. The top node represents an unwanted incident, or failure, and the different events that may lead to the top event are modelled as branches of nodes, with the leaf node as the causing event. Our threat diagrams often resemble fault trees, but may have more than one top node.

Event tree analysis (ETA) [8] focuses on illustrating the consequences of an event and the probabilities of these. Event trees can to a large extent also be simulated in our notation.

Attack trees [17] aim to provide a formal and methodical way of describing the security of a system based on the attacks it may be exposed to. The notation uses a tree structure similar to fault trees, with the attack goal as the top node and different ways of achieving the goal as leaf nodes. Our approach supports this way of modelling, but additionally facilitates the specification of the attack initiators (threats) and the harm caused by the attack (damage to assets).

Acknowledgements

The research for this paper has been funded by the projects SECURIS (152839/220) and DIGIT (180052/S10) funded by the Research Council of Norway, and the EU-project S3MS (IST-2006-027004). The authors thank Gyrd Brændeland, Iselin Engan, Mass Soldal Lund, Atle Refsdal and Bjørnar Solhaug for valuable input.

References

[1] Jan Øyvind Aagedal, Folker den Braber, Theo Dimitrakos, Bjørn Axel Gran, Dimitris Raptis, and Ketil Stølen. Model-based risk assessment to improve enterprise security. In *EDOC'02*, pages 51–64. IEEE Computer Society, 2002.

[2] Ian F. Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20(1):58–66, 2003.

[3] AS/NZS 4360:2004. *Australian/New Zealand Standard for Risk Management*, 2004.

[4] Folker den Braber, Ida Hogganvik, Mass Soldal Lund, Ketil Stølen, and Fredrik Vraalsen. Model-based security analysis in seven steps – a guided tour to the CORAS method. *BT Technology Journal*, 25(1):101–117, 2007.

[5] Ida Hogganvik and Ketil Stølen. On the comprehension of security risk scenarios. In *Proc.of 13th Int. Workshop on Program Comprehension*, pages 115–124. IEEE Computer Society, 2005.

[6] Ida Hogganvik and Ketil Stølen. Risk Analysis Terminology for IT systems: Does it match Intuition? In *Proc. of Int. Symposium on Empirical Software Engineering*, pages 13–23. IEEE Computer Society, 2005.

[7] Ida Hogganvik and Ketil Stølen. A Graphical Approach to Risk Identification, Motivated by Empirical Investigations. In *Proc. of 9th Int. Conf. on Model Driven Engineering Languages and Systems*, volume 4199 of *LNCS*, pages 574–588. Springer, 2006.

[8] IEC60300. *Event Tree Analysis in Dependability management – Part 3: Application guide – Section 9: Risk analysis of technological systems.* 1995.

[9] IEC61025. *Fault Tree Analysis (FTA).* 1990.

[10] ISO/IEC 14977:1996(E). *Information technology — Syntactic metalanguage — Extended BNF*, first edition, 1996.

[11] Ivar Jacobson, Magnus Christenson, Patrik Jonsson, and Gunnar Övergaard. *Object-Oriented Software Engineering. A Use Case Driven Approach.* Addison-Wesley, 1992.

[12] Jan Jürjens. *Secure Systems Development with UML.* Springer, 2005.

[13] Torsten Lodderstedt, David A. Basin, and Jürgen Doser. SecureUML: A UML-based modeling language for model-driven security. In *UML'02*, volume 2460 of *LNCS*, pages 426–441. Springer, 2002.

[14] Mass Soldal Lund, Ida Hogganvik, Seehusen Fredrik, and Ketil Stølen. UML profile for security assessment. Technical Report STF40 A03066, SINTEF ICT, 2003.

[15] OMG. *Unified Modeling Language Specification, version 2.0*, 2004.

[16] OMG. *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*, 2005.

[17] Bruce Schneier. Attack trees: Modeling security threats. *Dr. Dobb's Journal of Software Tools*, 24(12):21–29, December 1999.

[18] Guttorm Sindre and Andreas L. Opdahl. Eliciting security requirements with misuse cases. In *TOOLS-PACIFIC'00*, pages 120–131, 2000.

[19] Guttorm Sindre and Andreas L. Opdahl. Templates for misuse case description. In *REFSQ'01*, pages 125–136, 2001.

Appendix E

Model-based Security Analysis in Seven Steps – a Guided Tour to the CORAS Method

Model-based security analysis in seven steps – a guided tour to the CORAS method

Folker den Braber[1], Ida Hogganvik[1,2], Mass Soldal Lund[1,2], Ketil Stølen[1,2] and Fredrik Vraalsen[3]
[1]SINTEF ICT
{fbr, iho, msl, kst}@sintef.no
[2]Department of Informatics, University of Oslo
[3]Unified Consulting AS
fvr@unified.no

Abstract

This paper presents the CORAS method for model based security analysis. The presentation is case-driven. We follow two analysts in their interaction with an organisation by which they have been hired to carry out a security risk analysis. The analysis is divided into seven main steps, and the paper devotes a separate section to each of them. The paper focuses in particular on the use of the CORAS security risk modelling language as a means for communication and interaction during the seven steps.

Introduction

Have you ever asked yourself some of the following questions:

- Should I worry when using my credit card on the Internet?
- How safe is my Internet bank account?
- How many doctors or health care personnel have access to my personal health records?
- Can I be sure that I am the only one reading my email?
- How crucial can a single personal mistake be for my company?

A security risk analysis may provide answers to such questions. CORAS is a method for conducting security risk analysis, which is abbreviated to "security analysis" in the rest of this paper. CORAS provides a customised language for threat and risk modelling, and comes with a detailed guideline explaining how the language should be used to capture and model relevant information during the various stages of the security analysis. In this respect CORAS is model-based. The Unified Modeling Language (UML) [20] is typically used to model the target of the analysis. For documenting intermediate results, and for presenting the overall conclusions we use special CORAS diagrams which are inspired by UML. The CORAS method provides a computerised tool designed to support documenting, maintaining and reporting analysis results through risk modelling, table based documentation, consistency checking and more[1].

The seven steps of security analysis in CORAS are summarised as follows:

- Step 1: The first step involves an introductory meeting. The main item on the agenda for this meeting is to get the representatives of the client to present their overall goals of the analysis and the target they wish to have analysed. Hence, during the initial step the analysts will gather information based on the client's presentations and discussions.

[1] The tool may be downloaded from http://coras.sourceforge.net .

- Step 2: The second step also involves a separate meeting with representatives of the client. However, this time the analysts will present *their* understanding of what they learned at the first meeting and from studying documentation that has been made available to them by the client. The second step also involves a rough, high-level security analysis. During this analysis the first threats, vulnerabilities, threat scenarios and unwanted incidents are identified. They will be used to help directing and scoping the more detailed analysis still to come.
- Step 3: The third step involves a more refined description of the target to be analysed, and also all assumptions and other preconditions being made. Step three is terminated once all this documentation has been approved by the client.
- Step 4: This step is organised as a workshop gathering people with expertise on the target of analysis. The goal is to identify as many potential unwanted incidents as possible, as well as threats, vulnerabilities and threat scenarios.
- Step 5: The fifth step is also organised as a workshop. This time with the focus on estimating consequences and likelihood values for each of the identified unwanted incidents.
- Step 6: This step involves giving the client the first overall risk picture. This will typically trigger some adjustments and corrections.
- Step 7: The last step is devoted to treatment identification, as well as addressing cost/benefit issues of the treatments. This step is best organised as a workshop.

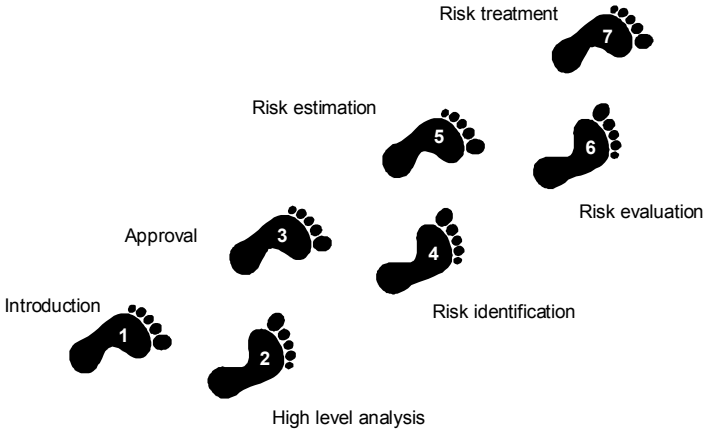


Figure 1 – Steps on "the CORAS tour"

In the following sections we will give a presentation of the seven steps by means of an example where the CORAS method is applied to a telemedicine case[2]. We distinguish the example from general descriptions of the method, language etc. by formatting the example in *italic.* We end each section with a summary of the process and guidelines for the relevant step. In addition to sections for the seven steps, there is also a summary and comparison to related work. Before we start, let us introduce the example that will follow us throughout this paper:

In one region of the country, an experimental telemedicine system has been set up. A dedicated network between the regional hospital and several primary health care centres (PHCC) allows a general practitioner (GP) to conduct a cardiological examination of a patient (at the PHCC) in cooperation with a cardiologist located at the hospital. During an

[2] This example is inspired by security analyses of real telemedicine systems conducted within the CORAS project [16, 25]. However, changes have been made so that it no longer represents any real case.

examination both medical doctors have access to the patient's health record and all data from the examination is streamed to the cardiologist's computer.

The National Ministry of Health is concerned whether the patient privacy is sufficiently protected, and hires a security analysis consultancy company to do a security analysis of the cardiology system. The consultancy company appoints a security analysis leader and a security analysis secretary to do the job (referred to as "the analysts"). In cooperation with the ministry, they organise the meetings and workshops that the security analysis consists of.

Step 1: Introductory meeting

Before starting to identify and analyse potential risks to something, it is necessary to know exactly what this something is. What is the scope and what are the assumptions that we make? In other words, we need to know what we are supposed to protect before we can start finding what threatens it and how it may be harmed, as well as how it should be protected. It is essential that the client of the security analysis and the analysts obtain a common understanding of the terminology to be used, the target of the analysis, the assets to be protected, and the scope of the analysis.

Let us see what this means in practice for the telemedicine case:

A meeting is organised where, in addition to the analysts and a representative from the ministry, the IT manager of the regional hospital and a general practitioner from one of the PHCC's participate.

This meeting is where the overall setting of the analysis is decided, and the first step is taken towards establishing the target description that will be used later in the analysis. The meeting starts with the security analysis leader giving a brief presentation of the method to be used, what the client (the National Ministry of Health) can expect from the analysis, and a proposed meeting plan. The analysis leader reminds the client of the responsibilities with respect to providing necessary information and documentation about the target in question, as well as allocating people with suitable background to participate in the security analysis meetings and workshops.

The IT manager then presents the telemedicine system which will be the target of analysis. As part of the presentation she draws the picture shown in Figure 2. From the picture we see that speech and other data from the examination of a patient is streamed over a dedicated network, while access to the patient's health record (stored in a database at the regional hospital) is given through an encrypted channel over the Internet. Next in line after the IT manager is the medical doctor from the PHCC. She talks about her personal experiences from using the system.

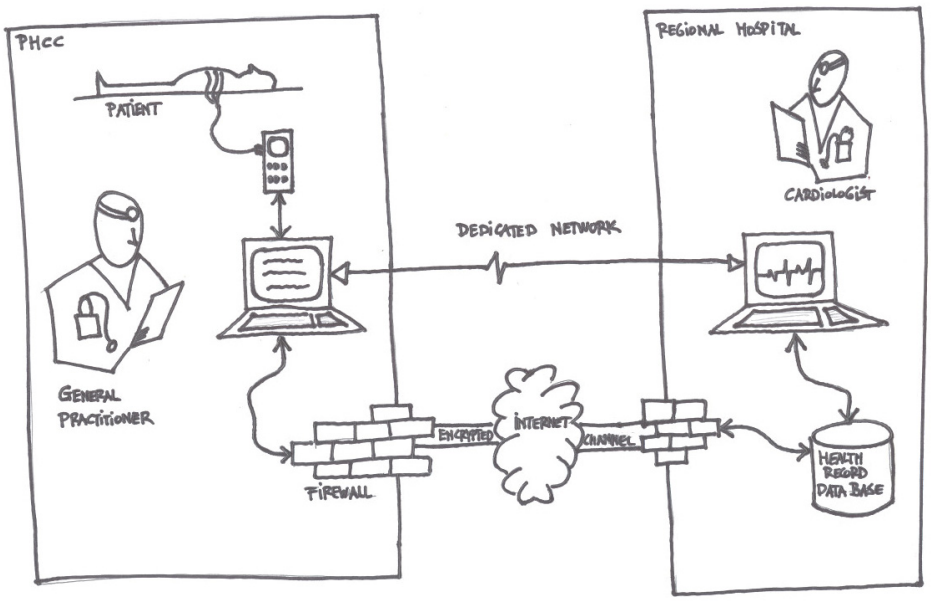


Figure 2 – Picture of the target

After the presentations, a discussion on the scope and focus of the analysis follows. The representative of the ministry emphasises that they are particularly worried about the confidentiality and integrity of the health records and other medical data, first and foremost for the sake of the patients' health, but also because of the public's trust in the national health care system. For the medical doctor the most important thing is the patients' health and well-being, and hence the availability and integrity of the telemedicine system. The IT manager explains that they have already made a security analysis of the health record database and the encrypted access, so she is confident that this part of the system is secure and reliable. After some discussion the representative of the ministry decides that the focus will be on confidentiality and integrity of medical data, and the availability of the service, but that the access to the health record database is outside the scope of analysis.

As the last point on the agenda, the participants set up a plan for the rest of the analysis with dates and indications of who should be present.

Step 1 - summary

Tasks:	**People that should participate:**

• The security analysis method is introduced.
• The client presents the goals and the target of the analysis.
• The focus and scope of the analysis is set.
• The meetings and workshops are planned.
	• Analysis leader (required)
	• Analysis secretary (required)
	• Representatives of the client:
	  o Decision makers (required)
	  o Technical expertise (optional)
	  o Users (optional)

Modelling guideline:	
1. At this early stage of the analysis it can be useful to describe the target with informal models like drawings, pictures or sketches on a blackboard.	2. The presentation can later be supplemented with more formal modelling techniques such as UML or data flow-diagram.

Step 2: High-level analysis

The second step is called the high-level analysis, and as the name indicates this involves conducting an initial analysis of the target. This step also typically involves a meeting between the analysts and the representatives of the client. The main purpose is to identify assets and get an overview of the main risks. Finding the assets that need protection is

initiated in step 2 and completed in step 3. The remaining four steps of the analysis will be directed towards these assets. The outcome of the high-level analysis helps the analysts to identify the aspects of the target having the most urgent need for in-depth analysis, and hence makes it easier to define the exact scope and focus of the full analysis.

The second meeting starts with the security analysis leader presenting the analysts' understanding of the target to be analysed. He has formalised the information presented by the client at the previous meeting, as well as documentation received in the mean time in UML diagrams [20] . The UML class diagram (Figure 3) shows the relevant concepts and how they relate, while the UML collaboration diagram (Figure 4) illustrates the physical organisation of the target. Furthermore, the medical doctor's description of use has been captured as a UML activity diagram (Figure 5). During this presentation the participants representing the client make corrections and eliminate errors, so that the result is a target description all parties can agree upon. In the class and collaboration diagrams the security analysis leader has also indicated what he understands as the focus areas of the analysis.

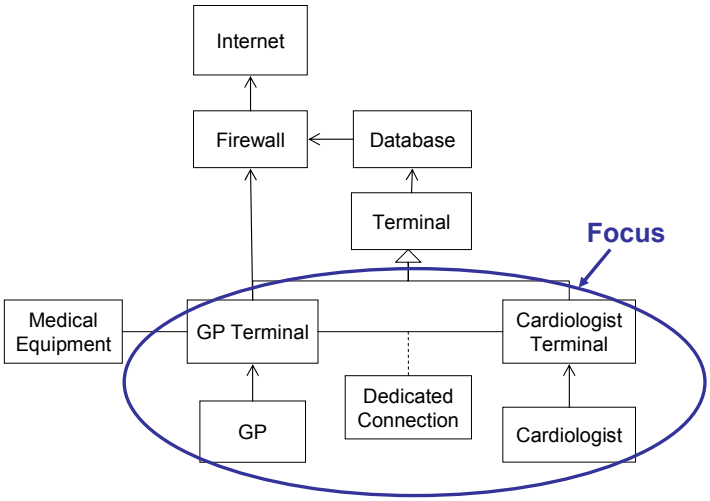


Figure 3 – Class diagram showing a conceptual view of the target

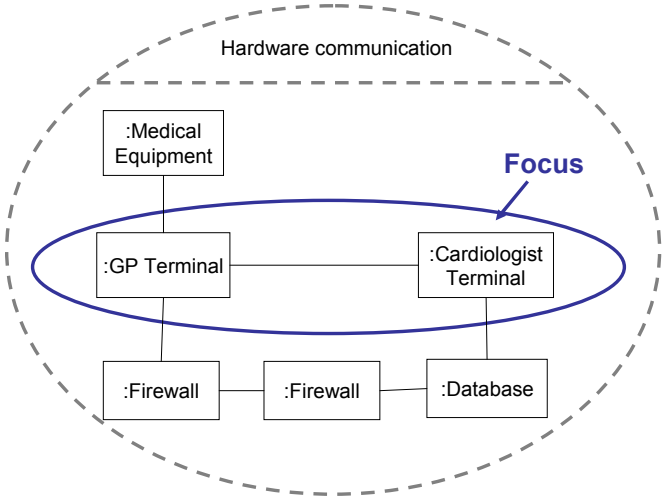


Figure 4 – Collaboration diagram illustrating the physical communication lines

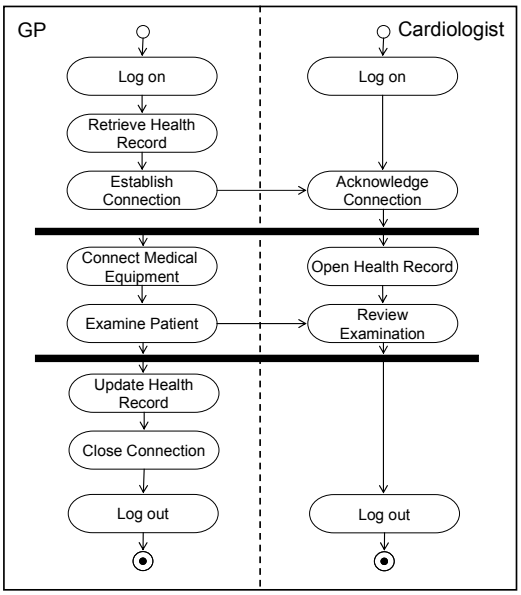


Figure 5 – Activity diagram describing the parallel processes of the GP and the cardiologist

After agreeing on a target description, the analysis moves on to asset identification. An asset is something in or related to the target to which the client assigns great value. Based on the discussion at the introductory meeting, the analysis leader has prepared an initial "CORAS asset diagram" (Figure 6) to help specifying the scope of the analysis. The asset diagram shows the National Ministry of Health as the client (i.e. the stakeholder that is initiating and paying for the analysis), and its four assets: "Health records", "Provision of telecardiology service", "Patient's health" and "Public's trust in system". Because trust and health are difficult to measure, especially in a technical setting like this, the analysis leader makes a distinction between direct *and* indirect *assets. He explains direct assets as assets that may be harmed directly by an unwanted incident, while the indirect assets are only harmed if one of the direct assets is harmed first. In the asset diagram the direct assets are placed within the target of analysis region and the indirect are placed outside. The arrows show dependencies between the assets, such that, e.g., harm to "Health records" may cause harm to "Public's trust in health care system". The dashed lines symbolises the client's, or other interested parties, relation to the assets.*

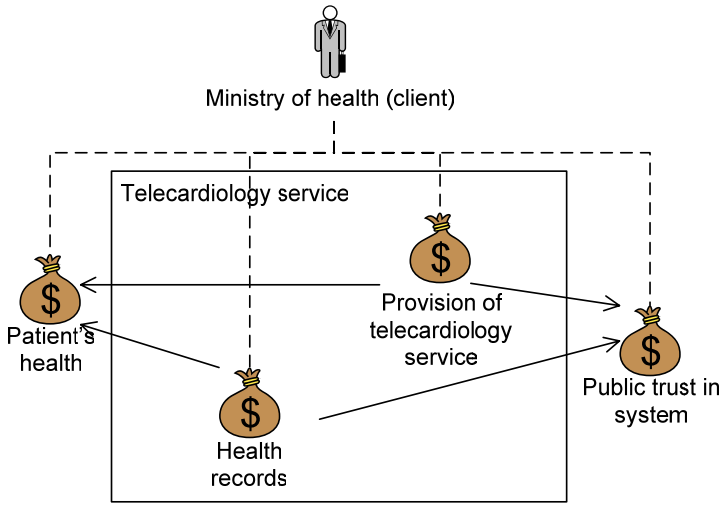
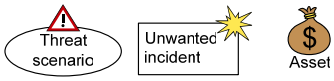


Figure 6 – Asset diagram

After agreeing on the assets, the analysts conduct a high-level analysis together with the analysis participants. The short brainstorming should identify the most important threats and

vulnerabilities, but without going into great detail. In this case the client is concerned about hackers, eavesdroppers, system failure and whether the security mechanisms are sufficient. These threats and vulnerabilities do not necessarily involve major risks, but give the analysis leader valuable input on where to start the analysis. The analysis secretary documents the results by filling in the high-level risk table shown in Table 1 .

Table 1 – High-level risk table

		
Who/what causes it?	**How? What is the incident? What does it harm?**	**What makes it possible?**
Hacker	*Breaks into the system and steals health records*	*Insufficient security*
Employee	*Sloppiness compromises confidentiality of health records*	*Insufficient training*
Eavesdropper	*Eavesdropping on dedicated connection*	*Insufficient protection of connection*
System failure	*System goes down during examination*	*Unstable connection/immature technology*
Employee	*Sloppiness compromises integrity of health record*	*Prose-based health records (i.e. natural language)*
Network failure	*Transmission problems compromises integrity of medical data*	*Unstable connection/immature technology*
Employee	*Health records leaks out by accident, compromises their confidentiality and damages the trust in the system*	*Possibility of irregular handling of health records*

Step 2 - summary

Tasks:	**People that should be present:**

• The target as understood by the analysts is presented.
• The assets are identified.
• A high-level analysis is conducted.
	• Security analysis leader (required)
	• Security analysis secretary (required)
	• Representatives of the client:
	  ○ Decision makers (required)
	  ○ Technical expertise (required)
	  ○ Users (optional)

Modelling guidelines:	
Asset diagrams: 1. Draw a region that logically or physically represents the target of analysis. 2. Place the direct assets within the region. 3. Place the indirect assets outside the region. Indirect assets are a harmed as a consequence of a direct asset being harmed first. 4. Indicate with arrows which assets may affect other assets. 5. Assets may be ranked according to their importance. 6. If the analysis has more than one client, the clients should be associated with their assets.	Target descriptions: 1. Use a formal or standardised notation such as UML, but ensure that the notation is explained thoroughly so that the participants understand it. 2. Create models of both the static and the dynamic features of the target. Static may be hardware configurations, network design etc., while dynamic may be work processes, information flow etc. 3. For the static parts of the description UML class diagrams and UML collaboration diagrams (or similar notations) are recommended. 4. For the dynamic parts we recommend UML activity diagrams and UML sequence diagrams (or similar notations)

Step 3: Approval

The last of the preparatory steps is the approval step. The approval is often conducted as a separate meeting, but may also take place via email. The main goal is to finalise the documentation and characterisation of target and assets, and get this formally approved by the client. At the end of this meeting there should be a document (possibly with a list of required changes) to which all parties agree on and commit to. The approval also involves defining consequence scales (for each asset) and a likelihood scale. Multiple consequence scales are used when it is difficult or inappropriate to measure damage to all assets according to the same scale. E.g., it is easier to measure "income" in monetary values than "company brand". There should only be one likelihood scale appropriate for the analysis scope, e.g. based on a time-interval (years, weeks, hours etc.) or probabilities. The last activity of the approval is to decide upon the risk evaluation criteria. The criteria states which level of risk the client accepts for each of the assets.

The security analysis leader has updated his presentation from the last meeting based on comments from the other participants, and the target and asset descriptions are now approved. Based on the discussions in the first two meetings and issues identified in the high-level analysis, it is decided to narrow the scope of the analysis, and agree upon the following target definition:

> *"The target of analysis will be the availability of the telecardiology service, and confidentiality and integrity of health records and medical data in relation to use of the service and related equipment. The indirect asset "Public's trust to system" is to be kept outside the scope*."

A risk is the potential for an unwanted incident to have an impact upon objectives (assets) [3], or in other words to reduce the value of at least one of the identified assets. Often the client accepts some risks that are not judged to be critical rather than eliminating or reducing them. This may be because of shortage of resources to implement changes, conflicting concerns or the treatment costs will be greater than the benefits. As a first step towards distinguishing risks that can be accepted from those that cannot, the representatives from the client are asked to rank the assets according to their importance (1= very important, 5= minor importance) and fill in the asset table (Table 2). Then the final treatment step can address the risks for the most important asset first.

Table 2 – Asset table

Asset	***Importance***	***Type***
Health records	*2*	*Direct asset*
Provision of telecardiology service	*3*	*Direct asset*
Public's trust in system	*(Scoped out)*	*Indirect asset*
Patient's health	*1*	*Indirect asset*

Having finished the asset table, they go on to define the likelihood scale (a general description of frequency or probability [3]) of which incidents occur, and the impact or consequence they have on the assets. The analysts initiate the discussion by suggesting a scale of likelihood based on the following rule of thumb: The lower incident likelihood "rare" is set to be maximum one occurrence during the target's lifetime; the remaining intervals have an increasing number of expected events until the maximum possible number of incidents per year is reached. Because assets of different types are involved, they make separate consequence scales for each of the direct assets. Table 3 shows the consequence scale defined for the asset "Health records" in terms of number of health records affected. If feasible the consequence description for an asset may include more than one measure, e.g. "major" could

be the number of disclosed health records, or the number of deleted records etc. Table 4 gives the likelihood scale defined for the target as such. By using the same scale for all scenarios and incidents, it is possible to extract combined likelihood values as shown later in the risk estimation step.

Table 3 – Consequence scale for "health records"

Consequence value	***Description***
Catastrophic	1000+ *health records (HRs) are affected*
Major	100-1000 *HRs are affected*
Moderate	10-100 *HRs*
Minor	1-10 *HRs are affected*
Insignificant	*No HR is affected*

Table 4 – Likelihood scale

Likelihood value	***Description***[3]
Certain	*Five times or more per year* (50-*: 10*y* = 5-*: 1*y*)
Likely	*Two to five times per year* (21-49: 10*y* = 2,1-4,9: 1*y*)
Possibly	*Once a year* (6-20: 10*y* = 0,6-2: 1*y*)
Unlikely	*Less than once per year* (2-5: 10*y* = 0,2-0,5: 1*y*)
Rare	*Less than once per ten years* (0-1:10*y* = 0-0,1:1*y*)

Finally, the representatives of the client need to define the risk evaluation criteria, the criteria which assert whether a risk to an asset is acceptable or whether it is necessary to evaluate possible treatments for it. They define these criteria by means of a risk evaluation matrix for each asset. The security analysis leader draws the matrix for the asset "Health records" on a blackboard. It has likelihood and consequence values as its axes so that a risk with a specific likelihood and consequence will belong to the intersecting cell. Based on a discussion in the group, the security analysis leader marks the cells in the matrix as "acceptable" or "must be evaluated". The resulting risk evaluation matrix is shown below (Table 5), and the participants decides to let this matrix cover the other assets as well.

Table 5 – Risk evaluation matrix

		Consequence				
		Insignificant	*Minor*	*Moderate*	*Major*	*Catastrophic*
Frequency	*Rare*					
	Unlikely	Acceptable				
	Possible					
	Likely				Must be evaluated	
	Certain					

After completing this task for all assets the analysts and the participants have the framework and vocabulary they need to start identifying threats (a potential cause of an unwanted incident [12]), vulnerabilities (weaknesses which can be exploited by one or more threats [12]), unwanted incidents and risks, and can move on to the next step.

Step 3 - summary

Tasks:	**Participants:**
• The client approves target descriptions and asset descriptions. • The assets should be ranked according to importance. • Consequence scales must be set for each asset within the scope of the analysis. • A likelihood scale must be defined. • The client must decide risk evaluation criteria for	• The same as in the previous meeting, but since this step sets the boundaries for the further analysis it is important that the relevant decision-makers are present.

[3] 50 - *:10*y* is short for 50 or more incidents per 10 years, equivalent to 5 or more incidents per year.

each asset within the scope of the analysis.	

Step 4: Risk identification

To identify risks CORAS makes use of a technique called structured brainstorming. Structured brainstorming may be understood as a structured "walk-through" of the target of analysis and is carried out as a workshop. The main idea of structured brainstorming is that since the analysis participants represent different competences, backgrounds and interests, they will view the target from different perspectives and consequently identify more, and possibly other, risks than individuals or a more homogeneous group would have managed.

The findings from the brainstorming are documented with the CORAS security risk modelling language. We will now exemplify how we model risks with the CORAS language, using the symbols presented in Figure 7.

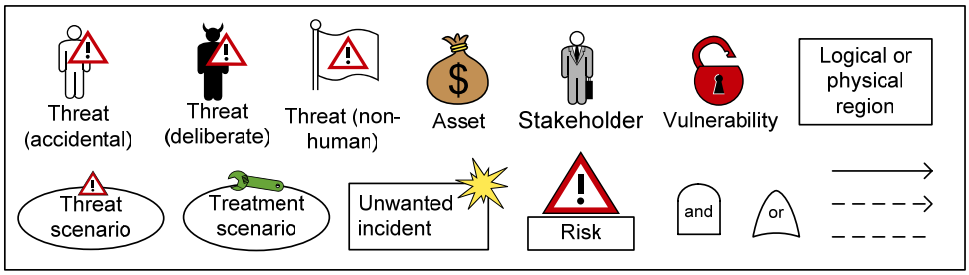


Figure 7 – Symbols from the CORAS risk modelling language

The analysis leader challenges the participants to work with questions like: What are you most worried about with respect to your assets? (Threat scenarios and unwanted incidents.) Who/what may initiate these? (Threats.) What makes this possible? (Vulnerabilities.) This information is modelled by the secretary in threat diagrams.

The analysis leader has used this technique on numerous occasions before. He does not use exactly the same procedure in every case, but adapts it to fit the target domain. Often he finds it useful to include checklists and "best practices" for a specific technology or domain. In this case he needs IT experts and medical personnel (general practitioners) to participate in the brainstorming, but some will only participate when their competences are needed for specific scenarios. Since people may be involved at different stages of the analysis, it is essential that information gathered during this session is documented in a simple and comprehensive way.

The analysis leader uses the target models from Step 2 (Figure 2, Figure 3, Figure 4 and Figure 5) as input to the brainstorming session. The models are assessed in a stepwise and structured manner and the identified unwanted incidents are documented on-the-fly (using the guidelines presented in the summary).

A set of initial threat scenario diagrams (Figure 8, Figure 9, Figure 10) has been prepared by the analysis secretary on the basis of the high level analysis table (Table 1). These represent a starting point for discussion and are often underspecified. She has decided to structure the three diagrams according to the categorisation in [12], describing the different types of threats: human accidental, human deliberate and non-human threats (environmental).

The threat diagram in Figure 8 shows how a combination of insufficient training or prose-based health records, and sloppiness may compromise the integrity and confidentiality of the patient's health records. The system also allows for irregular handling of health records

where an employee accidentally may cause a leakage of records. A confidentiality or integrity breach may harm the health record in the sense that it no longer is secret or correct. In the outmost consequence a faulty health record may affect the patient's health.

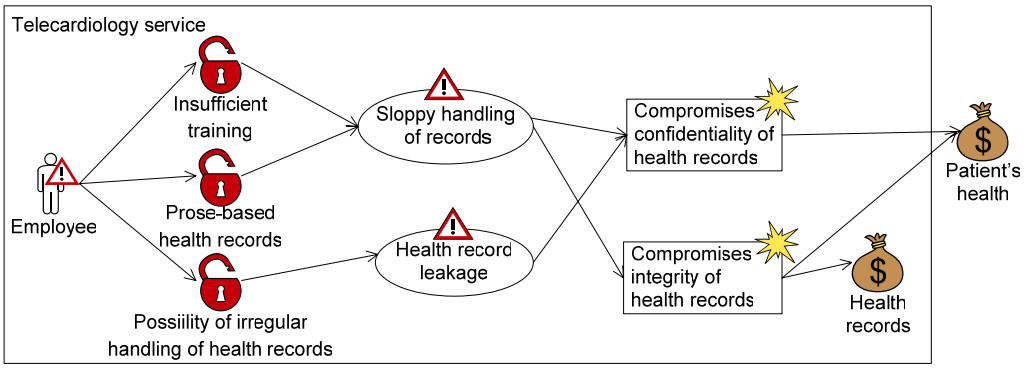


Figure 8 – Initial threat diagram: accidental actions

In the threat diagram describing deliberate harmful actions caused by humans, the participants have identified two main threats: hacker and eavesdropper (Figure 9). A hacker may exploit insufficient security mechanisms to break into the system and steal health records. An eavesdropper is a person that, due to insufficient protection of communication lines, may gather data that is transmitted and thereby compromise its confidentiality.

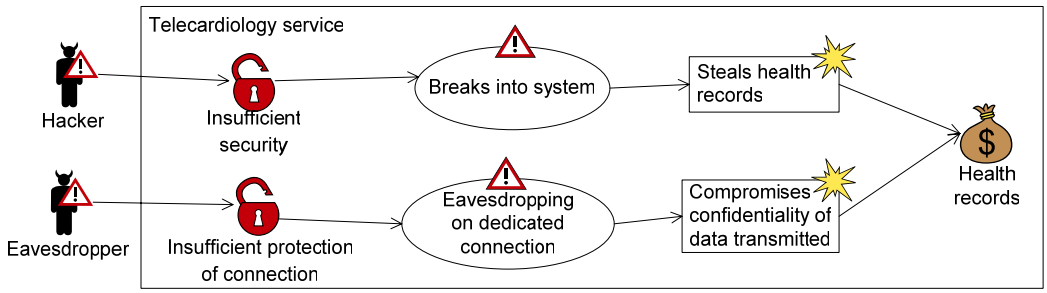


Figure 9 – Initial threat diagram: deliberate actions

The participants also worry about threats like system failure and network failure (Figure 10). They fear that unstable connections or immature technology are vulnerabilities that may lead to system crashes during examination or transmission problems. A transmission problem may interfere with the data that is stored in the system and leave the health records only partly correct.

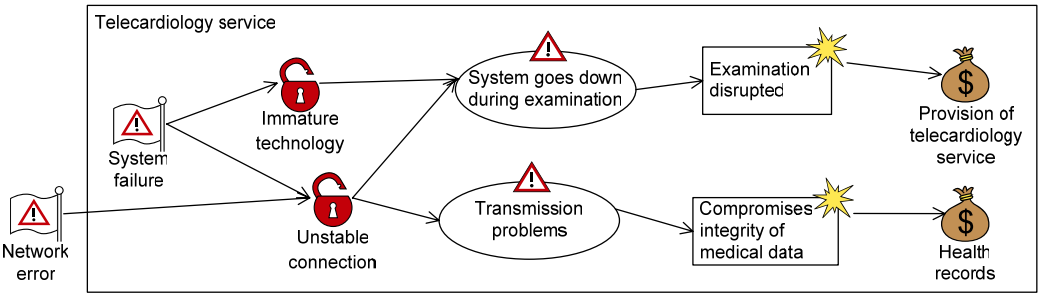


Figure 10 – Initial threat diagram: non-human threats

During the brainstorming session the initial threat diagrams are expanded with new information on-the-fly. If the amount of information is too large, the secretary may choose to write it down or use audiovisual equipment to make sure that nothing is missed. The diagrams may then be updated and completed after the session. The threat diagram illustrating incidents caused by employees' accidental actions (Figure 8) receives much attention among

the participants and develops into Figure 11. Due to space limitations we will not explore the other two threat diagrams further, but concentrate on just this one.

The participants decide that the threat "Employee" must be specified into "general practitioner (GP)" and "IT personnel" since they may cause different incidents. If the GP has too little security training, she may store copies of health records on a local computer. This may compromise the integrity of the records and in the worst case lead to an erroneous diagnosis of a patient. The same incidents may also occur if the GP enters wrong information in the patient's health record. The system allows for irregular handling of health records which makes it possible to accidentally send records to unauthorised people. This would compromise the confidentiality of the health record. The policy of the IT personnel with respect to access control has been very "loose". They explain this with their responsibility for making critical updates in emergencies and that they do not have the time to wait for a person with correct access rights to show up. An unfortunate consequence of this is that sometimes people without the required competence become responsible for critical changes. This may lead to misconfiguration of the system, which again may slow it down. A slow system may make it impossible to set a patient's diagnosis, and also the ability of providing a telecardiology service.

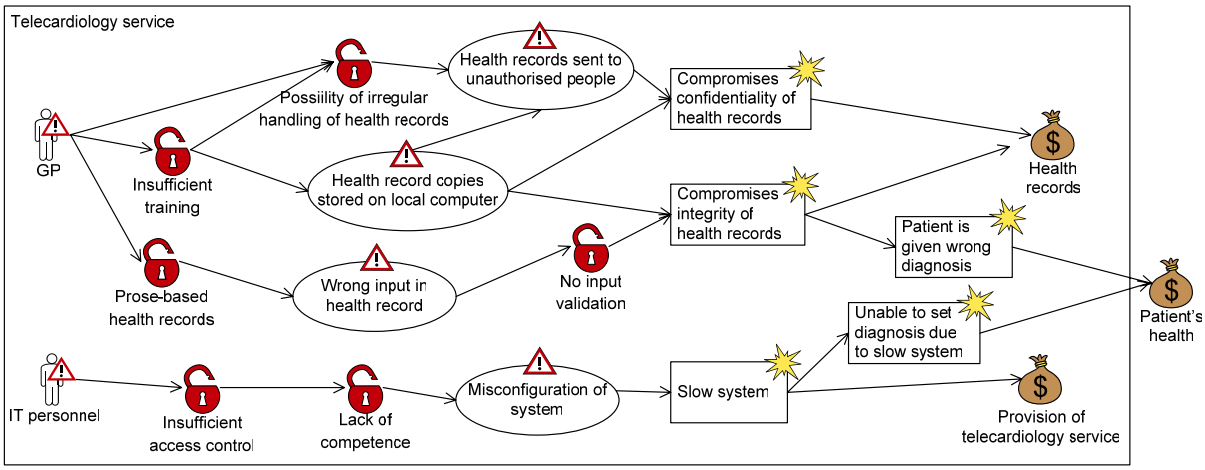


Figure 11 – Final threat diagram: accidental actions

Step 4 - summary

Tasks:	**People that should participate:**

• The initial threat diagrams should be completed with identified threats, vulnerabilities, threat scenarios and unwanted incidents.	• Security analysis leader (required)
	• Security analysis secretary (required)
	• Representatives of the client:
	  ○ Decision makers (optional*)
	  ○ Technical expertise (required)
	  ○ Users (required)
	*because this workshop often has a technical focus and the decision makers' competence is more relevant in the next step.

Modelling guideline:	
Threat diagrams: 1. Use the region from the asset diagram and add more regions if necessary. 2. Model different kinds of threats in separate diagrams, e.g. deliberate sabotage in one diagram, mistakes in an other, environmental in a third etc. ([12] contains a useful classification). This makes it easier to generalise over the risks, e.g. "these risks	5. Unwanted incidents are placed within the region with relations to the assets they impact. 6. Assets that are not harmed by any incidents are removed from the diagram. 7. Add threat scenarios between the threats and the unwanted incidents in the same order as they occur in real time (i.e. in a logical sequence). 8. Insert the vulnerabilities before the threat scenario

are caused by deliberate intruders" or "these risks are caused by human errors".
3. Threats are placed to the left in the region; threats that can be classified as external (hackers, intruders etc.) are placed outside the region.
4. Assets are listed to the right, outside the region.

or unwanted incident they lead to. E.g.: a vulnerability called "poor backup solution" is typically placed before the threat scenario "the backup solution fails to run the application database correctly".

Step 5: Risk estimation

When the threat scenarios, unwanted incidents, threats and vulnerabilities are properly described in threat diagrams it is time to estimate likelihood values and consequences. This is typically done in a separate workshop. The values are used to compute the risk value which decides whether the risk should be accepted or evaluated for treatments. The participants in the workshop provide likelihood estimates for each threat scenario in the threat diagrams. For scenarios that are difficult to estimate the analysis leader gives suggestions based on historical data like security incident statistics or personal experience. The likelihood of the threat scenarios are used to extract a combined likelihood for unwanted incidents. Consequences are estimated for each "unwanted incident – asset" relation. The consequence value is taken from the consequence scale of the asset decided in Step 3. In this workshop it is especially important to include people with the competence needed to estimate realistic likelihoods and consequences, meaning that technical expertise, users and decision makers must be included.

The analysis leader organises the estimation as a separate workshop where the input is the threat diagrams from the previous workshop. He knows that in this workshop it is especially important to include users, technical experts and decision makers to obtain estimates that are as correct as possible. The analysis participants decide that "most likely" estimates will provide more realistic risk values than "worst case" estimates. First they provide as many estimates as possible for the threat scenarios which help estimating the likelihood of the unwanted incidents (if this cannot be established by other means). Second, the consequences of the unwanted incidents for each harmed asset are estimated. The estimates are documented by annotating the diagrams as shown in Figure 12, further details can be specified in a table.

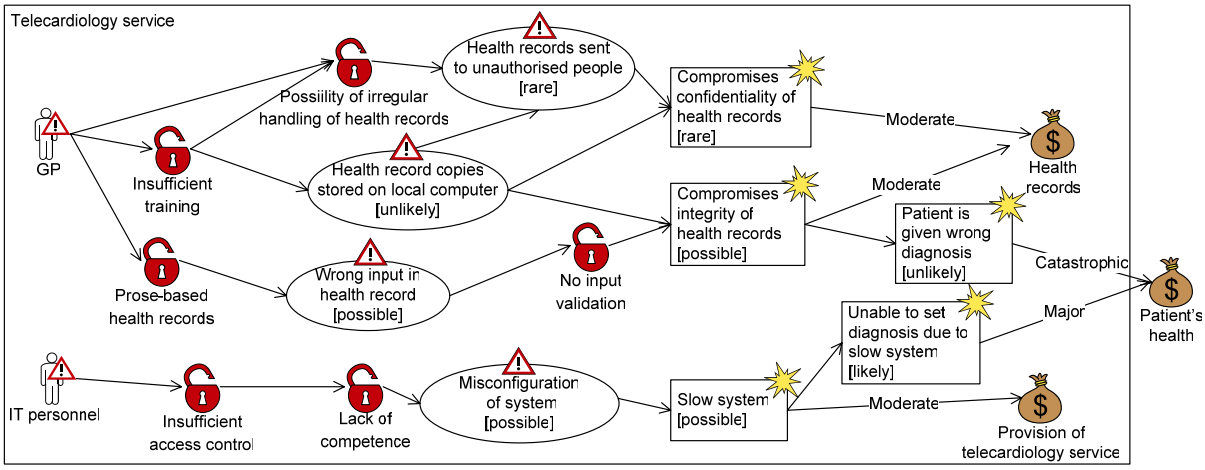


Figure 12 – Threat diagram with likelihood and consequence estimates

There are different ways of computing the likelihood of an incident that may be caused by more than one threat scenario. If the estimates are suitable for mathematical calculations a computerised tool may be used. Since the likelihood scale in our case is in the form of intervals, the analysis leader decides to use an informal method that is quite straight forward and transparent. The threat scenario "Health records sent out to unauthorised people" and "Health record copies stored on local computer" can both lead to "Compromises

confidentiality of health records". In Table 6 we show how the combined likelihood is estimated. The technique is informal, but suitable for the creative structured brainstorming setting. For more precise calculation of probabilities fault tree analysis (FTA)[11] may be used. It is of course important that the combined estimates reflect reality, meaning that the combined estimates should be presented to the participants for validation.

Table 6 – Combined likelihood estimates

Threat scenario	***Likelihood***	***Unwanted incident***	***Combined likelihood***
Health records sent out to unauthorised people	*Rare* (0-1:10*y*)	*Compromises confidentiality of health records*	(0-1:10*y*)+(2-5:10*y*)=(2-6:10*y*) *Some overlap between unlikely and possible, but it fits best in the unlikely interval.*
Health record copies stored on local computer	*Unlikely* (2-5:10*y)*		

In this case the participants reject the suggested estimate for "Compromises confidentiality of health records", arguing that the likelihood is less than "unlikely" and adjust it to "rare".

Step 5 - summary

Tasks:	**People that should be present:**
• Every threat scenario must be given a likelihood estimate and unwanted incident likelihoods are based on these. • Every relation between an unwanted incident and an asset must be given a consequence estimate.	• Security analysis leader (required) • Security analysis secretary (required) • Representatives of the client: o Decision makers (required) o Technical expertise regarding the target (required) o Users (required)
Modelling guideline:	
Risk estimation on threat diagrams: 1. Add likelihood estimates to the threat scenarios. 2. Add likelihood estimates to the unwanted incidents, based on the threat scenarios.	3. Annotate each unwanted incident-asset relation with a consequence taken from the respective asset's consequence scale.

Step 6: Risk evaluation

The risk evaluation consists of two activities. First the analysis secretary uses the likelihood and consequence estimates to compute the risk values and to place the risks in the risk matrix. Second, the resulting risk matrices are presented to the client for inspection. This presentation may be given in a separate meeting or included in the treatment workshop.

In our case the risk value is determined by the risk evaluation matrix. From the four unwanted incidents in the threat diagram, the analysis secretary extracts five risks. "Compromising the confidentiality of health records" (CC1) may affect health records. "Compromising the integrity of health records" may also harm health records (CI1), in addition to patient's health if it contributes to a faulty diagnosis (PR1). Finally, "slow system" may slow down an examination (SS2) and harm the patient's health (SS1). Only CC1 is within acceptable risk levels, the rest need further evaluation. Table 7 shows the risks placed in the risk evaluation matrix.

Table 7 – Risk evaluation matrix with risks

		Consequence				
		Insignificant	*Minor*	*Moderate*	*Major*	*Catastrophic*
Likelihood	*Rare*			CC1		
	Unlikely					PR1
	Possible			CI1, SS2		
	Likely				SS1	
	Certain					

The analysis leader gives the participants an opportunity to adjust likelihood and consequence estimates, and risk acceptance levels, to make sure that the results reflect reality as much as possible.

The participants request an overview of the risks. They want to know who, or what, is initiating them and which assets they harm. The analysis secretary models the risks with their associated risk values in a risk diagram according to the guidelines (see summary). The final risk diagram for unwanted incidents accidentally caused by employees is shown in Figure 13. Since the risk of compromising the confidentiality of health records is within the acceptable risk levels it will not be assessed in the treatment identification.

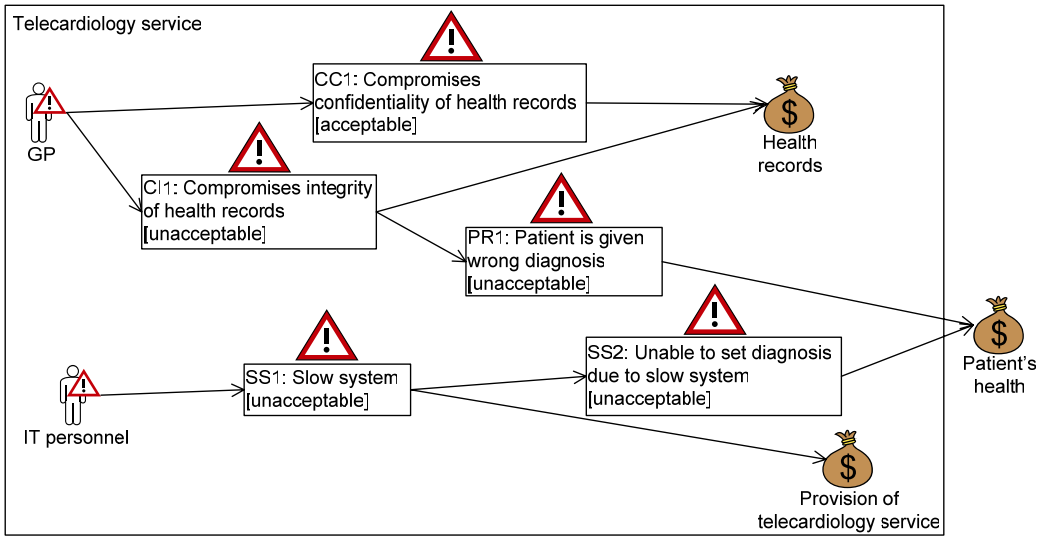


Figure 13 – Risk overview

Step 6 - summary

Tasks:	**People that should be present:**

• Likelihood and consequence estimates should be confirmed or adjusted.
• The final adjustments of the acceptable area in the risk matrices should be made (if needed).
• An overview of the risk may be given in a risk diagram.
	• Security analysis leader (required)
	• Security analysis secretary (required)
	• Representatives of the client:
	  ○ Decision makers (required)
	  ○ Technical expertise regarding the target (required/optional*)
	  ○ Users (required/optional*)
	*Depending on whether this step is included in step 7 or not. If part of step 7's workshop, all representatives should be present; otherwise it may be sufficient with only the decision makers.

Modelling guideline:	
Risk diagrams: 1. Use the threat diagram and replace all unwanted incidents with risk symbols, showing a short risk	3. If useful, split the risk diagrams into several diagrams according to type of threat, part of the target or asset importance (e.g. show all risks

description and whether the risk is acceptable or not. 2. Remove threat scenarios and vulnerabilities, but keep the relations between the threats and the risks.	related to network, all risks for specific assets etc.).

Step 7: Risk treatment

The last step in the security analysis is the treatment identification, which is also often organised as a workshop. The risks that are not acceptable are all evaluated in order to find means to reduce their likelihood and/or consequence. Since treatments can be costly, they are assessed with respect to their cost/benefit, before a final treatment plan is made. The initial treatment diagrams are similar to the final threat diagrams except that every relation between an unwanted incident and an asset representing an unacceptable risk is symbolised with a risk icon and an identifier.

The analysis leader presents each of the threat diagrams showing the unacceptable risks. He knows that analysis participants often find it most intuitive to address vulnerabilities when looking for treatments. Hence, he highlights the possibility of treating other parts of the target as well, such as threats or threat scenarios. The participants involve in a discussion of potential treatments, and decide which ones will reduce the risks to acceptable levels. On some occasions, if focus is slightly out of scope, the analysis leader suggests treatments taken from best-practice descriptions for network solutions, encryption, etc. to help the discussion back on track. The diagrams are annotated with the identified treatment options indicating where they will be implemented. Finally, the following treatments are suggested and annotated to the treatment diagram in Figure 14:

- *Extend the training program for practitioners with 1-2 days, with a special focus on security aspects.*
- *Revise the list of people that have maintenance access, restrict access to only the users that have competence on critical configuration tasks.*

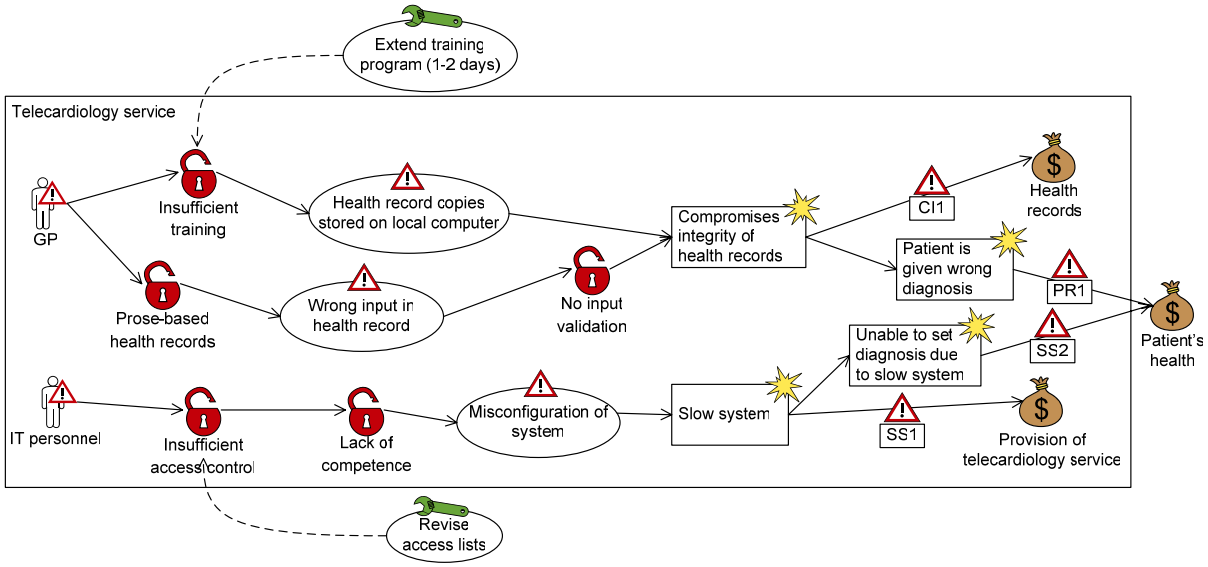


Figure 14 – Treatment diagram

When the final results from the analysis are to be presented to the client and other interested parties, an overview of the risks and the proposed treatments is useful. In our case the treatment overview diagram of Figure 15 is used for this purpose.

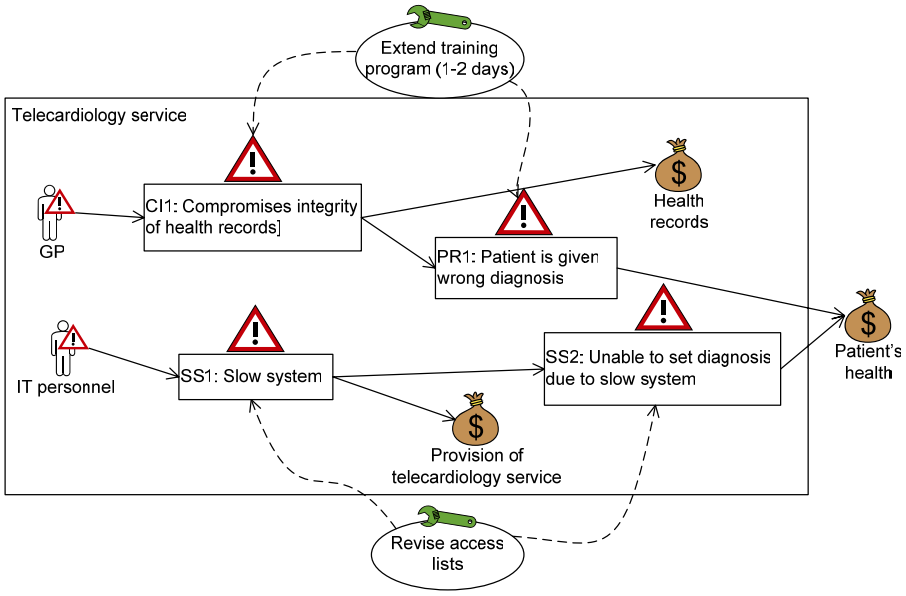


Figure 15 – Treatment overview diagram

Step 7 - summary

Tasks:	**People that should be present:**

• Add treatments to threat diagrams.
• Estimate the cost/benefit of each treatment and decide which ones to use.
• Show treatments in risk overview diagrams.
	• Security analysis leader (required)
	• Security analysis secretary (required)
	• Representatives of the client:
	  ○ Decision makers (required)
	  ○ Technical expertise (required)
	  ○ Users (required)

Modelling guidelines:	
Treatment diagrams: 1. Use the threat diagrams as a basis and annotate all arrows from unwanted incidents to assets with risk icons. Show only the unacceptable risks. 2. Annotate the diagram with treatments, pointing to where they will be applied.	Treatment overview diagrams: 1. Use the risk diagrams as a basis, remove the acceptable risks. 2. Add treatments according to the treatment diagram(s).

Summary and related work

This paper has demonstrated the use of the CORAS method in the security analysis of a telemedicine system. The main focus has been on the CORAS security risk modelling language and its role as a medium for communication and interaction during the various steps. In particular, the following core CORAS diagrams have been used:

- *Asset diagrams* to capture and relate the main assets to be protected.
- *Threat diagrams* to show the most relevant threats. We have used threat diagrams to investigate how employees accidentally may cause risks. The threat diagrams have also been used to capture likelihood and consequence estimates for each risk.
- *Risk diagrams* to show an overview of which risks that are acceptable and which are not. The overview also shows which threats that are involved and which assets they may harm.
- *Treatment diagrams* to capture treatment suggestions for the unacceptable risks. The final treatments are also shown in a *treatment overview diagram* suitable for presentations.

This paper has by no means presented all aspects of CORAS. In particular, the CORAS security risk analysis and modelling tool has not been covered.

The CORAS method and language have been developed iteratively based on experiences from numerous industrial field trials and empirical investigations. The field trials were carried out within the setting of the research project SECURIS (152839/220). Each field trial (in the form of a security analysis) was set to about 250 person hours from the analysts and 50-100 hours from the client. The clients and scopes for the analyses were:

- 2003: NetCom (telecom), information sharing service [5].
- 2004: DNV (vessel classification), information sharing service.
- 2004: NetCom (telecom), mobile access to personal information.
- 2005: Statnett (energy), control and supervisory system.
- 2005: Hydro Aluminium (metal production), control and supervisory system.
- 2006: Hydro (corporate), document management system.

A number of empirical investigations have also been carried out in order to improve the method and the language. These included:

- An experiment on the use of special icons in a modelling language, and how it helps people understand the models better. This experiment also included an initial exploration of the language's conceptual foundation using master students in software engineering [7].
- A survey of security risk analysis terminology that helped us to identify the intuitive and more problematic parts of the language's conceptual foundation. The subjects were professionals and master students within software engineering [8].
- An experiment that identified the preferred modelling alternatives among professionals within software and security [6].

The CORAS method follows the structure of the Australian/New Zealand Standard for Risk Management [3]. The originality of CORAS lies in the strong focus on graphical models during structured brainstorming, both for risk identification and documentation. Compared to two major alternative risk analysis methods CRAMM [4] and OCTAVE [1], the model orientation, instead of just text and table-based documentation, is the main difference.

The structured brainstorming activity in CORAS is inspired by HazOp [21], but developed further to make use of system models as input to the analysis, and threat and risk models as support during the analysis and documentation of outcome.

Misuse cases [2, 23, 24], a variation on UML use cases [13, 20] that characterises what a system should not allow, was an important source of inspiration in the development of the CORAS language. The first versions of the CORAS language was in the form of a UML profile [17, 19] (i.e. a specialisation of UML) based on use cases. As a result of our work to satisfy the modelling needs in security risk analyses the language has evolved into a more specialised and refined language. Other UML based and security oriented languages exist, e.g. UMLsec [14] and SecureUML [15], but they focus on specifying security properties of systems rather than documenting security risks.

Various tree-based notations that support security risk analysis, especially for calculating likelihood, have been around for several years. Most notably is fault trees [11] (and variants like attack trees [22] and threat trees [9]) and event trees [10]. Fault trees are based on decomposing an unwanted incident into its sub-events and use this information to calculate the probability of the top event. With event trees the aim is to model how an unwanted incident may propagate and find the probability and consequence of events originating from

the initial event. Cause-consequence diagrams [18] may be seen as the combination of fault trees and event trees. The CORAS diagrams can be said to be inspired by these notations, but do not require input in terms of exact probabilities.

What characterises CORAS is the focus on security risk of information systems, the heavy use of models to guide and structure the analysis, and the specialised language for documenting and communicating intermediate as well as the final results of the analysis. In addition, CORAS comes with a computerised tool that fully integrates all the steps of the analysis from initial planning to production of final reports.

Acknowledgments

The CORAS method has mainly been developed in the 5th Framework IST project CORAS (IST-2000-25031) and the Research Council of Norway projects SECURIS (152839/220) and COBRA (152209/431). Some of the work has also been founded by 6th Framework IST projects TrustCoM (01945) and MODELWARE (511731). The authors acknowledge and thank collaboration partners within these projects for stimulating input and feedback since the start-up of the CORAS project in 2001. The authors have also benefited considerably from teaching this material at the University of Oslo and receiving the students' response.

References

[1] Alberts, C., J. ; and Dorofee, A. J., "OCTAVE Criteria Version 2.0", Tech. report CMU/SEI-2001-TR-016. ESC-TR-2001-016, 2001.

[2] Alexander, I., "Misuse cases: Use cases with hostile intent", *IEEE Software*, vol. 20 (1), pp. 58-66, 2003.

[3] AS/NZS4360, Australian/New Zealand Standard for Risk Management: Standards Australia/Standards New Zealand, 2004.

[4] Barber, B. and Davey, J., "The Use of the CCTA Risk Analysis and Management Methodology CRAMM in Health Information Systems ", in Proc. MEDINFO'92, pp. 1589-1593, 1992.

[5] den Braber, F., Mildal, A.-B., Nes, J., Stølen, K., and Vraalsen, F., "Experiences from Using the CORAS Methodology to Analyze a Web Application", *Journal of Cases on Information Technology*, vol. 7 (3), pp. 110-130, 2005.

[6] Hogganvik, I. and Stølen, K., "A Graphical Approach to Risk Identification, Motivated by Empirical Investigations", in Proc. MoDELS'06 LNCS (4199), pp. 574-588, 2006.

[7] Hogganvik, I. and Stølen, K., "On the Comprehension of Security Risk Scenarios", in Proc. Int. Workshop on Program Comprehension (IWPC'05), pp. 115-124, 2005.

[8] Hogganvik, I. and Stølen, K., "Risk Analysis Terminology for IT-systems: does it match intuition?" in Proc. Int. Symposium on Empirical Software Engineering (ISESE'05), pp. 13-23, 2005.

[9] Howard, M. and LeBlanc, D., *Writing Secure Code*, 2 ed: Microsoft Press, 2003.

[10] IEC60300-3-9, Event Tree Analysis in Dependability management - Part 3: Application guide - Section 9: Risk analysis of technological systems., 1995.

[11] IEC61025, Fault Tree Analysis (FTA), 1990.

[12] ISO/IEC13335, Information technology - Guidelines for management of IT Security, 1996-2000.

[13] Jacobson, I., Christerson, M., Jonsson, P., and Övergaard, G., *Object-Oriented Software Engineering: A Use Case Driven Approach*: Addison-Wesley, 1992.

[14] Jürjens, J., *Secure Systems Development with UML*: Springer, 2005.

[15] Lodderstedt, T., Basin, D., and Doser, J., "SecureUML: A UML-Based Modeling Language for Model-Driven Security", in Proc. UML'02, LNCS (2460), pp. 426-441, 2002.
[16] Lund, M. S., den Braber, F., Stølen, K., and Vraalsen, F., "A UML profile for the identification and analysis of security risks during structured brainstorming." SINTEF ICT, Tech. report STF40 A03067, 2004.
[17] Lund, M. S., Hogganvik, I., Seehusen, F., and Stølen, K., "UML profile for security assessment", SINTEF Telecom and Informatics, Tech. report STF40 A03066, 2003.
[18] Nielsen, D. S., *The Cause/Consequence Diagram Method as a Basis for Quantitative Accident Analysis*: Danish Atomic Energy Commission, RISO-M-1374, 1971.
[19] OMG, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms", Object Management Group 2006.
[20] OMG, "Unified Modeling Language (UML): Superstructure, version 2.0", Object Management Group 2005.
[21] Redmill, F., Chudleigh, M., and Catmur, J., *HAZOP and Software HAZOP*: Wiley, 1999.
[22] Schneier, B., "Attack trees: Modeling security threats", *Dr. Dobb's Journal*, vol. 24 (12), pp. 21-29, 1999.
[23] Sindre, G. and Opdahl, A. L., "Eliciting Security Requirements by Misuse Cases", in Proc. TOOLS-PACIFIC, pp. 120-131, 2000.
[24] Sindre, G. and Opdahl, A. L., "Templates for Misuse Case Description", in Proc. Workshop of Requirements Engineering: Foundation of Software Quality (REFSQ'01), pp. 125-136, 2001.
[25] Stathiakis, N., Chronaki, C., Skipenes, E., Henriksen, E., Charalambous, E., Sykianakis, A., Vrouchos, G., Antonakis, N., Tsiknakis, M., and Orphanoudakis, S., "Risk assessment of a cardiology eHealth service in HYGEIAnet." in Proc. Computers in Cardiology (CIC'2003), pp. 201-204, 2003.

Appendix F

Assessing Enterprise Risk Level: The CORAS Approach

Chapter XVIII
Assessing Enterprise Risk Level: The CORAS Approach

Fredrik Vraalsen
SINTEF, Norway

Tobias Mahler
Norweigan Research Center for Compuers and Law, University of Oslo, Norway

Mass Soldal Lund
SINTEF, Norway

Ida Hogganvik
SINTEF, Norway

Folker den Braber
SINTEF, Norway

Ketil Stølen
SINTEF, Norway

ABSTRACT

This chapter gives an introduction to the CORAS approach for model-based security risk analysis. It presents a guided walkthrough of the CORAS risk analysis process based on examples from risk analysis of security, trust and legal issues in a collaborative engineering virtual organisation. CORAS makes use of structured brainstorming to identify risks and treatments. To get a good picture of the risks, it is important to involve people with different insight into the target being analysed, such as end users, developers, and managers. One challenge in this setting is to bridge the communication gap between the participants, who typically have widely different backgrounds and expertise. The use of graphical models supports communication and understanding between these participants. The CORAS graphical language for threat modelling has been developed especially with this goal in mind.

Copyright © 2008, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

Businesses face an increasing number of security risks in the online world, not limited to those of a technical nature. At the enterprise level, technical aspects of security are tightly interwoven with other aspects such as trust and legal issues. This is particularly true for the new breed of networked, virtual organisations. A virtual organisation (VO) can be understood as a temporary or permanent coalition of geographically dispersed individuals, groups, organisational units or entire organisations that pool resources, capabilities and information to achieve common objectives (Dimitrakos, Goldby, & Kearney, 2004).

Virtual organisations' dependency on information and communication technology for performing their daily work leads to a number of risks related to security, trust and legal issues. One area where VOs face risks is the protection of intellectual property (IP) and confidential information, which is the focus of the case study presented in this chapter. Confidentiality is the property that information is not made available or disclosed to unauthorised individuals, entities or processes (ISO/IEC 13335, 2004). The individual stakeholders in a VO desire to protect their IP and maintain confidentiality of information, but at the same time they need to share some of this information with the other partners in the VO in order to fulfil common objectives as well as specific obligations laid down in a co-operation contract. The risks are exacerbated by the international nature of many VOs, as well as their dynamic nature where participants can join and leave the VO at any point during its lifetime.

There is no general international legal framework for the establishment and operation of virtual organisations, and legal issues in relation to VOs are still a topic for research. A strategic roadmap for advanced virtual organisations points out that the analysis of legal risks arising from operating VOs, and the development of legal strategies to overcome them, is an important research task in order to support collaborative networked organisations (Camarinha-Matos et al., 2004). Such legal strategies for VOs should focus both on the contracts that need to be put into place and on the technology that may be utilised in order to facilitate and support the collaboration. When drafting the VO collaboration contract, parties need to identify and address risks that may arise from the collaboration. This risk analysis should preferably follow a clear methodology. Some approaches have considered project risk management in a more general setting (Baccarini & Archer, 1999; Raz & Michael, 1999), focusing mainly on the risk of project failure. However, collaborators also need to assess risks to their own assets.

To reduce the risks involved with establishing, joining and operating a VO, an approach for analysing and managing enterprise security risks is needed which takes into account both technical and nontechnical aspects. The collaboration of different experts, like computer scientists and lawyers, is necessary when analysing what may go wrong in a co-operation (Heymann, 2005; Müller-Hengstenberg, 2005). The CORAS model-based risk analysis approach facilitates the integration of these different perspectives, and focuses also on incorporating the context of the system into the analysis, that is, the organisations, processes and people which interact with the system.

CORAS is a framework for model-based security risk analysis. This framework consists of a method, a language, and a computerised tool. The method integrates aspects from different risk analysis techniques with state-of-the-art system modelling methods based on UML 2.0 (OMG, 2005), the de facto standard modelling language for information systems. The CORAS graphical language for threat modelling is an extension of the UML 2.0 specification language. It is defined as a UML profile (Lund, Hogganvik, Seehusen, & Stølen, 2003), and has recently become part of an OMG standard (OMG, 2006).

The goal of this chapter is to make the reader familiar with the CORAS method for model based

risk analysis as well as the graphical language used for threat modelling, and explain how they may be employed in the analysis of VOs. The next section provides more background on the CORAS approach, followed by a walkthrough of the CORAS risk analysis process. Finally, we present some concluding remarks.

The example case is based on a risk analysis which was performed in the TrustCoM IST project (*http://www.eu-trustcom.com/*) using CORAS (Mahler, 2005). The analysis focuses on a collaborative engineering project in the aerospace industry, where a group of companies establish a VO to collaborate on the upgrade of an airplane design. The focus of the analysis is on intellectual property rights (IPR) and confidentiality issues in relation to the sharing of trade secrets between the partners of the virtual organisation. Hence, the analysis is also an attempt to contribute to the investigation of methods for legal risk management, which are "in their infancy" (Burnett, 2005).

THE CORAS APPROACH

The CORAS risk management method is based on the AS/NZS 4360 standard for risk management (AS/NZS 4360, 2004). *Risk management* is the sum of the culture, processes and structures that are directed towards effective management of potential opportunities and adverse effects. The *risk management process* consists of systematic application of management policies, procedures and practice to the tasks of establishing the context and identifying, analysing, evaluating, treating, monitoring and communicating risks. Risk management thus covers the entire life cycle of the system or organisation, and may include several risk analyses with different focus areas and abstraction levels as the system or organisation and its surroundings evolve over time.

Risk analysis requires a clear understanding of the system or organisation to be analysed. This can only be achieved through the involvement of stakeholders and other interested parties with different backgrounds and knowledge about the system or organisation (e.g., decision makers, security experts, legal experts, system owners, developers, and users). These participants are involved in the identification and evaluation of risks and treatments through structured brainstorming sessions.

The effectiveness of such sessions depends on the extent to which the participants are able to communicate with and understand each other. The CORAS graphical language for threat modelling (den Braber, Lund, Stølen, & Vraalsen, 2005) has been designed to mitigate this problem within the security domain. The CORAS language covers notions like asset, threat, risk and treatment, and supports communication among participants with different backgrounds through the definition of easy-to-understand symbols associated with the modelling elements of the language. A recent study has shown that the graphical symbols allow the participants to understand and read the diagrams more quickly (Hogganvik & Stølen, 2005). Recent work has also focused on application of the CORAS language and method to the analysis of security, trust and legal issues (Brændeland & Stølen, 2004; Mahler & Vraalsen, 2005; Vraalsen, Lund, et al. 2005), as well as continuous improvements of the language based on experiences from use and empirical investigations.

Figure 1 shows the main elements of a risk analysis and gives examples of the graphical symbols used by the CORAS language. The *target* is the system or organisation, or parts thereof, which is the focus of the analysis. *Assets* are the parts or features of the target which have value to the client commissioning the analysis, such as physical objects, know-how, services, software and hardware, and so on. A *vulnerability* is a weakness of the system or organisation. A *threat* may exploit a vulnerability and cause an *unwanted incident*, an event which reduces the value of one or more of the assets. A *risk* is an unwanted incident along with its estimated likelihood and

Figure 1. Elements of a risk analysis

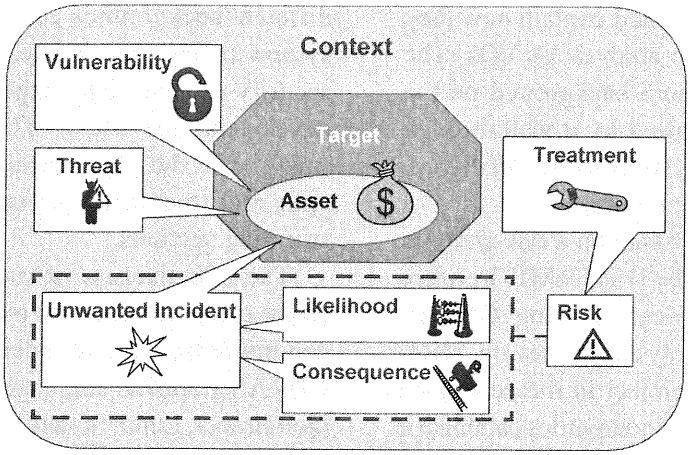


Figure 2. CORAS meetings

Meeting 1: Introduction
- Clients present the system or organisation they wish to analyse
- Identify the focus and scope for the analysis
- Set up analysis plan

Meeting 2: High-level analysis
- Risk analysts present their understanding of the target of analysis
- Identify assets
- Establish initial threats and vulnerabilities

Meeting 3: Approval
- Target of analysis documentation
- Assign values to assets
- Identify risk evaluation criteria

Meeting 4: Risk identification
- Identify risks through structured brainstorming

Meeting 5: Risk estimation and evaluation
- Estimate likelihood and consequence of risks
- Evaluate risks with respect to risk evaluation criteria

Meeting 6: Risk treatment
- Identify and evaluate treatments

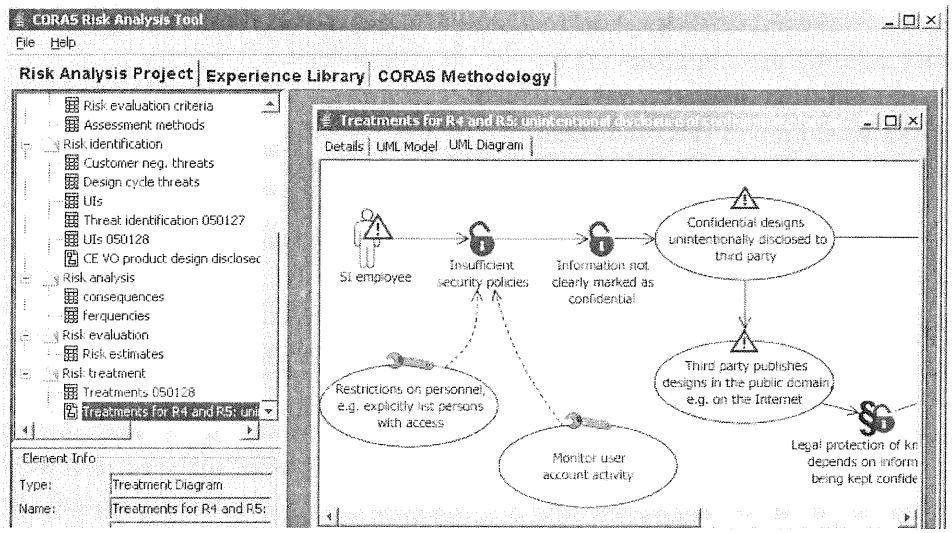
Meeting 7: Finalisation meeting (if necessary)
- Present results and get any missing input

consequence values. *Treatments* represent various options for reducing risk.

The CORAS risk analysis process is typically organised as a set of meetings, as summarised in Figure 2. Between the meetings, the risk analysts need time to process the collected information, gather additional necessary documentation, and prepare for the next step of the analysis.

The meeting schedule should be tailored to the needs of each individual risk analysis. Not all activities need to be conducted as face-to-face meetings, but may be performed through for ex-

Figure 3. The CORAS tool



ample phone or video conferences or via e-mail discussions. Some meetings may be combined to save time or costs. For instance, in several cases we have combined meetings four through six into a two or three day workshop in order to reduce travel costs for the involved participants. This requires careful planning and preparation, however, as well as scheduling time during the workshop to give the analysts a chance to process the output, for example at the end of each day. On other occasions, additional meetings are needed, for example if new information comes up during the analysis which necessitates extra risk identification work to properly identify all the relevant risks.

Risk analysis is an elaborate and prolific process which involves many different types of documentation from different sources, such as UML models, tables with analysis data, and natural language descriptions of the target of analysis. All this information needs to be organised and accessible. In addition, it is important to maintain consistency between all the elements to prevent errors, and we also wish to be able to reuse elements from previous analyses where appropriate to avoid starting from scratch every time. Computerised support for documentation, maintenance and reuse of analysis results is thus of high importance.

The CORAS Tool (Vraalsen, den Braber, Lund, & Stølen, 2005) is a Java-based risk analysis tool which is publicly available as open source (*http://coras.sourceforge.net/*). The client-server architecture of the tool enables multiple risk analysts to collaborate on the risk analysis projects. The risk analyst uses the CORAS client application to create new analysis projects, document and edit risk analysis results in tables and diagrams, generate analysis reports, and manage and reuse experiences from previous analyses. Information can be imported from various modelling and risk analysis tools used by the analyst through standardised data exchange formats, such as XMI for UML. The tool also contains a built in diagram editor for the CORAS graphical language. Help is provided to the user in the form of integrated online versions of the CORAS method and user guides. A screenshot of the CORAS client application is shown in Figure 3.

In the following sections, we will present the risk analysis meetings and activities in more detail. This will be illustrated with examples taken from the TrustCoM risk analysis.

INTRODUCTORY MEETING

The introductory meeting aims at achieving an initial understanding of what the client wishes to have analysed and what kind of risks the client is most concerned about. Some of the questions that should be answered include:

- For whom is the analysis carried out?
- For what purpose do we perform this analysis?
- What do we want to protect?
- What is the scope?

An in-depth analysis can be a time consuming and costly process, and the client typically has limited resources available for risk management. By clearly characterising the target and focus of the study, including identifying what falls outside the scope of the analysis, the available resources can be utilised in the most effective and efficient manner.

During a risk analysis, we make several assumptions and choices with regard to the system or organisation under analysis as well as its surroundings. Documenting these choices and assumptions is necessary in order to determine in which contexts the analysis results are valid. As the system or organisation and its surroundings change over time, these assumptions may no longer hold true. In this case, the analysis may need to be updated to determine whether the risk level of any of the previously identified risks has changed and to identify any new risks which may have arisen. Mechanisms thus need to be put in place to monitor the risks and assumptions and determine when a new risk analysis is necessary.

The introductory meeting should include the risk analysts and the client of the analysis, typically represented by a person with decision making powers with respect to the system or organisation being analysed. The meeting may also involve other stakeholders or parties who have an interest in or knowledge about the system or organisation.

The risk analysts should give a brief presentation of CORAS to familiarise the client with the risk analysis process and some of the methods and techniques which may be used, such as structured brainstorming and the graphical language.

Client Presents System or Organisation

The client presents the system or organisation they wish to have analysed and what kind of incidents they are most worried about. This presentation will typically include a mix of text (prose, tables, etc.), informal diagrams, such as rich pictures (Checkland & Scholes, 1990), and models describing the system or organisation to be analysed. Depending on what the client wishes to analyse, this presentation would normally cover a number of different areas, such as business goals and processes, users and roles, contracts and policies, hardware and software specifications, network layout, and so on.

SI is a company specialising in the integration of different aircraft subsystems. SI wants to win a contract with an airliner for the upgrade of their business jets with a new feature–an in-flight entertainment system. In order to be able to fulfil this objective, it joins a group of aerospace companies to form a virtual organisation in order to pool their resources and know-how and have a better chance of winning the contract. However, before joining the VO, SI wants to perform a risk analysis in order to determine the potential risks involved in this venture, and hires a consultant company to carry out the analysis.

The three main actors in this business scenario are:

- The airliner that operates a fleet of business jets.

- The proposed collaborative engineering VO (CE VO) which has the technical expertise to specify, design and integrate systems into complex products, and which may also manufacture the solution for the customer. Three partners would be involved in this VO; an avionics manufacturer, an in-flight system entertainment provider, and the aforementioned system integrator (SI)—the client of the risk analysis.
- An analysis consultancy which support design activities within engineering companies by performing general analysis work across engineering and scientific sectors.

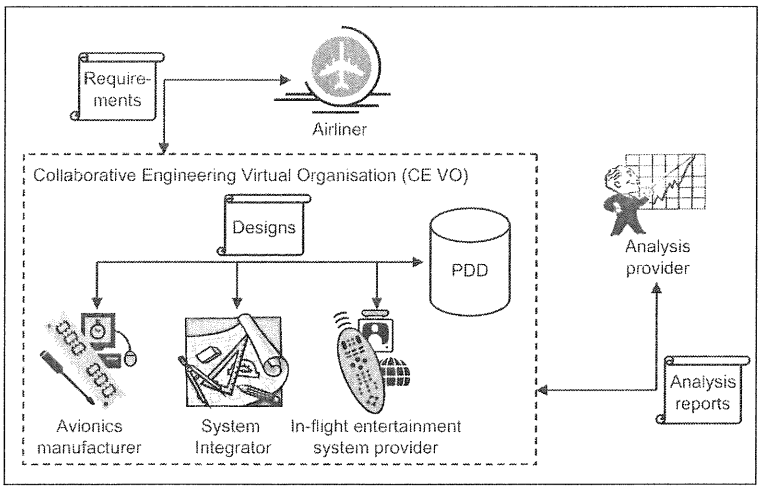
Figure 4 shows a diagram presented by SI, depicting the actors and their relationships. The various subsystem designs and integrated designs produced and shared during the design process are stored in the product design database (PDD).

Characterise Focus and Scope of the Analysis

The client and the risk analysts should characterise the focus and scope of the analysis. Characterising the focus and scope is important to ensure both a common understanding of the problem at hand and to ensure an efficient use of the available resources by focusing on the aspects of the system or organisation that are of real importance to the client. This includes defining the borders between what is to be part of the analysis (target) and what is to be left out. Part of defining the scope is selecting which security properties are to be considered in the analysis, such as confidentiality, integrity, and availability, as well as other aspects of interest. The risk analysts should interact with the client to clarify any questions or uncertainties with regards to the target of analysis to avoid misunderstandings later on.

The system integrator is particularly concerned about loss of intellectual property and confidential information and the possibility of industrial espionage in connection with exchange of information with other partners, both internal and external to the VO. Retaining confidentiality of the design information communicated with the partners and stored in the Product Design Database is therefore of utmost importance. To prevent other companies from competing with the CE VO proposal, it is also important to protect the confidentiality of the requirements which have been gathered from the airline through the discussions and initial design meetings.

Figure 4. Actors in CE scenario



To limit the size of the analysis, other aspects such as data integrity issues, for example, malicious modification or deletion of information because of industrial sabotage or for example virus attacks, are left outside the scope of this particular analysis.

Plan the Analysis

Finally, the rest of the risk analysis should be planned in more detail, including identifying participants and meeting times and venues, based for example on the suggested meeting schedule presented in Figure 2. To achieve continuity in the risk analysis process it is important that the core group of participants commit to the risk analysis and are able to participate during the whole process so that the risk analysts do not have to interact with new and different people at every meeting. Additional persons may be involved in the different meetings based on the competence which is required.

The risk analysis team typically consists of one or two risk analysts who perform the actual risk analysis. One risk analyst should be responsible for leading the risk analysis sessions, and an additional person may act as a secretary during the sessions, recording the results and assisting the risk analysis leader when necessary.

The analysis team should include a representative of the client with decision making power with regards to the target of analysis. In addition, it should include other stakeholders, domain experts, and other interested parties with knowledge about the target of analysis, such as system managers, developers, users, lawyers, security experts, and so on. The goal is to involve people with different backgrounds and different insight into the problem at hand in order to elicit as much relevant information about potential risks as possible. If the risk

Table 1. Risk analysis roles

Role	Name	Organisation	Background/Expertise
Risk analysis leader	Thomas	CORAS Ltd.	Risk analysis, security
Risk analysis secretary	Frank	CORAS Ltd.	Risk analysis, security
Target owner	David	AirFrame Inc.	Aerospace industry
Domain expert	Peter	EngiCorp	Engineering & design
Domain expert	Irene	U. of Oslo	Intellectual property law
Domain expert	Claire	U. of London	Socio-economy and trust

Table 2. Risk analysis plan

Date	Tasks	Participants
29th November	Target identification Asset identification	Analysis leader & secretary, legal expert
11th January	High-level analysis	Analysis leader & secretary, legal expert
27th January	Approval Risk identification	Whole risk analysis team
28th January	Risk estimation and evaluation Risk treatment	Whole risk analysis team
2nd February	Cleanup of results Risk analysis report	Analysis leader & secretary, legal expert

analysis team becomes large, it may be beneficial to split it into smaller groups during parts of the process (e.g., the brainstorming sessions described below). The point is to give everyone a chance to participate and feel useful, as well as to be able to control the group when necessary.

The risk analysis team consisted of two risk analysts with backgrounds in security. The risk analysis team also included two representatives from the client company, the project leader for the aircraft upgrade project and an engineer with good knowledge of the engineering design processes. In addition, it involved an IP lawyer and an expert on socio-economy and trust. The participants of the risk analysis are documented in the risk analysis roles table as shown in Table 1.

The risk analysis for SI was performed over the course of 2½ months. Because the participants were spread across several countries, the main part of the analysis was performed during a two day workshop involving the whole analysis team. Other activities were performed in smaller groups and through phone conferences and e-mail discussions. The plan for the analysis is summarised in Table 2.

HIGH-LEVEL ANALYSIS

One of the goals of the second meeting is to ensure a common understanding of the focus and scope of the analysis, as well as to identify the client's main assets in the system or organisation. Assets are central to the CORAS risk analysis method and help guide the entire risk analysis process. The assets are used to assist in identifying risks and estimating their consequences in terms of loss of (monetary) value of the different assets. A high level analysis of threats, vulnerabilities and unwanted incidents is performed to help identify what the client is most worried about happening, and thus to ensure a correct characterisation of the focus and scope of the analysis.

Risk Analysts Present Target of Analysis

Based on the background documentation from the client and the presentations and discussions from the introductory meeting, the risk analysts start by presenting their understanding of the target of analysis, inviting comments and cor-

Figure 5. Target of analysis

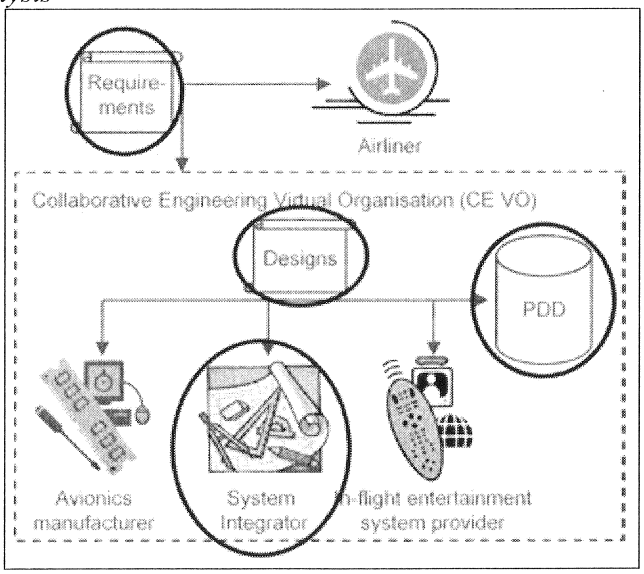
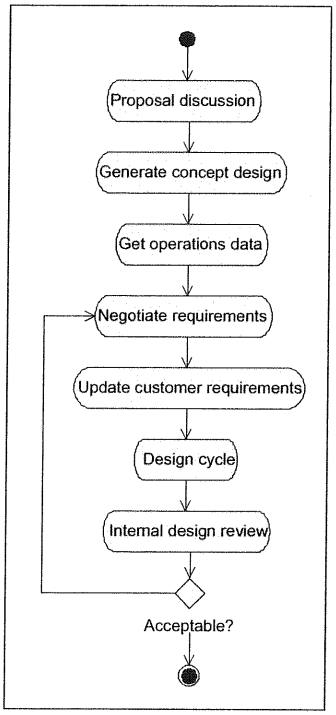


Figure 6. High-level CE VO design process



rections from the client. This is done to ensure a common understanding of what is to be analysed and what is to be considered outside the scope of the analysis. The target is characterised using for example UML diagrams or other types of models to specify the target and its relations with the surroundings.

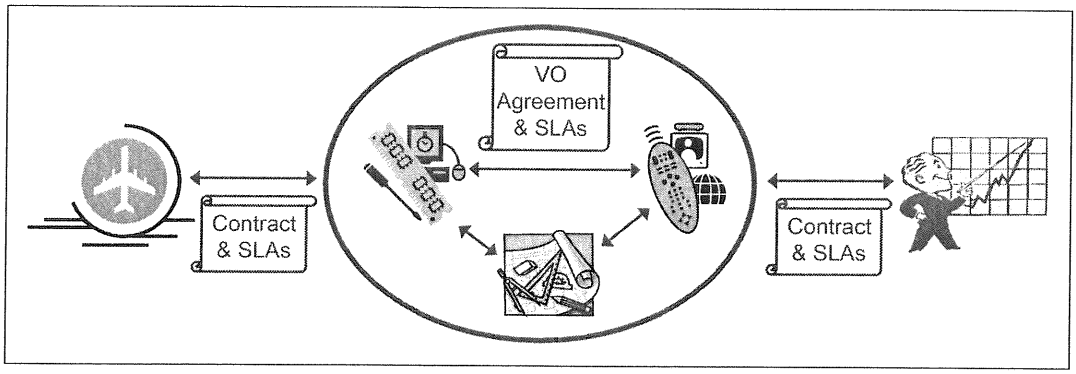
Based on SI's concerns, the focus of the risk analysis is defined as confidentiality of designs and customer requirements in relation to interaction between the partners of the CE VO and other external partners. The product design database (PDD) is central to the exchange of designs between the different CE VO partners and is regarded as a main focus point for the analysis. The target of the analysis is highlighted in the "rich picture" provided by the client of the VO and its partners, as shown in Figure 5.

The documentation provided by the client also contains descriptions of the main business processes related to the aircraft design process. A few of these are selected for analysis, based on where exchange of confidential information between the different participants occurs. The processes are modelled using UML activity diagrams, such as the high-level design process shown in Figure 6.

The legal expert and risk analysts also perform an analysis of the potential contractual obligations and rights of the VO and VO partners. It can be assumed that a number of different contracts will govern the internal and external relations and activities of the CE VO. These will most probably include at least three types of contracts:

- Consortium agreements, which establish a consortium of organisations with a common goal.
- Services or goods related contracts, e.g. outsourcing contracts, which govern the

Figure 7. Contracts in CE VO scenario



provision of services or the purchase of goods without establishing a consortium.
- Service level agreements (SLAs), that is, (mostly electronic) contracts that deal with the specific rules that partners in an operational business process are bound to.

An overview of these contracts and agreements are shown in Figure 7.

Identify Assets

Assets are the parts or features of the target of analysis that have value to the client and that the client wants to protect, such as physical objects, key personnel, services, software and hardware, or more intangible things such as know-how, trust, market share, and public image. By directing the analysis towards the assets of highest value to the client, one ensures that the available resources are spent on identifying the risks of highest impact on these assets. If the system or organisation does not contain any assets of value to the client commissioning the analysis, there is nothing that can be harmed and lose value for the client, and hence no point in a risk analysis.

The risk analysts typically perform an initial identification of assets based on the information provided by the client in presentations and target documentation. During the meeting, the list of assets is discussed and updated together with the client. To limit the size of the analysis, the number of assets should not grow too large; typically the four to six most important assets suffice.

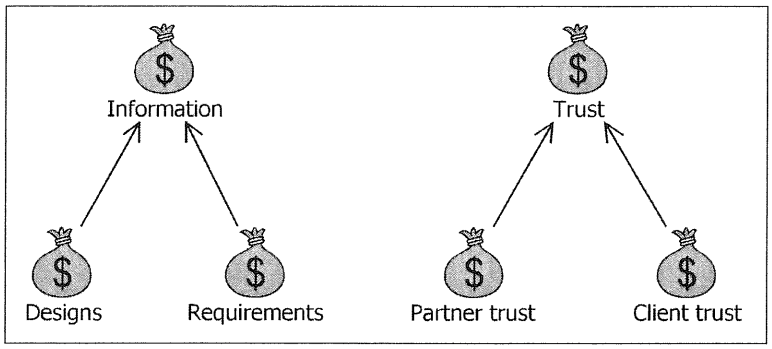
As mentioned in the target characterisation, the integrated aircraft designs and customer requirements were identified as the most important IP from the viewpoint of the system integrator. In addition, based on the discussion, it becomes clear that the system integrator is also concerned about its public image and how trust may be affected, both the trust of the other VO partners and the trust of customers of the system integrator. The identified assets are shown in the asset diagram in Figure 8.

High-Level Risk Analysis

Sometimes it may be difficult to determine exactly what should and should not be included in the risk analysis. For instance, identifying the most important assets may be hard without also looking at the relevant risks at the same time. Furthermore, the client is often tempted to include as much as possible. However, the result of this may be an inability to analyse anything at all in sufficient detail due to lack of time and resources for the analysis.

A preliminary high-level analysis of the target may be performed to identify the most important assets, threats, vulnerabilities and unwanted incidents to ensure that the focus of the analysis

Figure 8. Asset diagram



will be on the risks that the client is most worried about. The results of this analysis may help refine the focus and scope of the analysis and also serve as a starting point for the risk identification activity, where the results may be further refined and expanded upon.

This high-level analysis can thus be seen as a first iteration of the risk analysis. The same techniques for risk identification as described in the following sections may be utilised, but more informally. For example, one could use structured brainstorming as described in the section on risk identification below to identify focus areas but leave out the more detailed analysis of likelihood and consequence of the identified risks. The results of the high-level analysis are documented in a table such as the one below.

APPROVAL MEETING

The goal of the approval meeting is to ensure that the background documentation for the rest of the analysis, including the target, focus and scope as characterised by the risk analysts, is correct and complete as seen by the client of the analysis. The documentation of the target of analysis, assets and risk evaluation criteria must be approved by the client.

The client does not have unlimited resources to implement risk reducing measures. We therefore need a mechanism to prioritise the risks and select risks for further attention and treatment. To facilitate this, we must identify the level of risk that the client is willing to tolerate, in terms of loss of asset value over a given time interval. In order to assess the potential loss, we also need to determine the value of the assets.

This meeting should also include people who will be involved in the following risk meetings, such as domain experts, users, and so on, in order to give them an introduction to the analysis.

In preparation for the approval meeting, the risk analysts need to clean up the documentation of the target of analysis and assets. CORAS diagrams should be created for the results of the

Table 3. High-level analysis table

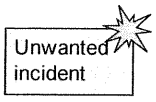
 Threat (deliberate)  Threat (accidental)  Threat (non-human)	   Asset	 Vulnerability
Who/what causes it?	How? What is the incident? What does it harm?	What makes this possible?
...	...	...

Table 4. CE VO analysis asset table

Asset ID	Description	Asset category	Asset value
Designs	SI's share in the designs of the passenger aircraft	Information	Very high
Requirements	The requirements of the VO's customer	Information	High
Partner trust	The VO partner's trust in SI	Other	High
Client trust	The client/customer's trust in SI	Other	Very high

high-level analysis. The resulting documentation should be sent to the client for perusal prior to the meeting.

Documentation of Target of Analysis

The documentation of the target of analysis, i.e. the system or organisation being analysed and the focus and scope of the analysis, forms the basis for the rest of the analysis activities. It is therefore essential that it correctly describes the target of analysis and captures the aspects that the client is most concerned about. A walkthrough is conducted of the documentation, and any errors or omissions are pointed out and recorded. Changes may be performed on the fly or by the risk analysts later on.

Asset Values

After identification, the assets should be ranked according to value or importance to the client, in order to facilitate selection of the most important assets and also prioritising the risks later on. Not all assets can be measured in monetary value, such as human life and health. In these cases, other criteria for risk evaluation may be needed. The assets should be documented in an asset table, as shown in Table 4.

Risk Evaluation Criteria

The goal of this activity is to determine what level of risk the client is willing to accept, in terms of what losses can be tolerated over a given period of time. Risk level is expressed in terms of likelihood, that is, what are the chances of this risk occurring, and consequence, what is the loss with regards to the asset which is affected by the risk. The likelihood and consequence values can be expressed in terms of quantitative values, such as statistical probability or amount of money lost. However, often we do not have the necessary data needed to calculate accurate values. Instead, we may use qualitative values for likelihood and consequence (e.g., low, medium, high), together with examples illustrating what these values mean. The values used for likelihood and consequence can be documented in a value definition table, such as the one shown in Table 5.

The risk evaluation criteria specify what level of risk the client is willing to accept, and should be expressed in terms of the likelihood and consequence values defined above. Based on the

Table 5. Value definition table from CE VO analysis

Value type	Values	
Likelihood	Rare:	Less than once per ten years.
	Unlikely:	Less than once a year.
	Possible:	About once a year.
	Likely:	2-5 times a year.
	Certain:	More than 5 times a year.
Consequence	Insignificant:	No impact on business. Minor delays.
	Minor:	Loss of profits. Lost project phases.
	Moderate:	Loss of project/client.
	Major:	Loss of business sector. Close department.
	Catastrophic:	Out of business

consequence and likelihood, a risk may either be accepted, or selected for further evaluation and treatment. Typically, this is done by setting up a matrix which shows the mapping of consequence and likelihood values to either "accept" or "evaluate," as shown in Table 6. Note that not all risks that end up in the "evaluate" region will necessarily be treated, depending on the availability and cost of effective treatments. Likewise, risks which end up in the "accept" region may still be treated if simple and inexpensive treatments are available.

RISK IDENTIFICATION

This meeting seeks to identify the risks to be managed, i.e. where, when, why and how incidents could prevent the achievement of objectives or reduce the value of an asset. The activity makes use of selected techniques and elements of conventional risk analysis methods which have been adjusted to fit the model-based approach of CORAS. The risk identification session is organised as a structured brainstorming, inspired by HazOp–Hazard and Operability Analysis (Redmill, Chudleigh, & Catmur, 1999).

The goal is to involve people with different backgrounds and different insight into the problem at hand in order to elicit as much relevant information about potential risks as possible. In addition to the risk analysts and the client, the meetings should include people with an interest in and knowledge of the system or organisation under analysis, such as security experts, lawyers, users, system managers, and so on.

Based on the identified assets, models describing the target, and the threats and weaknesses identified by the high-level analysis, the risk analysts should prepare the session by first selecting suitable models as a basis for the analysis, such as use cases, network diagrams, and so on, that match the desired level of abstraction. These should be illustrated using e.g. UML class, sequence or activity diagrams. The risk analysis leader should also prepare for vulnerability identification by selecting suitable checklists. The background documentation, in the form of models, checklists, and so on, should be sent out to the whole risk analysis team prior to the meeting.

Structured Brainstorming

The risk identification activity is organised as a structured brainstorming. The risk analysis team tries to identify scenarios describing how threats exploit vulnerabilities, leading to unwanted incidents which may reduce the value of one or more assets. The risk analysis leader uses the assets of highest value in conjunction with the diagrams of the target to guide the identification process, e.g. by asking relevant questions to the risk analysis team. The use of graphical diagrams also facilitates understanding and communication between the participants. The identification of threats and vulnerabilities may be supported with the use of

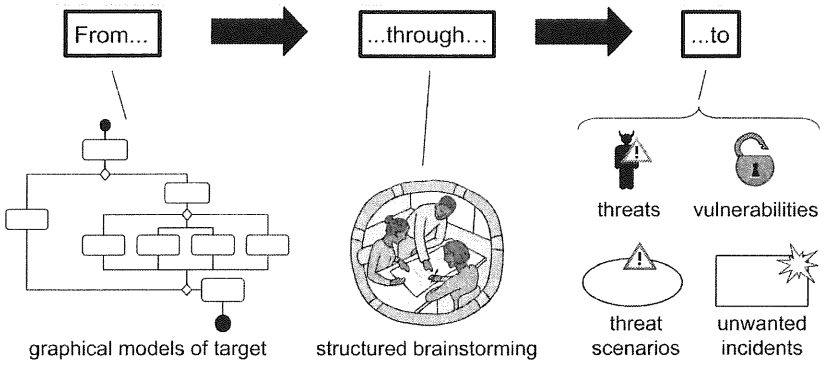
Table 6. Risk matrix from CE VO analysis

		Consequence				
		Insignificant	Minor	Moderate	Major	Catastrophic
Likelihood	Rare					
	Unlikely					
	Possible					
	Likely					
	Certain					

Accept risk

Evaluate risk

Figure 9. Model-based structured brainstorming



pre-defined questionnaires and checklists. This process is illustrated in Figure 9.

Vulnerabilities can be thought of as control mechanisms that ideally should be in place, but for some reason are missing or not sufficiently robust. Using this metaphor, vulnerabilities can be regarded as unsatisfactory controls, or exceptional circumstances that have not been planned for or that nullify the effect of existing, satisfactory, controls. Vulnerabilities can also be system characteristics that are impossible to treat; an internet connection that is crucial to the system, for example. Identifying new vulnerabilities is often a matter of finding the "blind spot". It is usually necessary to consider all aspects of the target (e.g., the organisational, judicial, physical, and computational characteristics) and compare these findings with the relevant policies.

During the meeting, one person from the risk analysis team should have the responsibility to record and document the results of the structured brainstorming. Following the risk identification meeting, the risk analysts structure the results and document the findings in diagrams using the CORAS graphical language for threat modelling. These diagrams are used later on as a basis for estimating the risk level as well as for identification of treatments. In the CORAS language, a threat (e.g., a disloyal employee or a computer virus) is related to a threat scenario, which is a sequence of events or activities leading to an unwanted incident. A vulnerability may be attached to this relation. An unwanted incident is an event resulting in a reduction in the value of the target asset. Furthermore, an unwanted incident may initiate or lead to other unwanted incidents, forming chains of events.

The risk analysts should also assess the need for further threat or vulnerability identification. For each unwanted incident the risk analysts should decide whether it is described at an appropriate level of abstraction, or whether additional analysis is required. The reason for the latter could be the need for more detailed incidents to make the assignment of frequencies feasible, or that the unwanted incident seems to require a higher priority than originally assigned. Additional information may be elicited from the client or other participants of the risk identification session, or the risk analysis leader may determine that an additional risk identification meeting is needed, but this time focusing on a smaller part of the target of analysis.

As a basis for the analysis, a number of models of the business processes in the organisation were selected. Figure 6 shows a high-level view of the iterative design process used by the CE VO. This process includes a lot of collaboration between the different VO partners, as well as interaction with the airliner at a number of points, such as in the concept and requirements phases.

The identified risks relate to different IPR issues, including the protection of confidential information (i.e., know-how and trade secrets), the ownership of IP, and liability for IPR infringements by other VO partners. The internal collaboration in the CE VO and its cooperation with the analysis company and the airliner, respectively, implies that confidential information is shared or otherwise disclosed to VO partners or to external parties. This involves the risk that such confidential information is disclosed to third parties or used by VO members for purposes that are not related to the VO.

Figures 10, 11 and 12 show use of the CORAS graphical language for describing some ways in which confidential information can be disclosed along with potential consequences this disclosure may have. For example, an employee may have access to confidential information which he/she could disclose to a third party, either willingly or by mistake. This disclosure could lead to the information being used for competitive purposes, or it could reach the public domain and thereby lose its legal protection and value as a trade secret.

RISK ESTIMATION AND EVALUATION

As mentioned in the approval meeting section, the client does not have unlimited resources to implement risk reducing measures. We therefore need to prioritise the risks and select a subset of them for further attention and treatment. Risk estimation is the systematic use of available information to determine how often specified events may occur and the magnitude of their consequences. A *risk* is an unwanted incident along with its estimated likelihood and consequence values. These values are the basis for risk evaluation. The goal of the risk evaluation is to prioritise the risks and identify which ones are in need of treatment by

Figure 10. Hacker steals designs and sells them to competitor

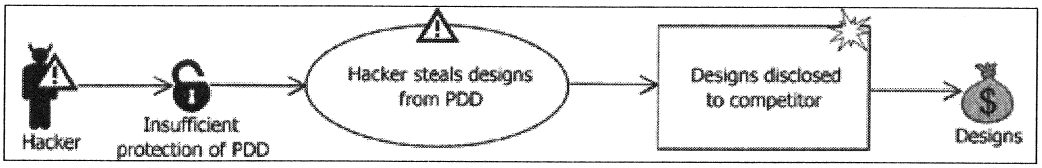


Figure 11. Unfaithful employee discloses customer requirements

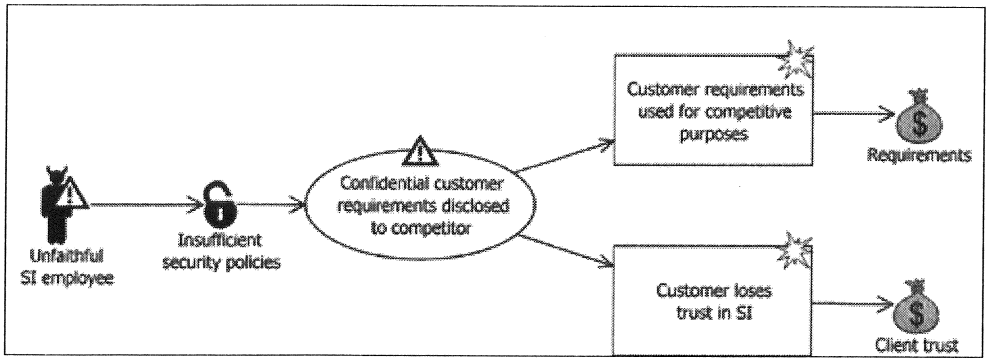
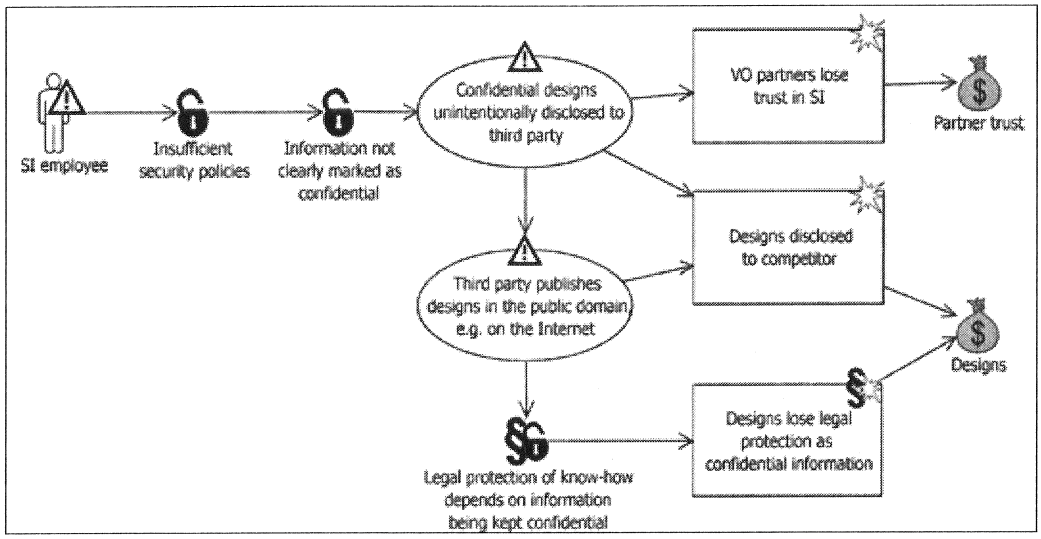


Figure 12. Loss of legal protection for know-how



comparing against the preestablished risk evaluation criteria.

Estimate Risk Level

The goal of this activity is to estimate the level of risk for the identified unwanted incidents. This consists of evaluating the likelihood and consequence of the incident. The consequence is a measure of loss of asset value when the incident occurs, while the likelihood is a measure of how often an unwanted incident occurs. The diagrams output by the risk identification activity are used as a basis for the likelihood and consequence evaluation. These document the identified threat scenarios, and may also contain consequence values which have been provided by the risk analysis team during the risk identification.

The methods chosen for consequence and likelihood evaluation depend on the results from the risk identification, the historical and statistical information available, and the analysis group's ability to assign consequence and likelihood values. In many cases, estimates are elicited from the client, domain experts or other people with knowledge of the target of analysis. If statistical

Table 7. Consequence and likelihood table

Risk	Asset	Unwanted incident	Consequence	Likelihood
R1	Designs	Designs disclosed to competitor	Moderate	Unlikely
R2	Requirements	Customer requirements used for competitive purposes	Moderate	Unlikely
R3	Client trust	Customer loses trust in SI	Major	Unlikely
R4	Partner trust	VO partners lose trust in SI	Major	Possible
R5	Designs	Designs lose legal protection as confidential information	Moderate	Possible

or historical data is available, more sophisticated methods may be used, for instance Fault Tree Analysis (IEC 61025, 1990) for calculating the frequency of an incident.

The risk analysis leader presents the CORAS diagrams. For each diagram, consequence and likelihood values are estimated for the different threat scenarios and unwanted incidents, based on expert judgements made by the system owner in collaboration with the risk analysis team. Some of the risks identified in the CE VO analysis are listed in Table 7 along with their consequence and likelihood values.

An example of how calculation of the likelihood of risk R1 could be performed using fault tree analysis is shown in Figure 13. For each event, a probability is given for it occurring during a time period of one year. The resulting probability of 0.28 fits the likelihood category 'unlikely' ("less than once a year").

Fault trees may also be used as a mechanism to decompose and structure scenarios and events without necessarily needing to perform the probability calculations.

Evaluate Risks

The risk evaluation compares the estimated risk level against the pre-established criteria which were identified in the approval meeting. This enables a prioritisation of risks, which is the basis for the subsequent decision about which risks should be targeted for treatment. Note that we may not be in a position to treat all risks, depending on the resources available for establishing risk reducing measures.

Prior to the evaluation, risks may be grouped or categorised. This categorisation can be done according to different concerns, for instance grouping risks which affect the same assets or which stem from the same vulnerability. This may reduce the work necessary for treatment identification and evaluation as the different risks in a category can often be treated using the same approach. An example based on the CE VO risk analysis is shown in Figure 14.

We then apply the risk evaluation criteria specified earlier during the approval meeting.

Figure 13. Fault tree

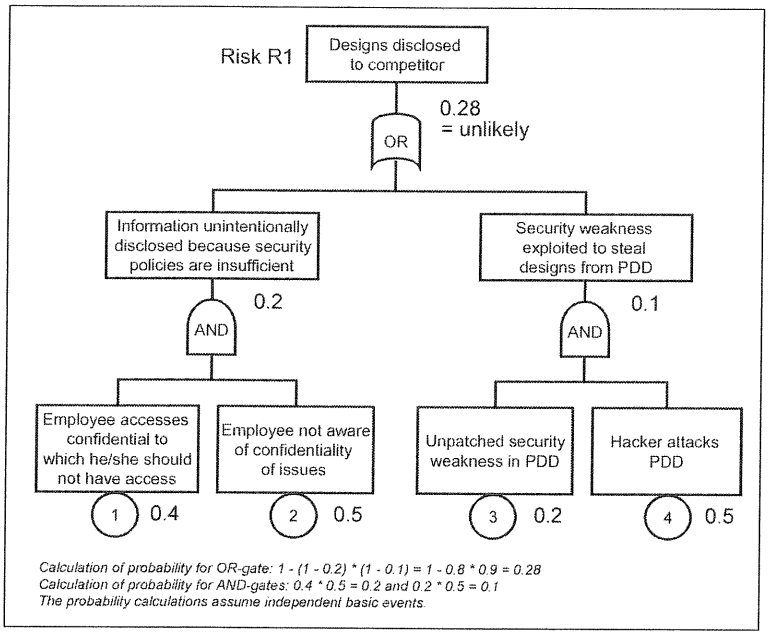
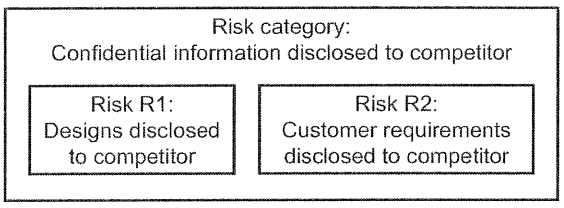


Figure 14. Risk category



After estimating the likelihood and consequence of the risks, they are plotted into the preestablished risk matrix, as shown in Table 8. As can be seen, risks R3, R4 and R5 need further evaluation, whereas risks R1 and R2 are accepted and may only need to be monitored to see if their risk level changes in the future. In the evaluation of R3-R5 it was decided that they are all in need of treatment.

RISK TREATMENT

This phase aims at treating the nonacceptable risks by developing and implementing specific cost-effective strategies and action plans for reducing the risk level.

Identify Treatments

For each risk which is not accepted, potential treatment options are explored in a similar manner to the structured brainstorming used for risk identification. This session typically involves the same participants as the risk identification. A walkthrough is performed of the CORAS diagrams created from the risk identification sessions, and the participants are asked to come up with suggestions for different ways to reduce the risk.

There are four main approaches to risk treatment:

- Reduce the likelihood of the incident occurring.
- Reduce the consequence if the incident should occur.
- Transfer the risk to another party (e.g., through insurance or outsourcing).
- Avoid the activity leading to the risk.

The outcome of the treatment identification is documented using the CORAS graphical language by adding treatments to the existing diagrams.

For each of the risks which were not accepted during risk evaluation, potential treatments are explored by the risk analysts and the other participants. A selection of treatments to the risks described above is shown in the CORAS diagrams in Figure 15 and Figure 16. The figures show some threat scenarios from Figure 11 and Figure 12 and some options for treating them.

The aim in the CE VO analysis was to develop an integrated set of treatments, where legal and

Table 8. Risks plotted into risk matrix

		Consequence				
		Insignificant	Minor	Moderate	Major	Catastrophic
Likelihood	Rare					
	Unlikely			R1, R2	R3	
	Possible			R5	R4	
	Likely					
	Certain					

Accept risk

Evaluate risk

other measures are seen together. In this context, the focus was on proactive legal mechanisms, which try to solve legal issues before they arise. Various access rights policies can be imposed via contractual clauses in the agreement between the CE VO partners as well as with the analysis provider (e.g., requiring that access is limited to only those people involved in the project), as well as requiring that access to the confidential information is monitored to allow for auditing. This is shown as two treatments in the figures below, which reduce the likelihood that some of the vulnerabilities from Figure 11 and Figure 12 will be exploited.

Furthermore, if the technology is available, a VO-internal enterprise digital rights management (DRM) system could also reduce the risk of confidential information being disclosed, particularly if some of the contractual obligations could be enforced through technological means. Information security mechanisms like limitations to storage time and the deletion of data after use were also identified as possible treatments.

Evaluate Treatments

To determine the best expenditure of the resources available for risk reducing measures, the identi-

Figure 15. Treatments for risk R3

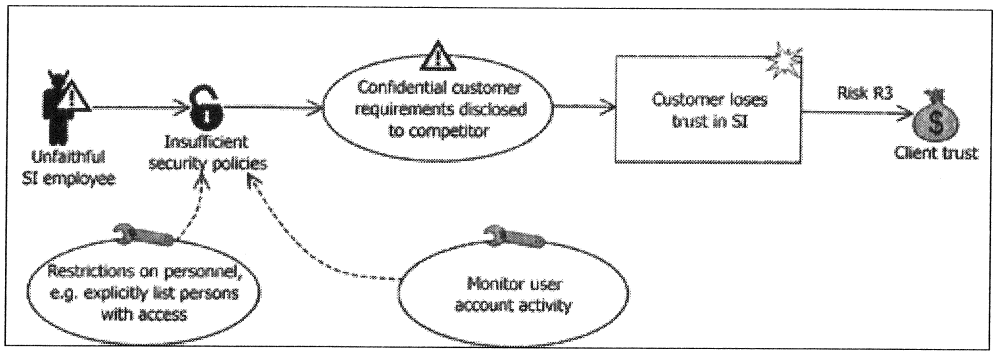
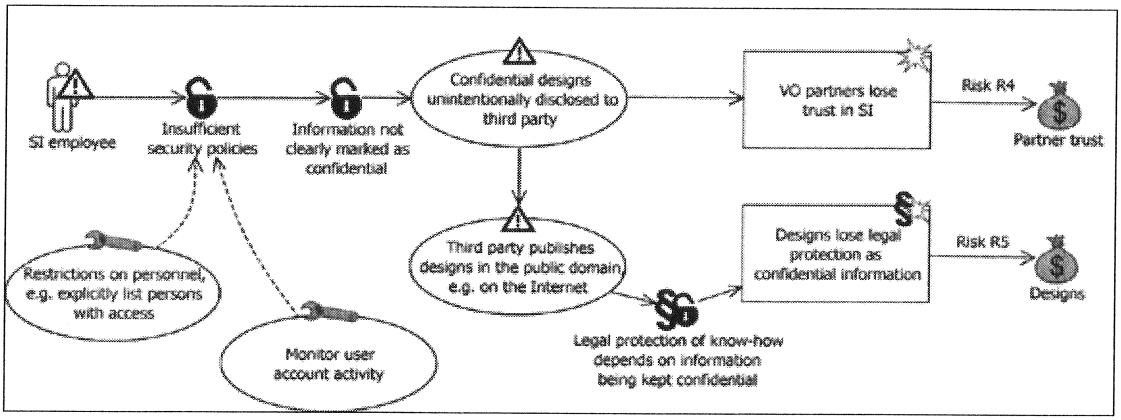


Figure 16. Treatments for risks R4 and R5



fied treatments are evaluated with respect to their usefulness. The degree to which the treatment reduces the level of risk is estimated, and a cost/benefit analysis is performed. Table 9 shows some examples of treatment evaluations from the CE VO analysis. Based on these results, the treatments can then be prioritised and implemented based on the available resources.

FINALISATION MEETING

For the risk analysis to have value, the findings of the risk analysis also need to be communicated to the relevant stakeholders to raise awareness and to ensure that relevant measures are put in place to prevent harmful events from occurring. In addition, the results may provide important input to future analyses, serving as a starting point and avoiding the need to start analysing from scratch every time.

The content of this meeting, and whether it is held at all, depends on how the client wants the findings of the risk analysis to be presented. To cut down on costs, the client may forego a written report in favour of a slide presentation of the main findings. Other clients want a written report, or a combination of both.

CONCLUDING REMARKS

In this chapter we have presented the CORAS method for model based security risk analysis and the CORAS graphical language. The risk analysis process has been illustrated with results from the analysis of a collaborative engineering VO scenario, where a number of risks and treatments were identified. The focus of this scenario was an integrated analysis of security, trust, and legal issues. The risk analyses conducted in the TrustCoM project indicate how legal risks, such as the loss of protection of confidential information, can be treated by an integrated solution, including contractual elements, trust management and security management (Mahler, 2005; Vraalsen, 2006). Interestingly, many of the relevant contractual treatments are also included in a general manner in the ALIVE contract template for VOs (ALIVE, 2002a). The risk analyses provide indications about how these rules can be adapted to the specific target under analysis. Since the graphical representation implies a simplification, a lawyer would have to integrate analysis results into the contractual document in an appropriate way, taking into account the terminology and the system of the contractual template.

Table 9. CE VO treatment evaluation

Risk	Unwanted incident	Asset	Treatment	Risk reduction	Cost
R3	Customer loses trust in SI	Client trust	Monitor user account activity	Major → Moderate	Low
R3	Customer loses trust in SI	Client trust	Access restrictions	Major → Moderate	High
R4	VO partners lose trust in SI	Partner trust	Monitor user account activity	Major → Moderate	Low
R4	VO partners lose trust in SI	Partner trust	Access restrictions	Major → Moderate	Medium
R5	Designs lose legal protection as confidential information	Designs	Monitor user account activity	No	N/A
R5	Designs lose legal protection as confidential information	Designs	Access restrictions	No	N/A

The analysis results presented in this chapter were generated during a number of brainstorming sessions involving participants with varied backgrounds, including law, computer science, engineering, economics, and formal methods and languages. Based on our experiences, the graphical models can indeed facilitate the communication and understanding with respect to security and legal issues in a multidisciplinary context, and this is also supported by other studies (Hogganvik & Stølen, 2005).

As a result of experiences and feedback from the risk analyses conducted in TrustCoM and other research projects, a number of improvements have been made both to the CORAS method and the graphical language (Vraalsen, 2006). Some of these improvements have been aimed at better support for legal risk analysis (Vraalsen, Lund, et al., 2005). Facilities have been added to enable modelling of legal risks and treatments, and reusable elements have been created in the form of e.g. checklists for legal risks. A number of general improvements have also been made. For instance, users were confused by the different types of lines and arrows in the diagrams, and these have now been cleaned up.

Current work focuses on updating the CORAS Tool with support for the new method and graphical language features. The built-in diagram editor has been extended with the new language facilities to support modelling of legal risks and treatments. Work is also being done on improving the reporting facilities in the tool and on updating the integrated online method handbook and tutorials.

ACKNOWLEDGMENT

The developments presented in this chapter were partly funded by the European Commission through the IST programme under Framework 6 grant 001945 to the TrustCoM Integrated Project and partly financed under the Research Council of Norway through the projects SECURIS (152839/220) and ENFORCE (164382/V30).

We would like to acknowledge the work done by David Goldby from BAE Systems, who has defined the collaborative engineering scenario for TrustCoM. We would also like to thank David Goldby, Xavier Parent and Claudia Keser for participating in the risk analysis sessions.

REFERENCES

AS/NZS 4360. (2004). *Risk management.* Standards Australia/Standards New Zealand.

ALIVE IST Project. (2002a). *Intellectual and industrial property rights (Tech. Rep. D13).* Retrieved June 5th, 2006, from http://www.vive-ig.net/projects/alive/docs.html

ALIVE IST Project. (2002b). *VE model contracts (Tech. Rep. D17a).* Retrieved June 5th, 2006, from http://www.vive-ig.net/projects/alive/models.html

Baccarini, D., & Archer, R. (2001). The risk ranking of projects: A methodology. *International Journal of Project Management, 19,* 139-145.

den Braber, F., Lund, M. S., Stølen, K., & Vraalsen, F. (2005). Integrating security in the development process with UML. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (pp. 1560-1566). USA: Information Resources Management Association.

Brændeland, G., & Stølen, K. (2004). Using risk analysis to assess user trust – A net-bank scenario. In C. Jensen, S. Poslad, & T. Dimitrakos (Eds.), *Proceedings of the Second International Conference on Trust Management (iTrust 2004)* (pp. 146-160). Oxford, England: Springer LNCS 2995.

Burnett, R. (2005). Legal risk management for the IT industry. *Computer Law & Security Report, 21,* 621-67.

Camarinha-Matos, L., Afsarmanesh, H., Löh, H., Sturm, F., & Ollus, M. (2004). A strategic roadmap for advanced virtual organizations. In L. Camarinha-Matos & H. Afsarmanesh (Eds.), *Collaborative networked oganizations: A research agenda for emerging business models.* New York: Springer.

Checkland, P., & Scholes, J. (1990). *Soft systems methodology in action.* New York: Wiley.

Dimitrakos, T., Goldby, D., & Kearney, P. (2004). Towards a trust and contract management framework for dynamic virtual organizations. In *E-Adoption and the knowledge economy: eChallenges 2004.* Vienna, Austria: IOS Press.

Heymann, T. (2005). Outsourcing in Deutschland – eine Bestandsaufnahme zur Vertragsgestaltung. Die Grundtypen des Outourcing und ihre Konsequenzen für die Vertragsgestaltung. *Computer und Recht, 10,* 706-710.

Hogganvik, I., & Stølen, K. (2005). On the comprehension of security risk scenarios. In *Proceedings of the 13th International Workshop on Program Comprehension (IWPC 2005),* 115-124.

IEC 61025. (1990). *Fault tree analysis (FTA).* International Electrotechnical Commission.

ISO/IEC 13335. (2004). *Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management.* International Organization for Standardization/ International Electrotechnical Commission.

Lund, M. S., Hogganvik, I., Seehusen, F., & Stølen, K. (2003). *UML profile for security assessment* (Tech. Rep. STF40 A03066). Oslo, Norway: SINTEF Telecom and Informatics.

Mahler, T. (Ed.). (2005). *Report on legal issues* (Tech. Rep. D15). Oslo, Norway: TrustCoM EU IST Project 01945.

Mahler, T., & Vraalsen, F. (2005). Legal risk analysis with respect to IPR in a collaborative engineering virtual organization. In *Proceedings of the 6th IFIP Working Conference on Virtual Enterprises (PRO-VE 2005).* Valencia, Spain.

Müller-Hengstenberg, C. D. (2005). Der Vertrag als Mittel des Risikomanagements. Ein Plädoyer für die dynamische Projektbegleitung im Vertrag. *Computer und Recht, 5,* 385-392.

OMG. (2006). *UML profile for modeling quality of service and fault tolerance characteristics and mechanisms, available specification* (OMG Document: ptc/2005-05-02) Author.

OMG. (2005). *Unified modeling language: Superstructure, version 2.0* (OMG Document: formal/2005-07-04).

Raz, T., & Michael, E. (1999). Use and benefits of tools for project risk management. *International Journal of Project Management, 19,* 9-17.

Redmill, F., Chudleigh, M., & Catmur, J. (1999). *HazOp and software HazOp.* Wiley.

Vraalsen, F. (Ed.). (2006). *Methods and Tools for legal risk management* (Tech. Rep. D17). Oslo, Norway: TrustCoM EU IST Project 01945.

Vraalsen, F., den Braber, F., Lund, M. S., & Stølen, K. (2005). The CORAS tool for security risk analysis. In P. Herrmann, V. Issarny, & S. Shiu, (Eds.), *Proceedings of the 3rd International Conference on Trust Management (iTrust 2005)* (pp. 402-405). Paris: Springer LNCS 3477.

Vraalsen, F., Lund, M. S., Mahler, T., Parent, X., & Stølen, K. (2005). Specifying legal risk scenarios using the CORAS threat modelling language – experiences and the way forward. In P. Herrmann, V. Issarny, & S. Shiu (Eds.), *Proceedings of the 3rd International Conference on Trust Management (iTrust 2005)* (pp. 45-60). Paris: Springer LNCS 3477.

Appendix G

Investigating Preferences in Graphical Risk Modeling

SINTEF

SINTEF ICT

Address: P.O.Box 124, Blindern
0314 Oslo NORWAY
Location: Forskningsveien 1
0373 Oslo
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

Investigating Preferences in Graphical Risk Modeling

AUTHOR(S)

Ida Hogganvik and Ketil Stølen

CLIENT(S)

Norges Forskningsråd / The Research Council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
SINTEF A57	Unrestricted	NFR152839/220	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
Open		40332800	48/3
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.) Ketil Stølen	CHECKED BY (NAME, SIGN.) Iselin Engan
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.)	
	2007-01-11	Bjørn Skjellaug, Research Director	

ABSTRACT

In a security analysis it is often helpful to draw diagrams to illustrate threat and risk scenarios. To ensure the effectiveness of such diagrams, it is essential that they are easily understood by people without training and experience in modeling and security analysis. In this report we present an empirical investigation of the risk modeling preferences among professionals and students in software engineering. The objective of the investigation was to identify the preferred way of refining an existing diagrammatic security risk modeling language without making it more difficult to understand. Our empirical investigation showed that mechanisms like size- and color coding used for conveying particular information in graphical models are less preferred by the subjects compared to textual information labels. The size or color of an element does not in general have any unique interpretation in a diagram, while textual information is more specific and self-explaining. The conclusion is that the subjects tend to prefer the representations where they get the most information without requiring them to interpret any additional graphical means.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Risk modeling	Risikomodellering
GROUP 2	Empirical investigation	Empirisk undersøkelse
SELECTED BY AUTHOR	Security analysis	Sikkerhetsanalyse
	Graphical means	Grafiske virkemidler

TABLE OF CONTENTS

1 INTRODUCTION 1

2 BACKGROUND AND MOTIVATION 3

3 MEANS AND MECHANISMS TO EASE COMPREHENSION 5

3.1 REPRESENTING GRAPH NAVIGATION 5
3.1.1 Set-up for "representing graph navigation" 6
3.2 REPRESENTING VULNERABILITIES 7
3.2.1 Set-up for "representing vulnerabilities" 8
3.3 REPRESENTING RISKS 8
3.3.1 Set-up for "representing risks" 10

4 EXPERIMENT DESIGN 11

4.1 SUBJECTS 11
4.2 MATERIAL 11
4.3 HYPOTHESES 12
4.4 ANALYSIS METHOD 12

5 RESULTS 13

6 DISCUSSION OF MODELING PREFERENCES 15

6.1 REPRESENTING GRAPH NAVIGATION 15
6.2 REPRESENTING VULNERABILITIES 15
6.3 REPRESENTING RISKS 16

7 THREATS TO VALIDITY 17

8 CONCLUSIONS 18

8.1 CONCLUSIONS REGARDING "REPRESENTING GRAPH NAVIGATION" 18
8.2 CONCLUSIONS REGARDING "REPRESENTING VULNERABILITIES" 18
8.3 CONCLUSIONS REGARDING "REPRESENTING RISK" 19
8.4 HOW THE RESULTS IMPACTED THE CORAS LANGUAGE 19
ACKNOWLEDGEMENTS 20

REFERENCES 22

LIST OF TABLES 25

LIST OF FIGURES 25

APPENDIX A – RELATED MODELING NOTATIONS 26

APPENDIX B – EXPERIMENT MATERIAL 28

B.I MODELING QUESTIONNAIRE 28
B.II TASKS THAT WERE LEFT OUT OF THE ANALYSIS 34

APPENDIX C – PRESTUDY 36

C.I PRESTUDY DESIGN 36
C.II PRESTUDY MATERIAL 36
C.III PRESTUDY RESULTS AND DISCUSSION 40
C.IV THREATS TO VALIDITY FOR THE PRESTUDY 43
C.V CONCLUSION 43

1 Introduction

As security risk analysts we often face clients with problems like the following: "we've installed a new system, how does it affect our security?", "the system has existed for a long time and undergone several updates and changes, but how secure is it now?" or "we like to publish more sensitive information via our intra/extranet solution, but what are the risks?" To help responding to these questions we have developed a graphical modeling language supporting the identification, communication and documentation of security threats and risk scenarios called the CORAS security risk modeling language (the CORAS language)[1].

The CORAS language with its detailed user guidelines has been applied in several large industrial field trials [11], and the major decisions regarding its underlying foundation, notation and guidelines are supported by empirical investigations [17, 18]. In this report we present the results from a study of risk modeling preferences among professionals in software engineering. The objective of the study was to find ways to refine the CORAS language to enhance comprehension. From earlier studies of security analysis concepts and terminology we know that some concepts are more difficult to understand than others [18]. These studies lead to the hypothesis that the relations that were found easy to understand would also be most suitable and appropriate to represent in a simple and straight forward manner, while the difficult concepts would need more sophisticated representations. A concept for which we searched for a more sophisticated representation was *likelihood of threat scenario paths.* The second issue in the investigation on which this report presents, was how to represent *vulnerabilities* in a more explicit manner, something that was insufficient in the original version of the CORAS language [31-33]. The last issue was whether or not we could identify a way of representing risk, and especially the *severity of risks*, to quickly give the reader an overview of the overall risk picture. To solve these issues we tested various means for representation known from the field of information visualization [45], in addition to textual information labels.

The CORAS language is intended to be used during a security analysis by the analyst, and has different purposes in each phase of the analysis. A security analysis process typically includes five phases: (1) context establishment, (2) risk identification, (3) risk estimation, (4) risk evaluation and (5) treatment identification [2]. A frequently used technique in security analysis, particularly during risk identification, is structured brainstorming. It may be understood as a structured "walk-through" of the target of analysis. The technique is not limited to a specific type of target, but can be used to assess risks towards anything from simple IT systems to large computerized process control systems or manual maintenance procedures e.g. in nuclear power plants. This does not mean that exact same technique can be applied directly to all kinds of targets; it has to be adapted to fit the target domain. The main idea of a structured brainstorming in security analysis is that a group of people with different competences and backgrounds will view the target from different perspectives and therefore identify a larger number, and possibly other, risks than individuals or a more heterogeneous group. Gathering people with a broad knowledge and experience with vulnerabilities, threats, bugs and flaws is important. The input to a brainstorming session is various kinds of target models (e.g. UML models, informal descriptions, pictures etc.). The models are assessed in a stepwise and structured manner under the guidance of the security analysis leader. The identified risks are documented by the analysis secretary. Deciding which roles that should be represented at the brainstorming session is part of tailoring the brainstorming technique towards the target in question. In some cases an expert is only participating in the part of the analysis where his or her expertise is needed. This makes it essential that we have a comprehensive and simple manner for communicating and documenting risk specific information.

[1] http://coras.sourceforge.net

Documenting security risks graphically has many similarities with information modeling. Information modeling is used to describe the behavior of a workflow or process and risk modeling deals with describing the "workflow" or "process" of a threat that initiates an unwanted incident. During risk analyses we have experienced the need for describing *how* incidents can happen, *who* initiates them and *what* they affect. We also need to show the relationships between assets, stakeholders, treatments, risks and other elements. Our work has been to develop a modeling language that is both sufficiently expressive to satisfy the modeler's needs, yet simple and intuitive.

The report is structured as follows: in Sect. 2 we give the background and motivation for our investigations. In Sect. 3 we introduce various techniques for information visualization that can be used in modeling, and describe the alternatives we have explored. Sect. 4 presents the experiment design and the results are given in Sect.5. Our findings are discussed in Sect. 6 and the identified threats to validity are reported in Sect.7. Sect. 8 describes how the findings impacted the CORAS language, and Sect. 9 brings the conclusions from our work. Appendix A gives an overview of related modeling approaches. Appendix B presents the experiment material, while Appendix C describes the prestudy that was conducted before the main experiment.

2 Background and motivation

The background for this investigation is partly based on previous findings regarding the conceptual foundation and also weaknesses of the previous version of the language [31-33]. In the following we describe this in more detail. Let us first explain the UML class diagram [34] in Figure 1 that presents the conceptual foundation. The model may be understood as a kind of abstract syntax for the CORAS language, and its definitions are taken from the following international standards:

- *Information technology: Guidelines for management of IT Security* (ISO/IEC13335) [21, 22]
- *Australian/New Zealand Standard for Risk Management* (AS/NZS4360) [2]
- *Information Security Risk Management Guidelines* (HB231) [15]

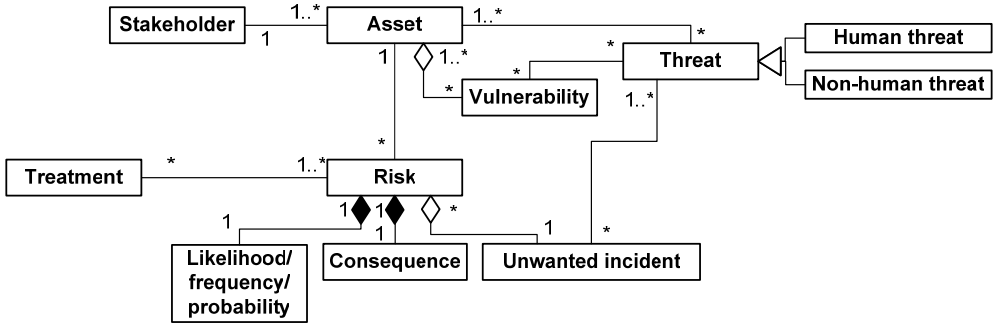


Figure 1 – The conceptual foundation

Figure 1 can be explained as follows: **stakeholders** are those people and organizations who may affect, be affected by, or perceive themselves to be affected by a decision or activity regarding the target of analysis [2]. An **asset** is something to which a stakeholder directly assigns value and, hence, for which the stakeholder requires protection [15]. Assets are subject to **vulnerabilities,** which are weaknesses which can be exploited by one or more threats [22]. A **threat** is a potential cause of an unwanted incident [22]. A threat may be classified as human (with accidental origin or deliberate harmful intentions) or non-human (also called environmental) [22]. An **unwanted incident** is an event that may harm or reduce the value of assets and is something we want to prevent [21]. In [22] they use the more specialized term "information security incident" (any unexpected or unwanted event that might cause a compromise of business activities or information security.) A **risk** is the chance of something happening that will have an impact upon objectives (assets) [2]. Our model captures this interpretation by defining a risk to consist of an unwanted incident, a likelihood measure and a consequence. The abstract concept "risk", the more concrete "unwanted incident", and their respective relationships to "asset" require some explanation. In our definition, an unwanted incident that harms more than one asset gives rise to one distinct risk for each asset it harms. This enables us to keep the consequences for different stakeholders separate, since an asset is always defined with respect to a single stakeholder. The level of risk is measured by a **risk value** [2] (e.g. low, medium, high or other scales) which is based upon the estimated **likelihood** (a general description of frequency or probability [2]) for the unwanted incident to happen and its **consequence** in terms of damage to an asset. A **treatment** is the selection and implementation of appropriate options for dealing with risk [2].

A previous experiment [18] focusing on the understanding of this conceptual foundation included 57 subjects, professionals within software engineering as well as students. They responded to a questionnaire with questions in six categories, each targeting a risk related concept: *assets*, *treatments*, *frequency measures (freq.)*, *risks*, *vulnerabilities (vuln.)*, *and threats*. It was found that questions related to vulnerability and asset received more correct answers than e.g. the frequency measure category (Figure 2). This finding motivated us to investigate:

1. How should we visualize frequency measures in threat diagrams to ease graph navigation, especially the likelihood of single paths and combined paths?

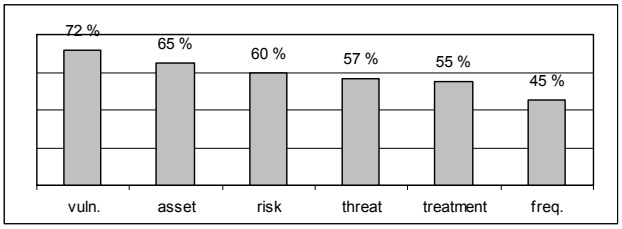


Figure 2 – Percentage of correct answers for each category (mean)

As mentioned, the first version of the CORAS language had some weaknesses, and among others it lacked a proper representation of vulnerabilities. Being able to specify vulnerabilities explicitly has been a requirement from the users of the CORAS language in field trials, but exactly how this should be done has not been clear. This made us investigate the following:

2. How should we visualize vulnerabilities? What is the preferred way among professionals within software engineering?

Both in empirical investigations [17, 18] and in field trials we have experienced that the concept "risk" belongs to our everyday vocabulary, but most people find it difficult to specify exactly what it is. The concept of risk is different from the other concepts we have studied in the sense that it is more abstract and is used to cover other, more concrete concepts. To make it less abstract we looked for the preferred way of representing it graphically. Hence, this motivated the investigation of:

3. How can we visualize risks in the models to inprove the understanding of this abstract concept?

In the following we deal with each of the three issues listed above in separate subsections. The first is referred to as "representing graph navigation" (Sect.3.1, Sect.6.1 and Sect.8.1), the second as "representing vulnerabilities" (Sect.3.2, Sect.6.2 and Sect. 8.2) and the third as "representing risk" (Sect.3.3, Sect.6.3 and Sect.8.3).

3 Means and mechanisms to ease comprehension

The CORAS language is not a programming language, i.e. it is not to be executed on a computer. Nevertheless, it has been carefully designed to precisely model detailed sequences of events, and at the same time serve as a tool for capturing information from the creative brainstorming session. There exist many means and mechanism for helping us present information in a better way. Software engineering studies have shown that applying graphical means to program- and component specifications increases the understanding of the system [4], and one has successfully applied graphical means to visualize the program interdependencies [29].

A presupposition for understanding a CORAS diagram is some familiarity with the domain it describes. According to Winn [48] people that are familiar with the domain seek, encode, process and recall information in a diagram more effectively than those that are less expert. In addition to domain understanding, some knowledge of the semantics and syntax of the language is also required. Our aim was to implement means that reduced the latter to a minimum by adding helpful information in terms of symbols, textual information labels or other graphical means. We were inspired by research within the fields of information visualization in computer displays and statistical graphs/figures, related modeling languages, diagrammatic reasoning and model quality.

In the following we motivate and present the means and mechanisms that were selected as subjects for investigation, structured according to the three main issues identified at the end of the previous section: *representing graph navigation, vulnerabilities* and *risk*. At the end of each subsection we present the diagram alternatives investigated in the experiment. We refer to the different tasks in the experiment using the following convention: "TE2" means Task 2 in the Experiment, "D1" means Diagram alternative 1, "TE2D1D2" means Diagram 1 compared to Diagram 2 in Experiment Task 2.

3.1 Representing graph navigation

As explained above, we refer to the process of understanding and reading CORAS diagrams as "graph navigation". When describing the different paths via which a threat may choose to harm assets we often find it useful to draw special attention to the paths that they are more likely to follow. It was also desired by the users of the CORAS language to have a way of describing the logical operators AND and OR when two paths are combined. An ambiguous threat scenario situation where logical operators could be useful is described in Figure 3.

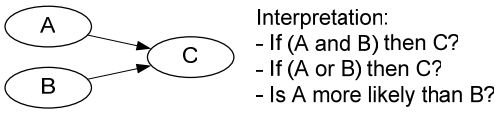


Figure 3 – Typical threat scenario situation

The first task was to find means that could help us express the likelihood of paths. The CORAS language is based on a node-link diagram type, which in its simplest form consists of nodes connected via edges. A closed contour in a node-link diagram generally represents a concept of some kind, and a linking line between concepts represents some kind of relationship between them. Manipulating the appearance of these lines may give them different meanings (Figure 4).

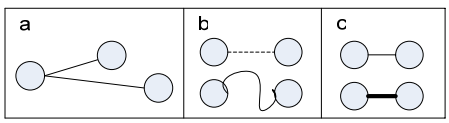


Figure 4 – Node-link configurations

Lines linking closed contours may have different colors, be straight or wavy, or have other graphical qualities that represent an attribute or type of relationship [45]. The thickness of a

connecting line may be used to represent the magnitude of a relationship (a scalar attribute). Contrasts in the size or color of a line may make it "pop-out" perceptually and thereby call for the reader's attention [44]. This may be a useful feature to include in the CORAS language. We chose to test thick, thin and dashed lines to illustrate more or less likely paths through the threat diagrams (Figure 6).The use of line variations was also inspired by network traffic maps which often use node-link diagrams to visualize network traffic. Typically one illustrates the load or type of link data using different line colors or varying the thickness of the lines [3]. Variation of line appearance is also inspired by the gestalt principle *similarity*, meaning something that is different from its surroundings, will be easier identified by the reader (e.g. *italic style* vs. normal style in text). The gestalt principles [46] are some of the most recognized set of rules for good graphical displays. Implementing the gestalt principles may reduce the effort needed to understand illustrations, program interfaces, websites etc.

When looking for an AND-operator symbol, it was natural to turn to one of the most frequently used technique in risk analysis: Fault Tree Analysis (FTA) [20]. Fault trees let you specify dependencies between events in a tree structure using logical AND/OR operators called "gates". One of our goals is to make the CORAS language fault tree compliant. In the CORAS language, there was no standard interpretation of two threat scenarios pointing to the same threat scenario or unwanted incident (Figure 3). In order to express fault tree-logic we needed a way of saying both "If A and B occurs, C must occur" and "If either A or B occurs, C must occur". The challenge was to find the preferred way of illustrating this. We decided to test the FTA-symbol for "AND" (Figure 5) which is also used in electrical circuit design [37]. We added an information label with the text "and" to the symbol to ease comprehension for people without background in FTA or circuit design. There exist a similar symbol for OR-gates, but we chose to test only one of them since they are very similar and our findings probably can cover both gate-symbols.

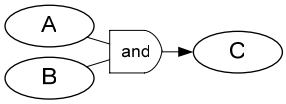


Figure 5 – The AND-gate symbol

In addition to the AND-gate symbol and dashed lines to show dependencies between combined paths, we tested a UML inspired "information note". In UML one may attach notes to a diagram element to convey additional information about the specific element. The idea was to use the notes as an alternative to logical gates.

3.1.1 Set-up for "representing graph navigation"

The effect of two different line styles was tested in TE2 for visualizing paths that are more likely to be chosen by a threat (Figure 6):

- D1: dashed lines that are "weaker" than solid lines to illustrate less likely paths.
- D2: thick lines that are easily noticed to highlight the more likely paths in the graph.

TE2: Which diagram is best at illustrating that it is more likely that threat K causes the unwanted incident T than L or F?

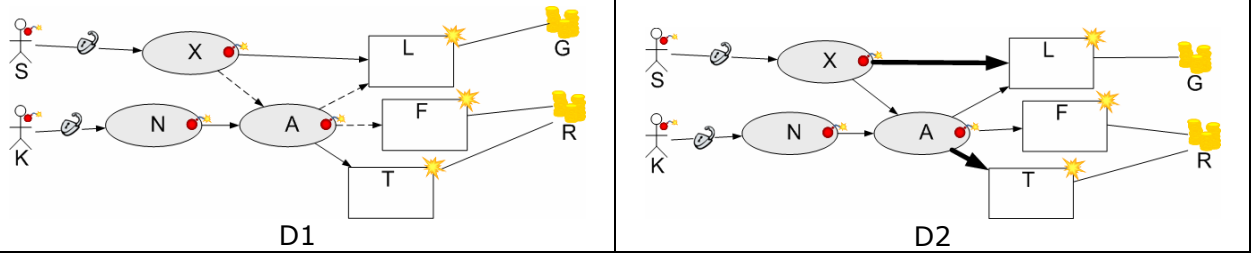


Figure 6 – The diagram alternatives for: "most likely path through a graph"

Three modeling alternatives for logical "AND" were tested in TE4 (Figure 7):

- D1: the special AND-gate symbol from fault tree notation and electrical circuit design, labeled with the text "and", was used.
- D2: we aimed to give the two paths a special meaning by "sublightning" them, changing their appearance from a solid line to a dashed line to indicate a "special AND-dependency".
- D3: we tested the effect of attaching UML-like notes to show the AND-dependency between the two paths.

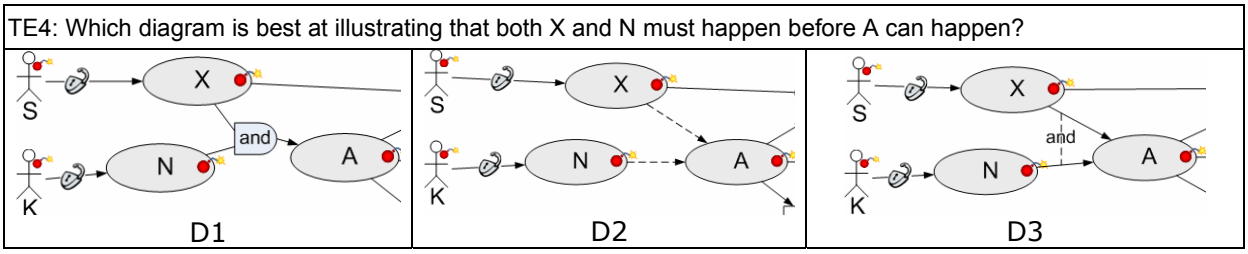


Figure 7 – The diagram alternatives for: "logical AND in graph paths"

3.2 Representing vulnerabilities

The original CORAS language could only illustrate vulnerabilities as properties of their respective assets. There was no way of showing exactly where in the chain of events a vulnerability was exploited, and there was no special vulnerability symbol. Both these aspects were requested by the users of the CORAS language.

The way vulnerabilities was modeled in the original CORAS language made it simple to see all the vulnerabilities of an asset at a glance, but involved the duplication of vulnerabilities that were shared between two or more assets. Vulnerability is a stand-alone concept, not a binary relation as the ones discussed in the previous section on graph navigation. The descriptive text accompanying the vulnerability is obligatory. The question was whether it should have a symbol in addition to the textual description. In research on the quality of modeling languages, pictures are claimed to have a much higher perceptibility, and information conveyed in pictures will be emphasized at the cost of textual information [28]. This is in accordance with the results from our experiment with graphical symbols where the textual stereotyping seemed to be overlooked [17]. In the same experiment we tested UML symbols vs. special symbols and found that the representation using special risk related symbols performed better than the one with conventional UML symbols. These finding suggests that carefully designed symbols are useful and that vulnerability should have a graphical symbol in addition to the text. We chose to use an *open chain lock symbol*, which is often used to symbolize a lack of security within the computer security domain. It is a simple and internationally known symbol. To achieve what [13] terms "syntactic disjointness" , the symbol is unique and it is neither too small, nor too large to cause misinterpretations (large elements often attracts attention at the sacrifice of the smaller elements). Syntactic disjointness makes it easier to separate the different elements in a model. We compared this representation to the original CORAS language representation with respect to "shared vulnerabilities".

One may argue that the way vulnerabilities was modeled in the original CORAS language was an example of the gestalt principle *proximity*, stating that things that logically belong together should be placed close to each other (commonly used in program interfaces and websites). This view is very suitable if the focus is on the assets and their vulnerabilities, but when it comes to addressing *which* threats exploiting *which* vulnerabilities and especially *where* in the chain of events the exploitation takes place, the situation is different. This kind of information is especially valuable in relation to treatment identification. As a consequence of this we decided to try modeling the vulnerabilities where they logically belong in the chain of events.

Winn [49, 50] has found that people, whose languages are written from right to left, also process graphics from right to left. A related finding is that the item to the left always was interpreted as the cause while the item to the right was the effect [51]. This is in accordance with the modeling style already used in the CORAS language. Threats always initiate the chain of events from the left and terminate at the right side of the diagram, where the assets are placed. This is also in accordance with one of the node-link rules [45] stating that "placing closed contours spatially in an ordered sequence can represent conceptual ordering of some kind" (Figure 4, alt. b), and vulnerabilities may very well be placed within this ordering. The direction of the line should also be indicated with arrows along the line, since this have been found helpful in understanding similar notations like flow charts (used in schematic representations of processes) [7]. If the number of lines increases one may easily loose track of the path and repeated arrows help distinguishing between crossovers and connection points.

3.2.1 Set-up for "representing vulnerabilities"

Two alternative representations of shared vulnerabilities (common for two or more assets) were tested in TE5 (Figure 8):

- D1: we represented vulnerabilities with an open chain lock symbol and a vulnerability description. The vulnerabilities were located where they logically belonged in the sequence of events.
- D2: we linked the vulnerabilities closely to the assets they belonged to by specifying them as textual attributes of each asset. Since a vulnerability may yield for more than one asset, it may be listed under several assets.

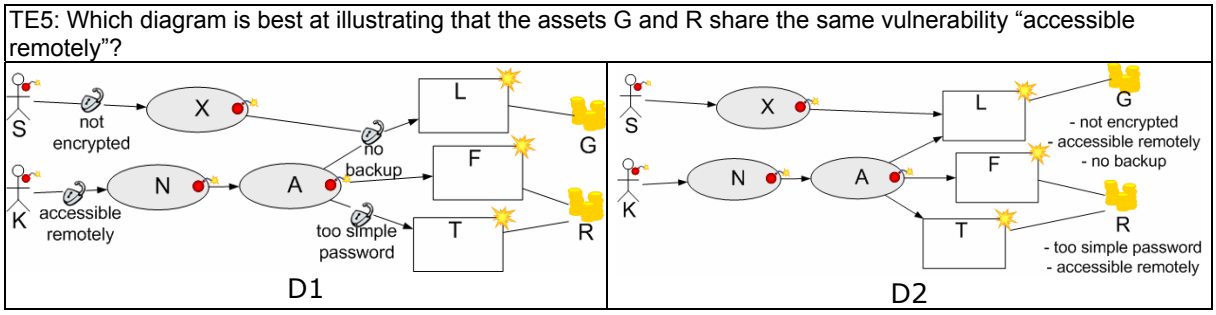


Figure 8 – The diagram alternatives for: "shared vulnerabilities"

The same alternatives were also tested in task 6 (TE6), but in the opposite order (TE6D1 = TE5D2 and TE6D2 = TE5D1).

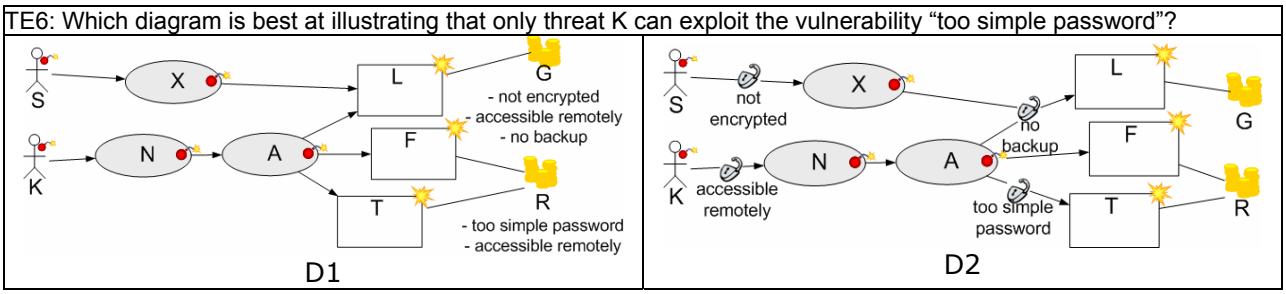


Figure 9 – The diagram alternatives for: "which vulnerabilities are exploited by which threats?"

3.3 Representing risks

Representing risk is a great challenge to its abstract nature. In the threat diagrams of the original CORAS language risks were not illustrated explicitly, but represented as relations between unwanted incidents and assets. Illustrating the severity or magnitude of risks and unwanted

incidents helps keeping focus on the most critical risks, something which is especially useful during the final risk treatment phase. To visualize this issue, we investigated means like line thickness, textual information labels, shapes that varied in size or color and more. The color of an enclosed region may represent a concept type (Figure 10) and the size may be used to represent the magnitude of a concept (Figure 11). This is also inspired by the gestalt principle of breaking the pattern to attract attention. We want the reader to quickly discover the most serious unwanted incidents, since they often represent major risks. We therefore tested the effect of marking the most serious unwanted incident with a darker color (D1, Figure 13), since dark colors are more easily noticed against white background than light colors.



Figure 10 – Color



Figure 11 – Size

There are some aspects to be aware of when using color coding. The number of different colors one may use is limited by the reader's ability to remember and distinguish the colors. Much research has been conducted to find the optimal number of colors, and the suggestions varies from 5 to 8 [5, 9, 10, 38, 45, 47]. In our case we only used gray and white. One should also have in mind how a model with colored symbols will look when printed in black-white and whether this may affect the interpretation of the symbols [13]. Different types of coding in statistical graphs have been investigated with respect to display search time. The coding that gave the best performance was color, the second best coding was shape and then came letters/digits [8]. This top ranking of color has been confirmed by many other studies [25]. According to these results we should benefit from using color to emphasize the most serious incidents, meaning that the reader should identify them more quickly compared to using other means. The color-alternative was compared to one using *size* to symbolize the most serious incident since large elements are more easily noticed than small ones (D2, Figure 13). As in the case of colors, the number of different size-categories is not indefinite. The number of different size steps that can be distinguished from each others at a glance is as low as four [45] (p. 182). An important aspect when using shapes is to avoid symbols that are of similar shapes. Shapes that are too similar have been found to increase search time and are therefore not recommended [42]. In our experiment material we only compared two sizes, whereas the number of size steps in a real diagram would most likely correspond to the three steps *low, medium* and *high risk* i.e. only one step more than in our material.

We also explored the use of textual information in the unwanted incident symbol (D3, Figure 13) and as an annotation to the risk association (D2, Figure 14). Our motivation for doing this was that studies of rapid processing of information [6, 35] have found letters and digits to be some of the best coding forms. These investigations involved studying short-time memory and how graphical means are memorized using Sternberg tasks[2]. Digits, colors and letters were found to be the top three coding forms for rapid processing of information. Another investigation found digits, words and letters to be superior to colors and shapes when it comes to processing speed [43], and digits, letters and words were found to represent less subjective workload than colors and shapes [43]. These findings support the modeling alternatives that use textual information labels to illustrate the seriousness of unwanted incidents and risks.

In [45] some general rules regarding the use of shapes are described. For instance that the shape of a closed contour can be used to represent a concept type (Figure 12). Risk was one of the few relevant concepts for which the original CORAS language did not have a special symbol. We therefore investigated representing risks with a symbol and using size to represent its seriousness

[2] For a limited time the subject is shown a set of objects which has to be memorized.

(D3, Figure 14). We selected a traditional, red and white road sign that symbolizes "danger" that varies in size according to the severity of the risk.

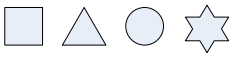


Figure 12 – Shapes

3.3.1 Set-up for "representing risks"

We investigated which of size, color or textual information is better suited to represent the magnitude of an unwanted incident in TE3 (Figure 13):

- D1: we gave the most serious unwanted incident a darker color than the other incident.
- D2: we represented the most serious unwanted incident with a larger symbol.
- D3: we used textual information labels to specify the severity of the unwanted incidents.

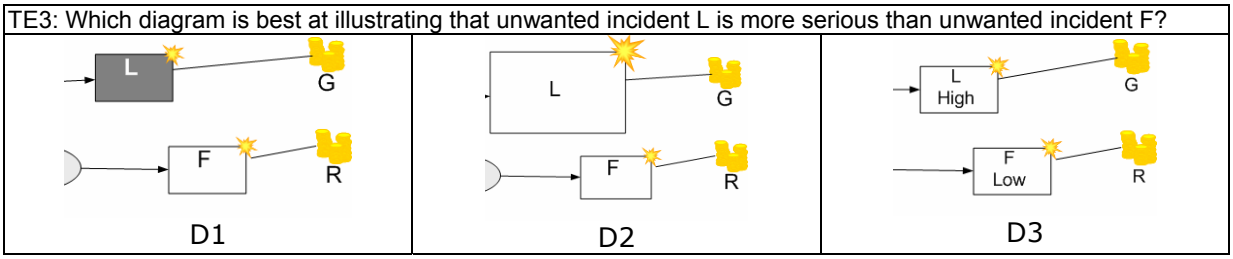


Figure 13 – The diagram alternatives for: "the magnitude of unwanted incidents"

In task 7 (TE7) we investigated the following modeling alternatives (Figure 14):

- D1: we used thick lines to illustrate the largest risks (discussed in Section 3.1).
- D2: we used textual information.
- D3: we tested a combination of symbol and size to illustrate both the concept of risk, and how to convey its seriousness.

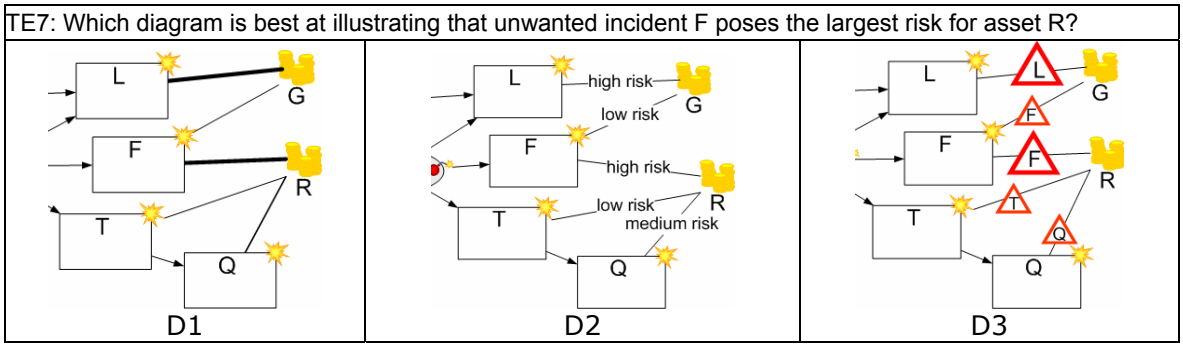


Figure 14 – The diagram alternatives for: "the magnitude of risks"

4 Experiment design

To decide upon the preferred notation we conducted an empirical study using professionals as subjects. Similar representations had been tested in a prestudy using students.

4.1 Subjects

The survey was distributed via e-mail to people in various parts of the software industry. The 33 subjects received no payment and participated voluntarily. The majority of the subject group was male between the age of 26 and 35 with 1-3 years of work experience. Work experience is shown in Figure 15. The term task experience includes the following task categories: *system design, development, testing, quality evaluation, maintenance, marketing/sale* and *security*. System security was added later in the experiment when we were interested in obtaining answers from a group of subjects that was less likely to have system development experience, but that were more experience with system security. The subjects' task experience is shown in Figure 16. The demographic data was not used for hypothesis testing, but rather to give a picture of the type of subjects that participated.

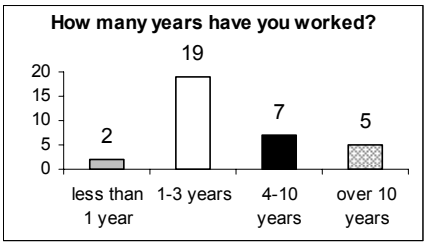


Figure 15 – The subjects' years of work experience

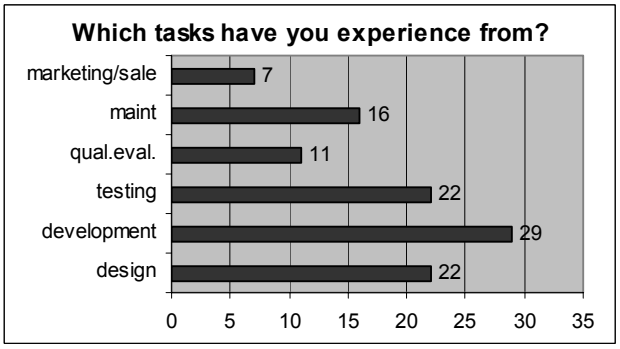


Figure 16 – The subjects' task experience

4.2 Material

The material was in the form of a questionnaire where the subjects were to prioritize between different modeling alternatives. It was similar to the prestudy using students, and a complete version is provided in Appendix B. It consisted of eight tasks and was estimated to take about 15-20 minutes to fill in. The tasks were related to variations of the basis threat diagram shown in Figure 17. The focus of each varied, but the main issues were how to simplify graph navigation, how to best represent vulnerabilities and how to highlight major risks and unwanted incidents. The explanation given in the material was as follows: *This questionnaire is about threat modeling. We present various ways of modeling the same situation, and your task is to prioritize between them and choose the diagram you think is best! This type of modeling is meant to support the risk analyst in situations where the participants in meetings have different backgrounds and expertise (technical and non-technical). The diagrams are meant to help them understand the system and each other better and faster. Compare two and two diagrams and choose the one you prefer by marking with an "X" in the table below the weight symbol, like shown in this example:*

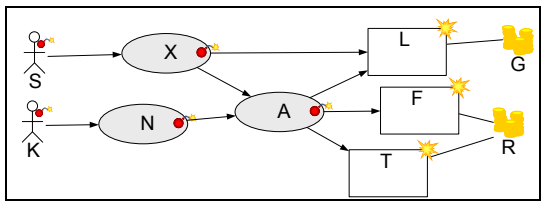


Figure 17 – The basic threat diagram

Which diagram do you prefer, diagram 1 or diagram 2?

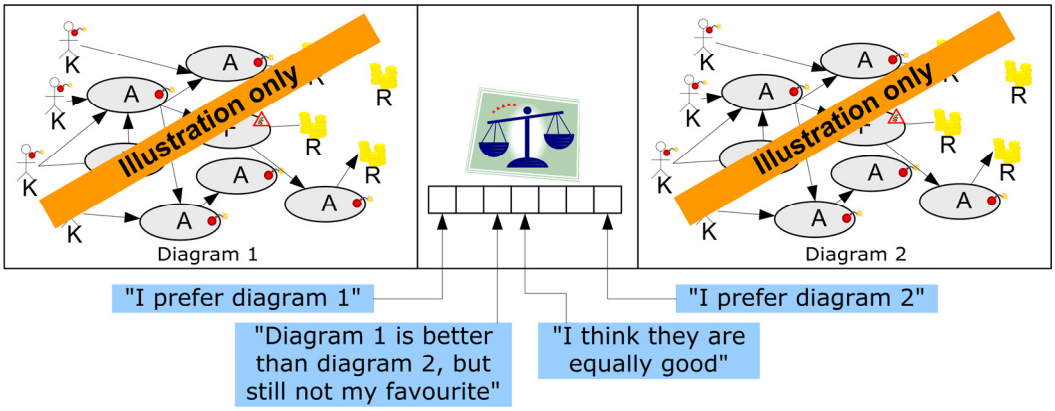


Figure 18 – Task explanation

A mark to the outmost left means "I prefer diagram 1", the next cell means "Diagram 1 is better than diagram 2, but still not my favorite". A mark in the middle means "I think they are equally good" while the outmost right means "I prefer diagram 2".

4.3 Hypotheses

The null hypothesis and alternative hypothesis for all tasks were:

- H_0: the modeling alternatives are equally preferred by the subjects.
- H_1: the modeling alternatives are not equally preferred.

4.4 Analysis method

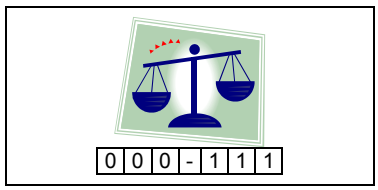


Figure 19 – Transforming data

The data was recoded into "0" and "1" to reflect which diagram that was preferred, an X in one of the three fields to the left was transformed into "0" and marks in the three fields to the right were recoded as "1". The middle field was interpreted as missing data to obtain a cut-off point between the two sides, (Figure 19). The frequency of 1's and 0's could thereafter be compared. The data was not distributed normally; therefore the non-parametric Chi-Square test [39] was used to identify possible statistical differences between the representations with a significance level of 0,05. The Chi-Square test uses the assumption that the frequency of 1's and 0's are the same. SPSS v13.0 on Windows XP was used for the statistical tests.

5 Results

The results from the statistical tests are presented in the three tables below. As mentioned, the Chi-Square test is based on the assumption the there will be equally many observations in each category (here: the number of "0" and "1" will be the same in each task). In the tables the difference between observed and expected observations is denoted as "Residual". "df" denotes the degrees of freedom which is the number variables that can vary when the sum of the variables have to add up to the observed frequency. E.g. in our case, once the frequency for the "1's" is set, the frequency for the "0's" is given from observed frequency minus the number of "1's", meaning that only the first variable is free and therefore the degrees of freedom equals 1. The outcome of the computation in a Pearsons Chi-Square test is the Chi-Square value (abbreviated to Chi-Square), which together with the degrees of freedom (df) gives us the asymptotic significance value from the Chi-Square distribution. "Asymp. Sig." is short for asymptotic significance (or p-value) which indicates whether the difference between two diagrams is statistically significant (in our test it must be below 0,05 to be significant). As we can see there was a significant difference between the diagram alternatives in four of the six tasks.

Table 1 shows the results from representing graph navigation: the likelihood of paths (TE2) and logical "and" (TE4).

Table 1 – Statistical results for graph navigation (TE2, TE4)

Likelihood				
		Frequencies		
TE2D1D2		*0*	*1*	*Total*
	Observed N	16	13	29
	Expected N	14,5	14,5	
	Residual	1,5	-1,5	
	Chi-Square	df	Asymp. Sig.	
TE2D1D2	0,310	1	0,577	

Logical "and"				
		Frequencies		
TE4D1D2		*0*	*1*	*Total*
	Observed N	30	2	32
	Expected N	16	16	
	Residual	14	-14	
TE4D3D1		*0*	*1*	*Total*
	Observed N	1	29	30
	Expected N	15	15	
	Residual	-14	14	
TE4D2D3		*0*	*1*	*Total*
	Observed N	2	30	32
	Expected N	16	16	
	Residual	-14	14	
	Chi-Square	df	Asymp. Sig.	
TE4D1D2	24,500	1	0,000	
TE4D3D1	26,133	1	0,000	
TE4D2D3	24,500	1	0,000	

The conclusion from the test of the likelihood of paths (TE2) was that neither of the modeling alternatives were preferred to the other (keep H_0). For logical "and" (TE4) we found significant differences between all modeling alternatives: D1 is preferred over both D2 and D3, and D3 is preferred over D2 (reject H_0).

Table 2 shows the results from representing vulnerabilities: shared vulnerabilities (TE5) and threats & vulnerabilities (TE6).

Table 2 – Statistical results for vulnerabilities (TE5, TE6)

Shared vulnerabilities				
		Frequencies		
TE5D1D2		*0*	*1*	*Total*
	Observed N	13	17	30
	Expected N	15	15	
	Residual	-2	2	
	Chi-Square	df	Asymp. Sig.	
TE5D1D2	0,533	1	0,465	

Threats & vulnerabilities				
		Frequencies		
TE6D1D2		*0*	*1*	*Total*
	Observed N	4	25	29
	Expected N	14,5	14,5	
	Residual	-10,5	10,5	
	Chi-Square	df	Asymp. Sig.	
TE6D1D2	15,207	1	0,000	

The test of representing vulnerabilities show that neither of the diagram alternatives in shared vulnerabilities (TE5) are preferred to the other (keep H_0), while for threats & vulnerabilities (TE6) the representation in D2 is significantly preferred (reject H_0).

Table 3 shows the results from representing the magnitude of risks (TE7) and unwanted incidents (TE3).

Table 3 – Statistical results for unwanted incident/risk (TE3, TE7)

The magnitude of an unwanted incident				
		Frequencies		
TE3D1D2		*0*	*1*	*Total*
	Observed N	15	16	31
	Expected N	15,5	15,5	
	Residual	-0,5	0,5	
TE3D3D1		*0*	*1*	*Total*
	Observed N	23	7	30
	Expected N	15	15	
	Residual	8	-8	
TE3D2D3		*0*	*1*	*Total*
	Observed N	13	16	29
	Expected N	14,5	14,5	
	Residual	-1,5	1,5	
	Chi-Square	df	Asymp. Sig.	
TE3D1D2	0,032	1	0,857	
TE3D3D1	8,533	1	0,003	
TE3D2D3	0,310	1	0,577	

The magnitude of an risk				
		Frequencies		
TE7D1D2		*0*	*1*	*Total*
	Observed N	9	22	31
	Expected N	15,5	15,5	
	Residual	-6,5	6,5	
TE7D3D1		*0*	*1*	*Total*
	Observed N	14	19	33
	Expected N	16,5	16,5	
	Residual	-2,5	2,5	
TE7D2D3		*0*	*1*	*Total*
	Observed N	25	6	31
	Expected N	15,5	15,5	
	Residual	9,5	-9,5	
	Chi-Square	df	Asymp. Sig.	
TE7D1D2	5,452	1	0,020	
TE7D3D1	0,758	1	0,384	
TE7D2D3	11,645	1	0,001	

From the Chi-Square test of TE3 we can conclude that the modeling alternative in D3 is preferred to that of D1, but not significantly to that of D2 (reject H_0). The result for TE7 shows that D2 is preferred to both D1 and D2 (reject H_0).

6 Discussion of modeling preferences

In the following the results for each task is discussed. The diagram alternatives are referred to as D1, D2, D3 meaning “diagram 1”, “diagram 2” etc. in the relevant figures. Whenever the results from the prestudy are relevant they will be mentioned, however since the experiment material was quite different from the one used in the prestudy we could not directly compare the results.

6.1 Representing graph navigation

The result from evaluating TE4 (Figure 6) showed that D1 was preferred to D3 which again was preferred to D2. Both alternatives with text labels were preferred to the non-text alternative. The same result was found in the prestudy using similar representations. The conclusion is that the subjects prefer a combination of a closed contour and a textual information label. Besides the possibility of being known from other notations, the and-gate shape used in D1 is clearer and more easily noticed than the other two alternatives. Using the and-symbol will also make the diagrams less cluttered compared to alternative D3.

In TE2 (Figure 7) the result showed that neither of the alternatives were preferred for this purpose. The same result was also obtained in the prestudy. It was a bit surprising that the thick line in D2 was not preferred over the thinner line in D1 since the modeling alternative is comparable to the use of thick lines in network maps to illustrate heavy traffic. A possible explanation for this is that the solidity of a line, in contrast to text, does not convey a unique interpretation. During field trials we have found it more helpful to show likelihood of paths by annotating the threat scenarios with likelihood estimates (numbers or text). The unwanted incident likelihood can then be estimated on the basis of the likelihood estimates of the threat scenarios that cause the incident. This has lead to the conclusion that we should use textual information labels to show likelihood of paths, rather than relying on graphical means only.

6.2 Representing vulnerabilities

Vulnerabilities are deficiencies with respect to the analysis object that expose assets to threats. During field trials we have experienced a need for representing vulnerabilities explicitly in the diagrams. Vulnerabilities may be modeled from several perspectives: in some situations one is interested in specifying which assets that are subjects to the same vulnerabilities, while in other cases we like to see which vulnerabilities a given threat may exploit.

In TE5 (Figure 8) the experimental results showed that neither of the two representations was preferred to illustrate shared vulnerabilities. The prestudy showed a preference for D2. This task only dealt with assets and their vulnerabilities, therefore we find it surprising that D2 was not preferred since this alternative had grouped all information concerning vulnerabilities below each asset. With respect to solving a task like TE5, it would have required much more effort to use D1 where it is up to the reader to deduce which vulnerabilities are relevant for which asset.

The same two diagram alternatives were also used in TE6 (Figure 9) aiming to identify the vulnerabilities a threat may exploit. This task was new in the experiment and therefore not tested in the prestudy. The result from TE6 showed that the alternative using the chain lock symbol (D1) was significantly preferred over the alternative representation. In field trials the vulnerability symbols has proved helpful in describing threat scenarios and an excellent support in treatment identification. One of the participants in a field trial actually pointed to a vulnerability and said “*we’ve fixed this one now, so you should close the chain lock*”.

Even though the result was inconclusive for TE5, the vulnerability symbol was preferred in TE6 and has therefore been implemented in the CORAS language.

6.3 Representing risks

In security analyses we often experience that people have a problem understanding the concept "risk" due to its abstract nature. A risk is an unwanted incident that has been given specific likelihood and consequence estimates. To make the risk-concept less abstract we investigated how it can be graphically represented in threat diagrams. In risk evaluation and treatment identification we find it useful to be able to specify the magnitude of both risks and unwanted incidents.

The results for TE3 (Figure 13) showed a significant difference between the representations where D3 (textual information) was preferred to the other two alternatives. This result was also found in the prestudy. A possible explanation is that the textual information in this case had a unique interpretation, while size or color could have many interpretations. Traditionally the size of an element in software modeling has not been given a specific interpretation. The dark color was not preferred to show magnitude, but we might have obtained a different result if we had used a color signalizing "danger" to a larger extent than grey . It would have been interesting to see the result if we had tested colors like red, orange and green, colors that are often used to illustrate high, medium and low risk.

It was quite surprising that the risk symbol alternative in TE7 (Figure 14, D3) received the lowest score. Also, the textual information alternative was preferred and was probably the representation that required the least effort to understand. It is important to consider the effects of giving size and color a specific semantic. Such a decision would mean that every time an element diverges from the standard size or color, the reader will expect it to mean something special, even if it was not the intention of the modeler. Also the degree of difference will play an important role. To stand out as different, the color or size has to be sufficiently different from the other elements to avoid confusion.

The overall findings from this experiment suggest that textual information in graphical models often is preferred to pure graphical means. In the already mentioned study of icons in stereotyping the original UML-based CORAS language [17] we compared a set of UML profile models to the standard UML models, both stereotyped with text labels (Figure 20). We found that the subjects receiving the special symbol version of the material completed more tasks than the other group and concluded that the text label stereotyping was not particularly significant. Comparing that conclusion to the findings in this investigation, we may draw the refined conclusion that text labels used as categorization labels may quickly be ignored by the reader because they do not convey important information needed to understand the diagram. Rather than categorizing elements with text labels, one should use text to communicate model information that the reader otherwise has to seek for in other documentation.

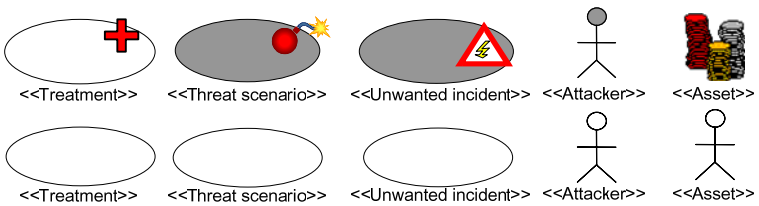


Figure 20 – Comparing stereotyping alternatives

7 Threats to validity

The main threats to validity of the empirical results of the investigation on which this report presents are described in this section. First of all, the diagrams we used were a compromise between realistic diagrams and naive notation-only tests. We cannot be sure that our results would be confirmed if we used real risk and threat diagrams. Real risk and threat diagrams contain more descriptive text and this additional text might make the text labels less visible. This weakness is difficult to avoid, and we will therefore validate our findings by testing them in real threat diagrams in a future field trial. With respect to text labels, it is important that the information is non-trivial and not used for categorization since they may be ignored (like the findings regarding UML stereotyping mentioned in 6.3).

The issues tested address a selection of several different security analysis modeling challenges. It would be both interesting and useful to also test other aspects and modeling alternatives, but this case we had to limit the size of the material to increase the likelihood of it being completed and handed in by the subjects.

The various modeling alternatives were tested on a population with considerable technical background. A background which is similar to that of most participants involved in security analyses and therefore considered to be a representative population.

Since the experiment material was distributed by email we could not control how the subjects chose to fill in the questionnaire (helping facilities, time spent etc.). We do not believe this affected the overall validity of the results since the subjects were asked to choose the alternatives they *preferred,* i.e. a completely subjective measure.

The subjects received no other introduction to security analysis than the short background provided with the material. We do not know whether the results would have been different if the subjects had been familiar with security analysis. One of the success factors of the CORAS language is however that it may be understood by people without a background in modeling and security analysis. Therefore we believe that it was a good idea to first test the language on this type of population, and rather test it on subjects with more security expertise at a later occasion.

8 Conclusions

In a security analysis it is often helpful to draw diagrams to illustrate threat and risk scenarios. It is essential that these diagrams are easily understood by people without training and experience in modeling and security analysis. This report deals with issues related to the following question: To what extent and in what way may we improve the reading and comprehension of such diagrams by adding information with text labels or simple graphical means?

The challenge was to identify the preferred way of capturing additional information. The investigation, which this report presents, focused on identifying modeling *preferences*.

Our empirical investigation showed that mechanisms like size- and color coding used to convey particular information in graphical models are less preferred by the subjects compared to textual information labels. The size or color of an element does not in general convey a unique interpretation in a model, while textual information is more specific. The subjects tend to prefer the representations where they get the most information without requiring them to interpret any additional graphical means.

8.1 Conclusions regarding "representing graph navigation"

To assess the various threats in security analysis, it is useful to describe the paths via which they are most likely to harm the assets. By highlighting (attract attention to) or "sub lightning" (remove attention from) paths in the threat diagram one may visualize the most or least likely paths. Our study of these mechanisms in task 2 (TE2) showed that neither thick nor dashed lines were well suited for conveying this. The reason may be that different types of lines do not have a specific interpretation and therefore understanding them requires additional explanation. We did not test text labels for this purpose as they most certainly would have made the diagrams messy.

We investigated various alternatives to represent the logical and-gate from fault tree analysis and electrical circuit logic [37]. The conclusion was clear: the classical logical and-gate symbol annotated with "and" (TE4) was preferred. In other words, when joining two paths in a graph with the meaning "and" using a graphical symbol with a text label on is clearly advisable. The result for the and-gate is considered representative for the or-gate too, since these should be modeled in the same style.

To summarize: neither of the alternatives investigated were preferred, except from the gate symbol. A possible reason for this may be that the likelihood of paths between threat scenarios are less interesting than the likelihood of the actual scenarios when it comes to establishing which path a threat may choose.

8.2 Conclusions regarding "representing vulnerabilities"

In task 5 and 6 (TE5, TE6) we saw that identifying the vulnerabilities along a path between a threat and an asset is simplest done by reading the diagram from the threat on the left to the assets on the right. Looking separately at the assets and their vulnerabilities, it is preferred to have the information grouped for each asset. We conclude that since the concept of a vulnerability is closely tied to that of an asset, it is more suitable to show shared vulnerabilities in diagrams that specify assets, like in a specialized *asset diagram* (see example in Figure 22).

To summarize: if the purpose is to show the relation between assets and vulnerabilities, the context in which the vulnerabilities reside is not as important as when the threat's preferred strategy is to be analyzed. Since we are interested in both views (threats and which vulnerabilities

they exploit, and assets and their vulnerabilities), the first should be covered in the threat diagrams and the second in the asset diagrams.

8.3 Conclusions regarding "representing risk"

An important aspect of a security analysis is to identify the risks that are above the tolerance level. It is therefore often helpful to be able to express the magnitude, or severity of both risks (TE7) and unwanted incidents (TE3) in the threat diagrams. We tested different perceptual mechanisms to illustrate magnitude and found that textual information in the diagrams is preferred to both size and color. We believe that this is because text can carry a more precise and unique interpretation than an element of varying size and color. If one decides that size has a meaning for one element, then one must consider this for the other types of elements as well and this may become too complex. For the CORAS language this means that textual information in the diagrams, possibly in combination with visual effects, is better than visual mechanisms alone. Our conclusion is that text labels should not be used as categorization labels in a diagram if they do not convey important information needed to understand the diagram. Rather than categorizing elements with text labels, one should use text to communicate diagram specific information that the reader otherwise has to seek for in other documentation.

To summarize: one should use textual information labels to indicate the severity of a risk or an unwanted incident, since the use of element size or color is too unambiguous.

8.4 How the results impacted the CORAS language

The CORAS security risk modeling language consists of several different diagram types that are used in different stages of the security analysis [16]. The result from the investigation which this report presents, had the most impact on the so-called *threat diagrams* used for risk identification and risk estimation, and the *risk diagrams*. Figure 21 shows a simple example of a threat diagram, which can be explained as follows: the target of analysis in this example is the information servers (email and file servers) in a company. The two most important assets with respect to the target are access to email and access to data files. The two threats that may affect both assets are hackers and employees, while the power supply and the electrician in combination may affect access to the file server. The hacker may exploit the old firewall solution to spread malicious code to one of the company's computers, while employees may unintentionally introduce the same type of code. After infecting a local PC, the malicious code may spread to the email and file servers. The malicious code may cause a server crash, making the email and company files unavailable. If the power supply fails and there are no electricians available, the small stand-by-unit cannot produce sufficient power to run the file server for more than two hours. The threat scenarios and unwanted incidents have been given likelihood estimates, and the consequences of the incidents are specified on the associations to the assets.

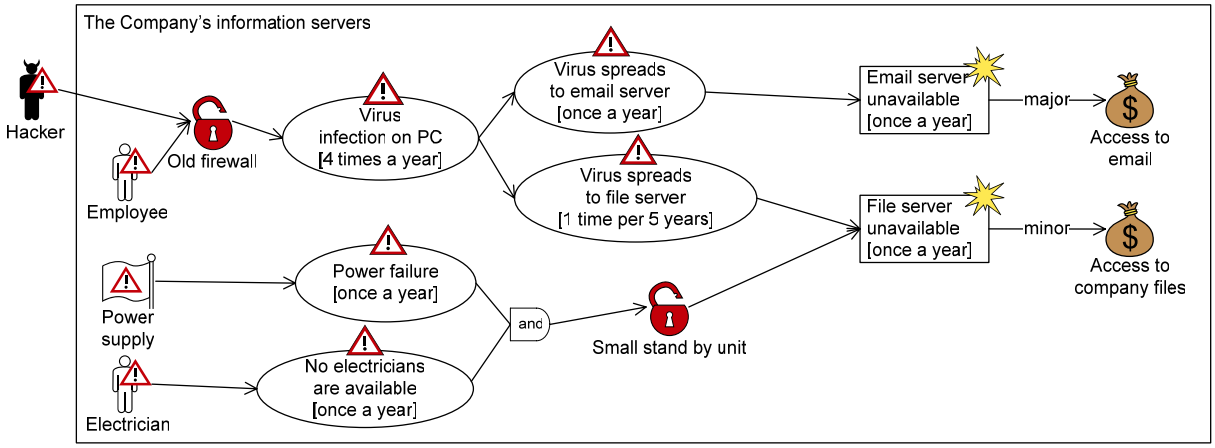


Figure 21 – Example of a threat diagram

The results from the investigations which this report presents, had the following impact:

- The logical gate symbols from FTA and circuit design were included in the language (only the and-gate is used in Figure 21, but the language provides a similar or-gate).
- The likelihood of paths is now shown by adding likelihood estimates as text labels to the threat scenarios (Figure 21).
- Vulnerabilities are now distributed in the diagram (Figure 21). This way of representing vulnerabilities attracts attention to the actual point where they are exploited. Too identify which vulnerabilities an asset is exposed to, one may follow each path from the asset in question towards the threats, but as we concluded this information is better presented in a specialized asset diagram (Figure 22).
- To illustrate the severity of a risk we provide a label indicating whether the risk is acceptable or unacceptable (Figure 23). The severity of unwanted incidents is not specified explicitly, but these are implicitly shown as textual consequence- and likelihood labels in the threat diagram (Figure 21).

The asset diagram in Figure 22 specifies which vulnerabilities each asset is subject to. The risk diagram in Figure 23 illustrates which risks that are unacceptable/acceptable, which threats that may cause them and which assets that may be harmed. As we see, the employee and the hacker may cause both risks, while a power failure and the electrician may only make the file server unavailable.

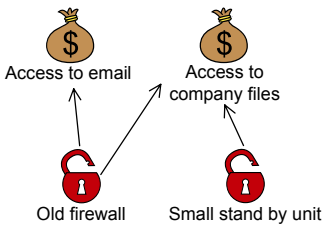


Figure 22 – Asset diagram: vulnerabilities

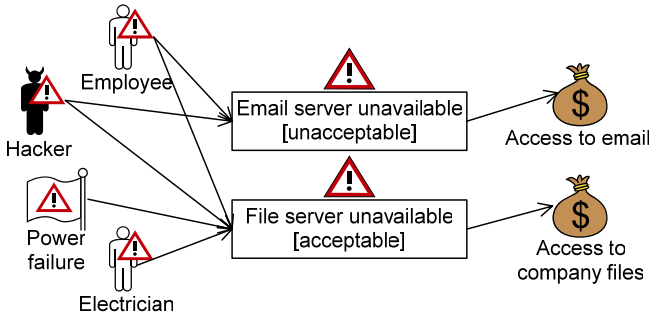


Figure 23 – Risk diagram

Acknowledgements

The research which this report presents has been funded by the Research Council of Norway project SECURIS (152839/220). The authors are very grateful to the subjects that participated voluntarily. We also thank Jan Heim for his help and guidance, and the SECURIS team: Gyrd Brændeland, Heidi Dahl, Iselin Engan, Mass Soldal Lund, Fredrik Seehusen, Bjørnar Solhaug and Fredrik Vraalsen.

References

[1] Alexander, I., "Misuse cases: Use cases with hostile intent", *IEEE Software*, vol. 20 (1), pp. 58-66, 2003.

[2] AS/NZS4360, Australian/New Zealand Standard for Risk Management: Standards Australia/Standards New Zealand, 2004.

[3] Becker, R. A., Eick, S. G., and Wilks, A. R., "Visualizing Network Data", *IEEE Transactions on Visualization and Computer Graphics*, vol. 1 (1), pp. 16-21, 1995.

[4] Bratthall, L. and Wohlin, C., "Is it Possible to Decorate Graphical Software Design and Architecture Models with Qualitative Information? - An Experiment", *IEEE Transactions on Software Engineering*, vol. 28 (12), pp. 1181-1193, 2002.

[5] Cahill, M.-C. and Carter, R. C. J., "Color code size for searching displays of different density", *Human Factors*, vol. 18 (3), pp. 273-280, 1976.

[6] Cavanagh, J. P., "Relationship between the immediate memory span and the memory search rate", *Psychological Review*, vol. 79, pp. 525-530, 1972.

[7] Chattratichart, J. and Kuljis, J., "An Assessment of Visual Representations for the 'Flow of Control'", in Proc. 12th Workshop of the Psychology of Programming Interest Group (PPIG'00), pp. 45-48, 2000.

[8] Christ, R. E., Research for evaluating visual display codes: an emphasis on colour coding., in *Information Design: The design and evaluation of signs and printed material*, R. Easterby and H. Zwaga, Eds.: John Wiley and Sons Ltd, 1984, pp. 209-228.

[9] Christ, R. E., "Review and analysis of color coding research for visual displays", *Human Factors*, vol. 17 (6), pp. 542-570, 1975.

[10] Cleveland, W. S. and McGill, R., "Graphical perception and graphical methods for analyzing scientific data." *Science*, vol. 229, pp. 828-833, 1985.

[11] den Braber, F., Mildal, A.-B., Nes, J., Stølen, K., and Vraalsen, F., "Experiences from Using the CORAS Methodology to Analyze a Web Application", *Journal of Cases on Information Technology*, vol. 7 (3), pp. 110-130, 2005.

[12] Freeman, R. E., *Strategic Management: A Stakeholder Approach*: Ballinger Publishing, 1984.

[13] Goodman, N., *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis: Hackett, 1976.

[14] HB231, Information security risk management guidelines: Standards Australia/Standards New Zealand, 2000.

[15] HB231, Information security risk management guidelines: Standards Australia/Standards New Zealand, 2004.

[16] Hogganvik, I. and Stølen, K., "A Graphical Approach to Risk Identification, Motivated by Empirical Investigations", in Proc. MoDELS 2006 (LNCS 4199), pp. 574-588, 2006.

[17] Hogganvik, I. and Stølen, K., "On the Comprehension of Security Risk Scenarios", in Proc. 13th Int. Workshop on Program Comprehension (IWPC'05), pp. 115-124, 2005.

[18] Hogganvik, I. and Stølen, K., "Risk Analysis Terminology for IT-systems: does it match intuition?" in Proc. Int. Symposium on Empirical Software Engineering (ISESE'05), pp. 13-23, 2005.

[19] IEC60300-3-9, Event Tree Analysis in Dependability management - Part 3: Application guide - Section 9: Risk analysis of technological systems., 1995.

[20] IEC61025, Fault Tree Analysis (FTA), 1990.

[21] ISO/IEC13335, Information technology - Guidelines for the management of IT Security (Part 3), 1998.

[22] ISO/IEC13335, Information technology - Security techniques - Management of information and communications technology security (Part 1), 2004.

[23] Jacobson, I., Christerson, M., Jonsson, P., and Övergaard, G., *Object-Oriented Software Engineering: A Use Case Driven Approach*: Addison-Wesley, 1992.
[24] Jensen, F. V., *Bayesian Networks and Decision Graphs*: Springer, 2002.
[25] Jubis, R. M. T., "Coding effects on performance in a process control task with uniparameter and multiparameter displays", *Human Factors*, vol. 32 (3), pp. 287-297, 1990.
[26] Jürjens, J., *Secure Systems Development with UML*: Springer, 2005.
[27] Kontio, J., Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation, in *Dept. of Computer Science and Engineering*: Helsinki University of Technology, 2001.
[28] Krogstie, J., Conceptual Modeling for Computerized Information Systems Support in Organizations, in *Faculty of Electrical Engineering and Computer Science*: The Norwegian Institute of Technology, The University of Trondheim, 1995.
[29] Linos, P. K., Aubet, P., Dumas, L., Helleboid, Y., Lejeune, D., and Tulula, P., "Visualizing program dependencies: An experimental study", *Software Practice and Experience*, vol. 24 (4), pp. 387-403, 1994.
[30] Lodderstedt, T., Basin, D., and Doser, J., "SecureUML: A UML-Based Modeling Language for Model-Driven Security", in Proc. UML'02, LNCS (2460), pp. 426-441, 2002.
[31] Lund, M. S., den Braber, F., Stølen, K., and Vraalsen, F., "A UML profile for the identification and analysis of security risks during structured brainstorming." SINTEF ICT, Tech. report STF40 A03067, 2004.
[32] Lund, M. S., Hogganvik, I., Seehusen, F., and Stølen, K., "UML profile for security assessment", SINTEF ICT, Tech. report STF40 A03066, 2003.
[33] OMG, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms", Object Management Group, 2006.
[34] OMG, "Unified Modeling Language (UML): Superstructure, version 2.0", Object Management Group, 2005.
[35] Schneider, W. and Shiffren, R. M., "Controlled and automatic human information processing I: Detection, search, and attention." *Psychological Review*, vol. 84, pp. 1-66, 1977.
[36] Schneier, B., "Attack trees: Modeling security threats", *Dr. Dobb's Journal*, vol. 24 (12), pp. 21-29, 1999.
[37] Sedra, A. S. and Smith, K. C., *Microelectronic Circuits*: Oxford University Press, 2003.
[38] Shneiderman, B., *Designing the User Interface*: Addison-Wesley, 1992.
[39] Siegel, S. and Castellan, J., *Non-parametric Statistics for the Behavioural Sciences*, 2 ed: McGraw-Hill International Editions, 1988.
[40] Sindre, G. and Opdahl, A. L., "Eliciting Security Requirements by Misuse Cases", in Proc. TOOLS-PACIFIC, pp. 120-131, 2000.
[41] Sindre, G. and Opdahl, A. L., "Templates for Misuse Case Description", in Proc. Workshop of Requirements Engineering: Foundation of Software Quality (REFSQ'01), pp. 125-136, 2001.
[42] Smith, L. and Thomas, D., "Color versus shape coding in information displays", *Journal of Applied Psychology*, vol. 48 (3), pp. 137-146, 1964.
[43] Tan, K. C., Effects of Stimulus Class on Short -Term Memory Workload in Complex Information Displays, in *Department of Industrial Engineering and Operations Research*: Virginia Technical University, 1990.
[44] Treisman, A. and Gormican, S., "Feature analysis in early vision: Evidence from search asymmetries", *Psychological Review*, vol. 95 (1), pp. 15-48, 1988.
[45] Ware, C., *Information Visualization: Perception for Design*, 2 ed: Elsevier, 2004.

[46] Wertheimer, M., *Laws of Organization in Peceptual Forms [English Translation of: "Untersuchungen zur Lehre von der Gestalt", II, Psychologische Forschung 4 (1923), pp. 301 –350.] In Willis D. Ellis (ed.): A Source Book of Gestalt Psychology*: Routledge & Kegan Paul, 1923.

[47] Wickens, C. D., *Engineering Psychology and Human Performance*, 2 ed: HarperCollins, 1992.

[48] Winn, W., "An Account of How Readers Search for Information in Diagrams", *Contemporary Educational Psychology*, vol. 18, pp. 162-185, 1993.

[49] Winn, W., "Perceptual strategies used with flow diagrams having normal and unanticipated formats", *Perceptual and Motor Skills*, vol. 57, pp. 751-762, 1983.

[50] Winn, W., "The role of diagrammatic representation in learning sequences, identification, and classification as a function of verbal and spatial ability", *Journal of Research in Science Teaching*, vol. 19, pp. 79-89, 1982.

[51] Winn, W. and Solomon, C., "The effect of the rhetorical structure of diagrams on the interpretation of simple sentences", *University of Washington, unpublished manuscript*, 1991.

[52] Aagedal, J. Ø., Braber, d. F., Dimitrakos, T., Gran, B. A., Raptis, D., and Stølen, K., "Model-based risk assessment to improve enterprise security", in Proc. Enterprise Distributed Object Communication (EDOC'02), pp. 51-64, 2002.

List of Tables

Table 1 – Statistical results for graph navigation (TE2, TE4) 13
Table 2 – Statistical results for vulnerabilities (TE5, TE6) 14
Table 3 – Statistical results for unwanted incident/risk (TE3, TE7) 14
Table 4 – Results for the tasks left out of the report 34
Table 5 – Results from SPSS 41
Table 6 – Task conclusions 42

List of Figures

Figure 1 – The conceptual foundation 3
Figure 2 – Percentage of correct answers for each category (mean) 4
Figure 3 – Typical threat scenario situation 5
Figure 4 – Node-link configurations 5
Figure 5 – The AND-gate symbol 6
Figure 6 – The diagram alternatives for: “most likely path through a graph” 6
Figure 7 – The diagram alternatives for: “logical AND in graph paths” 7
Figure 8 – The diagram alternatives for: “shared vulnerabilities” 8
Figure 9 – The diagram alternatives for: “which vulnerabilities are exploited by which threats?” 8
Figure 10 – Color 9
Figure 11 – Size 9
Figure 12 – Shapes 10
Figure 13 – The diagram alternatives for: “the magnitude of unwanted incidents” 10
Figure 14 – The diagram alternatives for: “the magnitude of risks” 10
Figure 15 – The subjects’ years of work experience 11
Figure 16 – The subjects’ task experience 11
Figure 17 – The basic threat diagram 11
Figure 18 – Task explanation 12
Figure 19 – Transforming data 12
Figure 20 – Comparing stereotyping alternatives 16
Figure 21 – Example of a threat diagram 20
Figure 22 – Asset diagram: vulnerabilities 20
Figure 23 – Risk diagram 20
Figure 24 – Example of a Riskit graph 27
Figure 25 – Bayesian network 27
Figure 26 – Example 28
Figure 27 – Task set-up 29

Appendix A – Related modeling notations

The starting point for our work was a UML [34] profile [33] developed in the EU-funded research project CORAS [52]. As a result of our empirical investigations with the aim to satisfy the modeling needs in security analysis, the CORAS language has undergone numerous modifications and refinements since the completion of the CORAS project, and the current version is no longer a UML profile.

Misuse cases [1, 40, 41] was an important source of inspiration in the development of the above mentioned UML profile. A misuse case is a kind of UML use case [23] which characterizes functionality that the system should *not* allow. The use case notation is often employed for high level specification of systems and considered to be one of the most understandable notations in UML. A misuse case can be defined as "a completed sequence of actions which results in loss for the organization or some specific stakeholder" [41]. To obtain more detailed specifications of the system, the misuse cases can be specialized into e.g. sequence or activity diagrams.

There are a number of security oriented extensions of UML, e.g. UMLSec [26] and SecureUML [30]. These and other related languages have however all been designed to capture security properties and security aspects at a more detailed level than the CORAS language. Moreover, they do not emphasize brainstorming sessions as in our case. SecureUML: modeling access control policies in the relationship of model-driven software development. UMLsec: an extension of UML that gives the possibility to model security related features like access control, integrity and confidentiality.

Fault tree is a tree-notation used in fault tree analysis (FTA) [20]. The top node represents an unwanted incident, or failure, and the different events that may lead to the top event are modeled as branches of nodes, with the leaf node as the causing event. The probability of the top node can be calculated on the basis of the probabilities of the leaf nodes and the logical gates "and" and "or". The CORAS threat diagrams often look a bit like fault trees, but may have more than one top node. Computing the top node probability of a fault tree requires precise quantitative input, which is rarely available for incidents caused by human errors or common cause failures. It is possible to model fault trees using the CORAS language, but it also allows qualitative likelihood values as input.

Event tree analysis (ETA) [19] focuses on illustrating the consequences of an event and the probabilities of these. It is often used as an extension of fault trees to create "cause-consequence" diagrams. Event trees can also to a large extent be simulated in our notation.

Attack trees [36] aim to provide a formal and methodical way of describing the security of a system based on the attacks it may be exposed to. The notation uses a tree structure similar to fault trees, with the attack goal as the top node and different ways of achieving the goal as leaf nodes. Our language supports this way of modeling, but facilitates in addition the specification of the attack initiators (threats) and the harm caused by the attack (damage to assets).

The Riskit method [27] includes a risk modeling technique that makes it possible to specify factors that may influence a software development project (example in Figure 24 from [27]). It has similarities to the CORAS language, but targets project risks: "When dealing with software risks, the risks caused by the *operation* of software are often also considered. As our primary focus has been to support software development organizations in *developing* their products, we have excluded the evaluation and management of risks that might occur during the operation of software" [27] (p. 12).

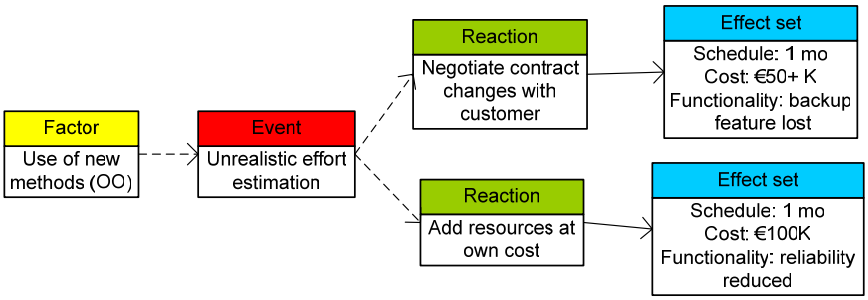


Figure 24 – Example of a Riskit graph

The Riskit modeling approach is related to the CORAS approach, but targets project risks and utilizes different concepts and definitions. Since the CORAS method targets security risks, its definitions are taken from information security related standards [2, 14, 21, 22]; Riskit, on the other hand uses its own definitions inspired by e.g. organizational strategy research [12]. As an example of the difference in concepts, Riskit lacks "threat", possibly because it unusual to experience deliberate harmful actions towards a software development project.

A Bayesian network [24] is a directed acyclic graph that represents a probability distribution. The nodes in the network represent variables and links represent relationships between the variables (Figure 25).

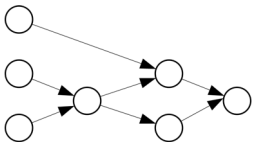


Figure 25 – Bayesian network

The root nodes are independent, while the probabilities of the intermediate nodes depend on the probabilities of its preceding nodes. A Bayesian network lets you decompose a probability distribution into a set of local distributions. These local distributions are used to compute the resulting top-node probability. A Bayesian network specifies both the qualitative structure of the case and lets you compute the quantitative probability values. Our modeling technique also provides a means to describing the behavior of a threat qualitatively. The statistical distributions that a Bayesian network requires to perform probability computations are seldom available in our types of analyses and precise statistical computations have therefore not played an important role in our work. However, if such statistics are available and all the assumptions of constructing a Bayesian network are met (statistical independence of the distributions), the CORAS language can be used to describe a Bayesian network.

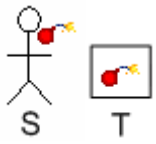
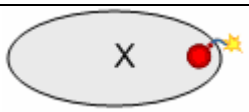
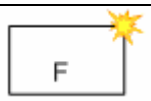
Appendix B – Experiment Material

This appendix contains the experiment material used in the main study. The material has been kept in its original format, to the extent that was possible. However, seeing that the original task set-up (illustrated in Figure 27) was space consuming it has been left out in this report. Two tasks have been discarded from the analysis: TE1 and TE8. The reasons for this are explained at the end of this appendix.

B.I Modeling questionnaire

This questionnaire is about threat modeling. We present various ways of modeling the same situation, and your task is to prioritize between them and choose the diagram you think is best! This type of modeling is meant to support the risk analyst in situations where the participants in meetings have different background and expertise (technical and non-technical). The diagrams are meant to help them understand the system and each other better and faster.

Symbols and definitions:

Symbols			
 (2 symbols)	**Threat**: a human/non-human that intentionally or non-intentionally can cause an event that may harm an asset.		**Threat scenario**: a sequence of events corresponding to a threat exploiting vulnerabilities, potentially causing unwanted incidents.
	**Asset**: something of value which needs to be protected.		**Unwanted incident**: an undesired event that may reduce the value of an asset.
	**Vulnerability**: a weakness or a deficiency that a threat can exploit.		

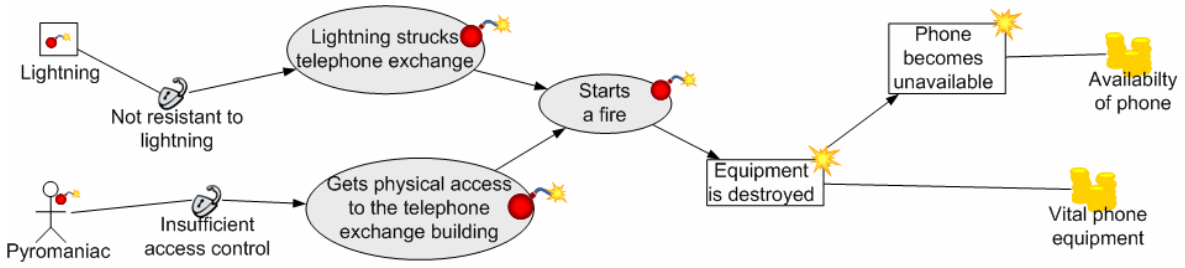


Figure 26 – Example

About you:

A) Gender: ☐ Female ☐ Male

B) Age: ☐ Under 25 years old ☐ 26-35 ☐ 36-45 ☐ 46-55 ☐ 56-65 ☐ Over 65

C) Current position/title:

D) Years of work experience:

- ☐ Less than 1 year
- ☐ 1-3 years
- ☐ 4-10 years
- ☐ Over 10 years

E) Which tasks have you experience from?

- ☐ System design
- ☐ System development
- ☐ System maintenance
- ☐ System security
- ☐ System testing
- ☐ System quality evaluation
- ☐ System marketing/sale

Prioritizing different ways of modeling threat diagrams:
Compare two and two diagrams and choose the one you prefer by marking with an "X" in the table below the weight symbol, as shown in this example:

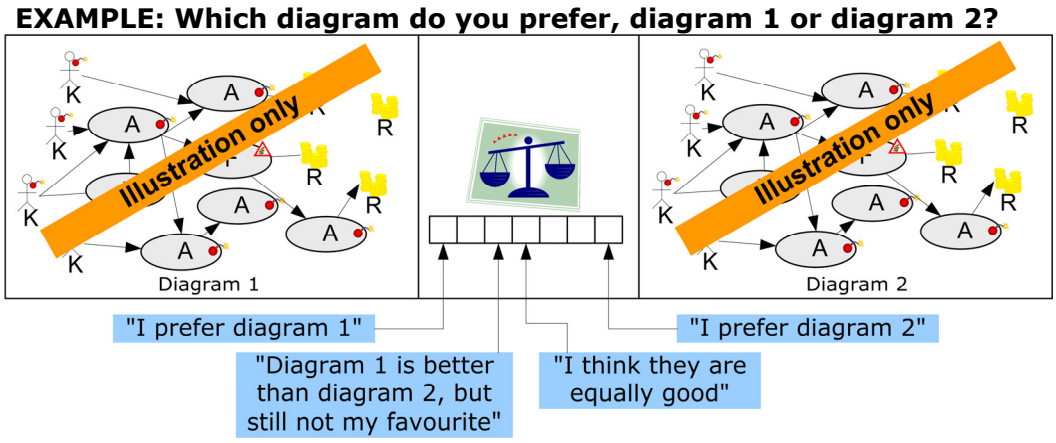


Figure 27 – Task set-up

TE1 - Which diagram is best at illustrating that threat scenario N can cause the unwanted incident F?

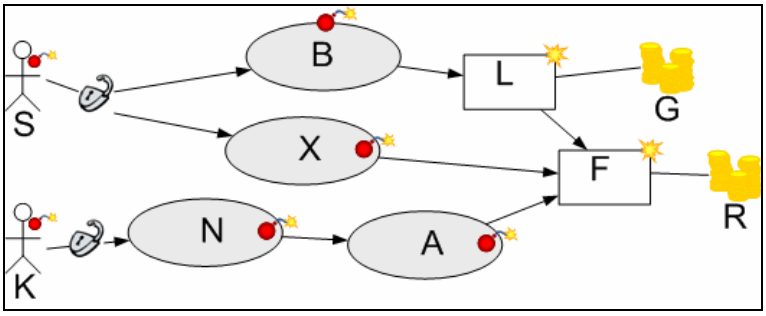


Diagram 1

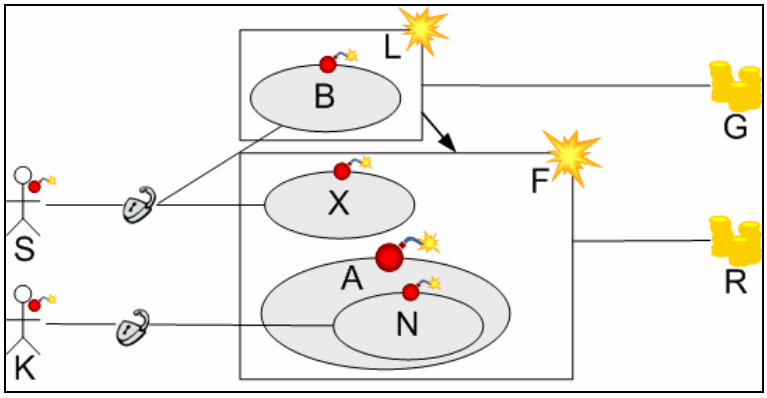


Diagram 2

TE2 - Which diagram is best at illustrating that it is more likely that threat K causes the unwanted incident T than L or F?

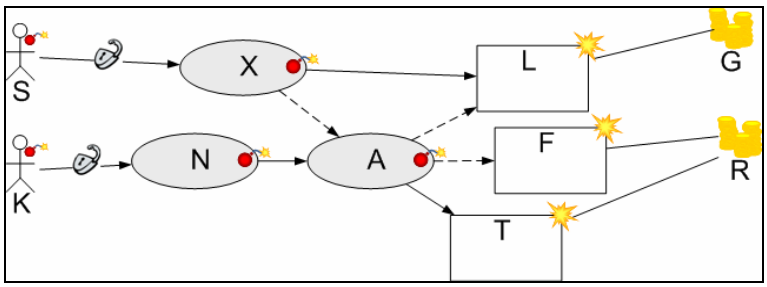


Diagram 1

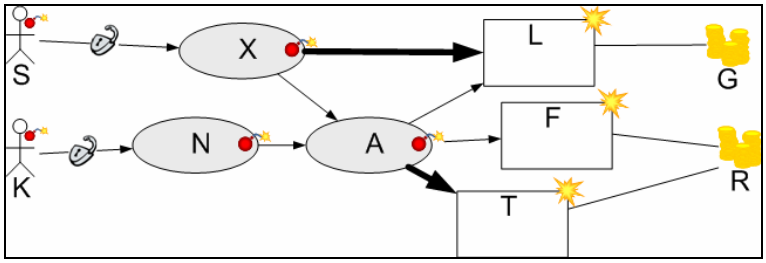


Diagram 2

TE3 - Which diagram is best at illustrating that unwanted incident L is more serious than unwanted incident F?

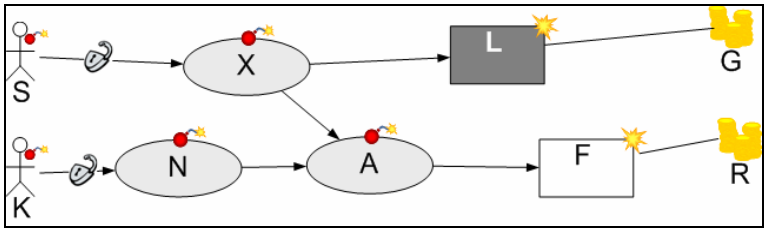


Diagram 1

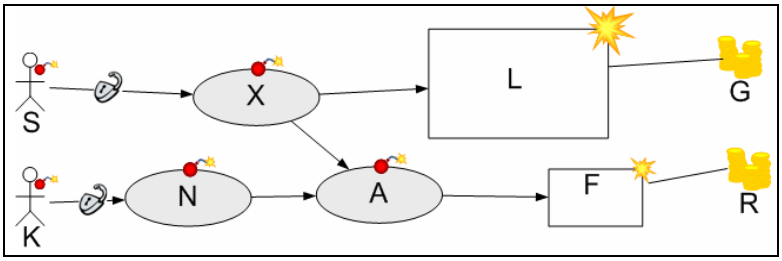


Diagram 2

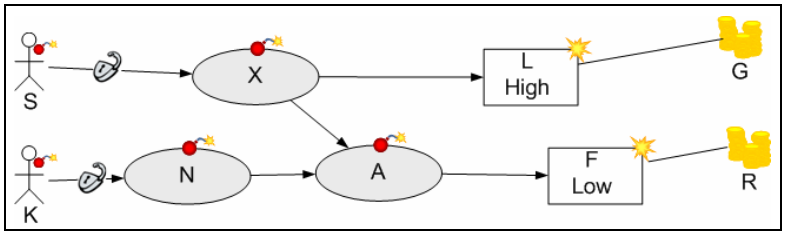


Diagram 3

TE4 - Which diagram is best at illustrating that both X and N must happen before A can happen?

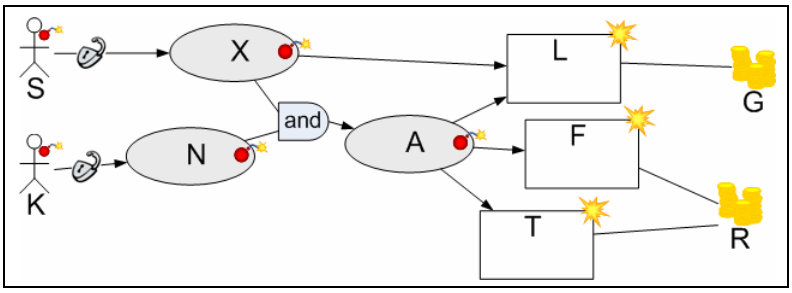


Diagram 1

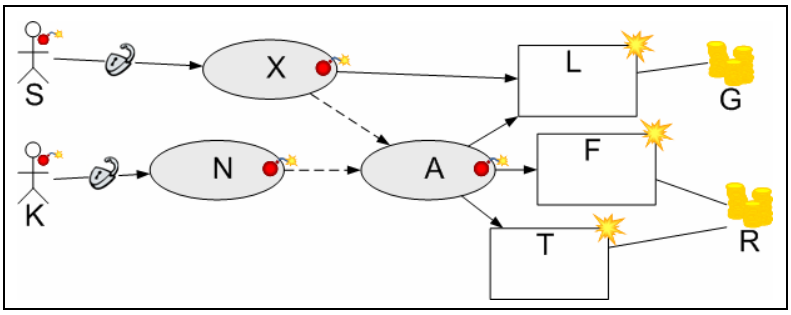


Diagram 2

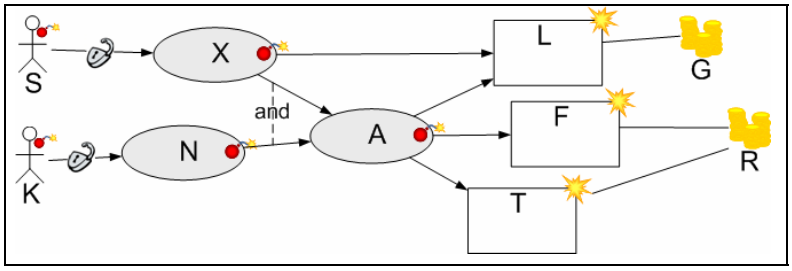


Diagram 3

TE5 - Which diagram is best at illustrating that the asset G and R share the same vulnerability "accessible remotely"?

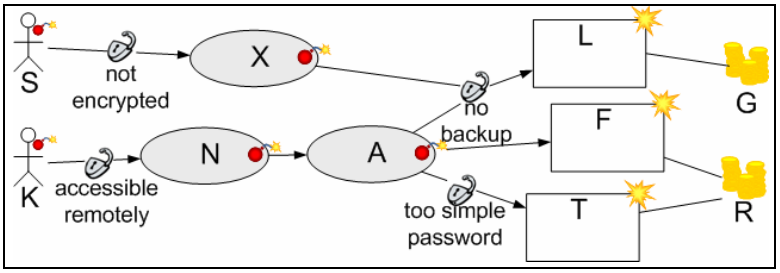


Diagram 1

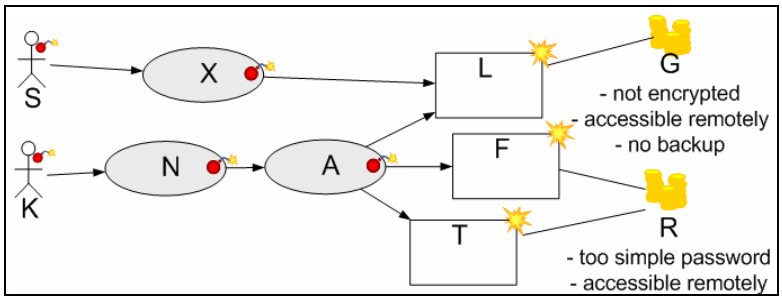


Diagram 2

TE6 - Which diagram is best at illustrating that only threat K can exploit the vulnerability "too simple password"?

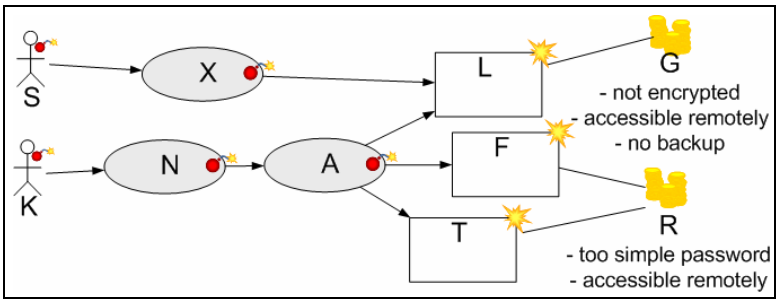


Diagram 1

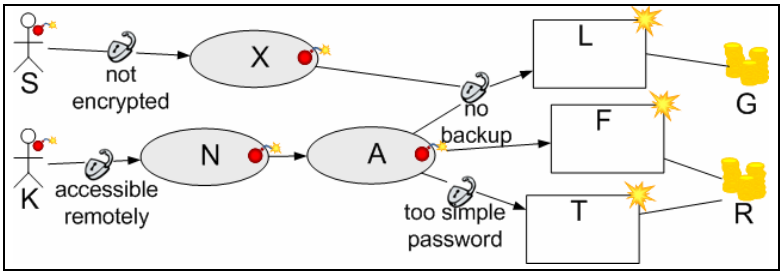


Diagram 2

TE7 - Which diagram is best at illustrating that unwanted incident F poses the largest risk for asset R?

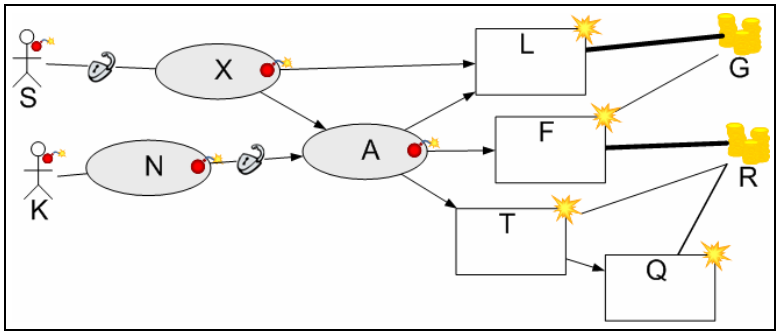


Diagram 1

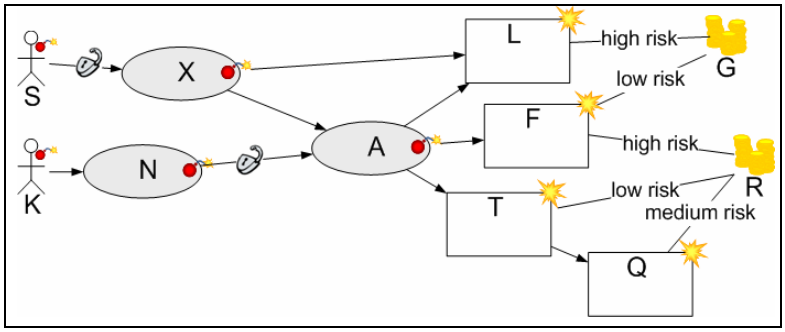


Diagram 2

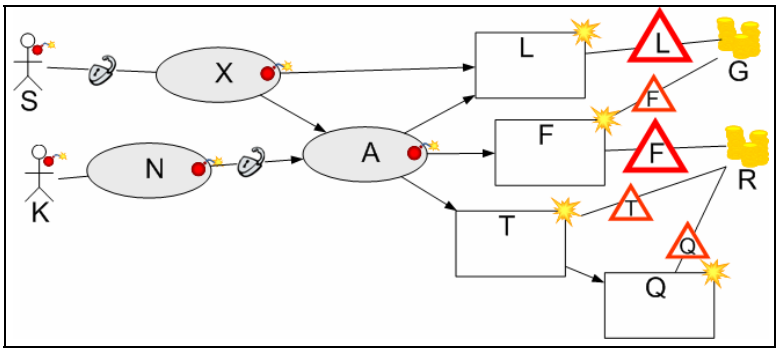


Diagram 3

TE8 - Which diagram is best at illustrating that unwanted incident F represents two *different* risks?

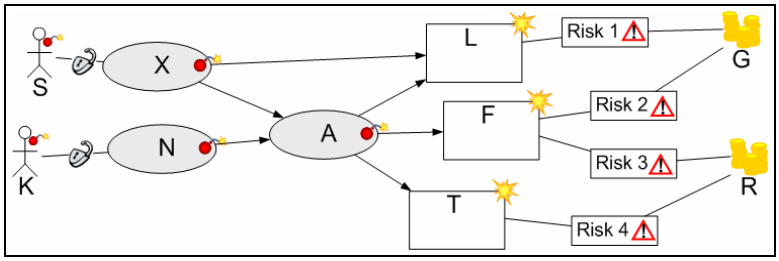


Diagram 1

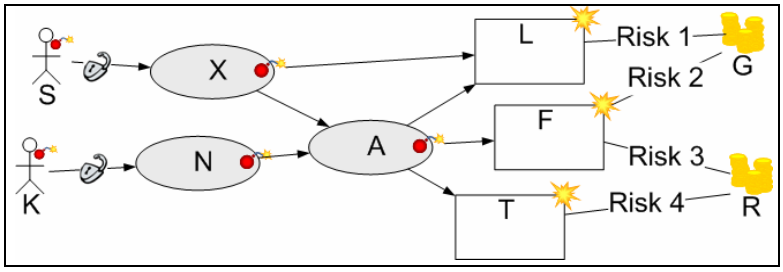


Diagram 2

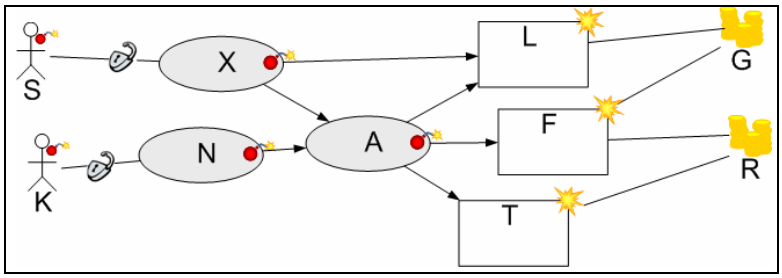


Diagram 3

B.II Tasks that were left out of the analysis

Two tasks were found either superfluous or too inconsistent to be part of the report. TE1 was similar to TP8 in the prestudy, but using the new notation. The alternative representation in Diagram 1 has later been found to be too impractical for real use. A diagram with several inclusions would imply several large boxes in a potentially cluttered and complex pattern. Since this task had only two modeling alternatives, it seemed unfair to compare a bad modeling alternative with another, because this could an unintentional advantage.

The other modeling task that was omitted from the analysis was TE8. The intention behind the task was to investigate the difference between no text labels, text labels and boxes with text labels. By mistake, a risk icon was also included in the box, which made it more complex than the other two. Any effects, either positive or negative, with respect to this alternative could not have been said to have been caused by the use of boxes, textual labels or icons. Due to this inconsistency we decided to leave it out from this investigation.

Nevertheless, we provide the statistical results for TE1 and TE8 in the table below.

Table 4 – Results for the tasks left out of the report

TE1				
		Frequencies		
TE1D1D2		*0*	*1*	*Total*
	Observed N	30	3	33
	Expected N	16,5	16,5	
	Residual	13,5	-13,5	
	Chi-Square	df	Asymp. Sig.	
TE1D1D2	22, 091	1	0,000	

TE8				
		Frequencies		
TE8D1D2		*0*	*1*	*Total*
	Observed N	17	9	26
	Expected N	13	13	
	Residual	4	-4	
TE8D3D1		*0*	*1*	*Total*
	Observed N	12	20	32
	Expected N	16	16	
	Residual	-4	4	
TE8D2D3		*0*	*1*	*Total*
	Observed N	19	12	31

	Expected N	15,5	15,5	
	Residual	3,5	-3,5	
	Chi-Square	df	Asymp. Sig.	
TE8D1D2	2,462	1	,117	
TE8D3D1	2,000	1	,157	
TE8D2D3	1,581	1	,209	

Appendix C – Prestudy

This appendix gives an overview of the prestudy, its design, material and results. Some of the material was also used in the main study, but the results cannot be compared directly since the material has been slightly modified.

C.I Prestudy design

The subjects in the prestudy were 11 master students in informatics at University of Oslo. The survey was taken as a part of one of the classes in the course "INF5150-Unassailable IT-systems", autumn 2005 (http://www.uio.no/ studier/emner/matnat/ifi/INF5150/h05/). The subjects had to assess several modeling alternatives and select the one they preferred the most.

C.II Prestudy material

The prestudy aimed to investigate the original UML profile notation [32] in addition to various new alternative modeling notations. The survey consisted of 7 tasks, numbered from 8-14 (task 1-7 were conventional educational tasks and therefore not part of the experiment). In the result section the following numbering convention is used: TP8D1D2 meaning "Task 8 in Prestudy, Diagram 1 compared to Diagram 2". Each task includes a question and 2-3 diagram alternatives. The material set-up and introduction was similar to the one presented in Appendix B.

TP8 - Which diagram shows best that threat scenario N can cause the unwanted incident of incident scenario L?

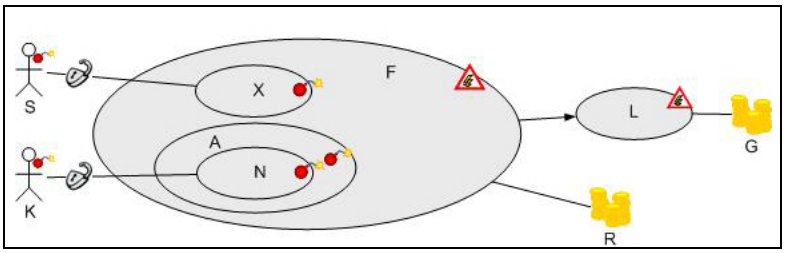


Diagram 1

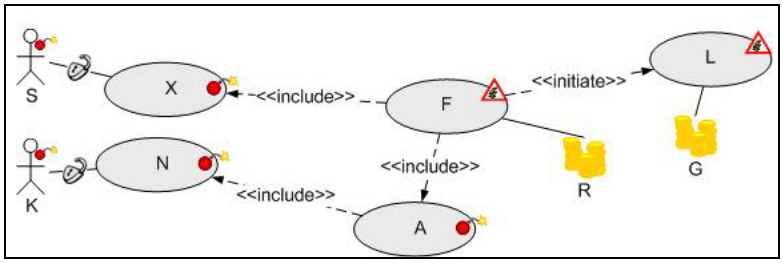


Diagram 2

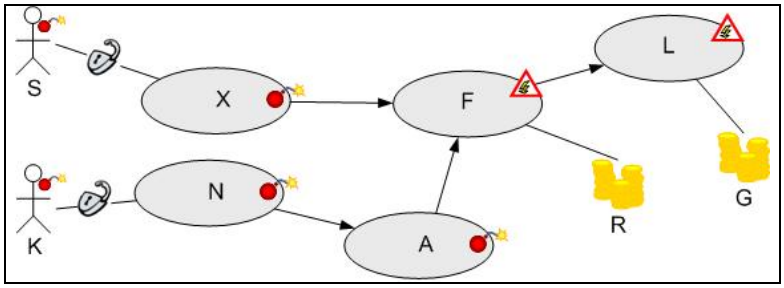


Diagram 3

TP9 - Which diagram shows best that the unwanted incident of incident scenario L is more serious than unwanted incident of F?

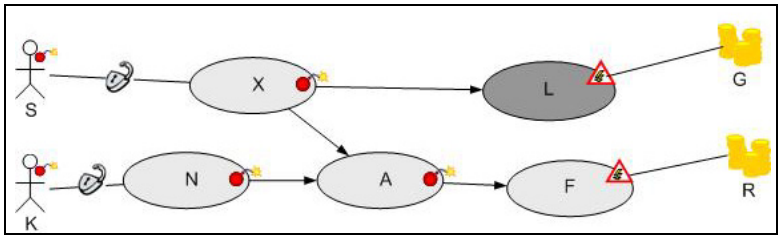


Diagram 1

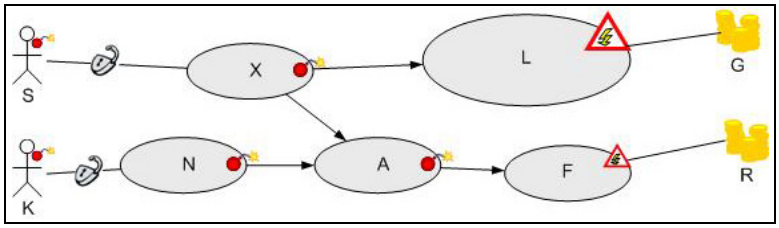


Diagram 2

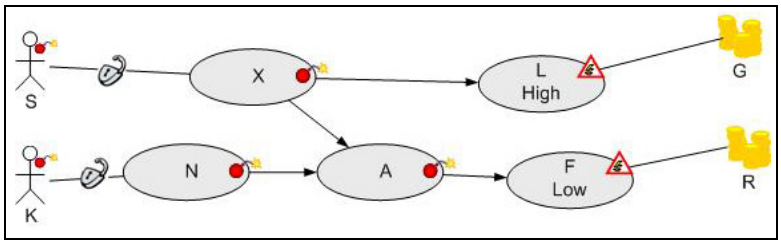


Diagram 3

TP10 - Which diagram shows best that it is more likely that T happens after A, than L or F?

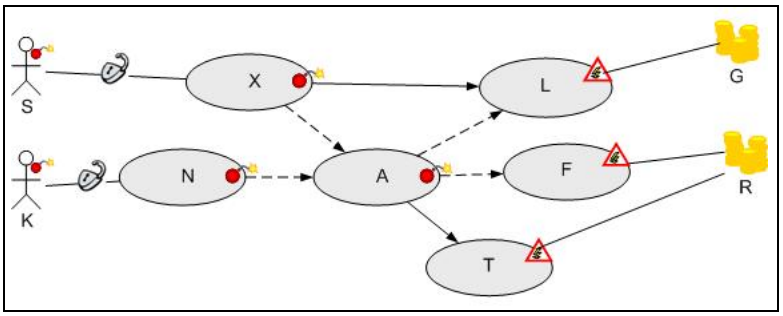


Diagram 1

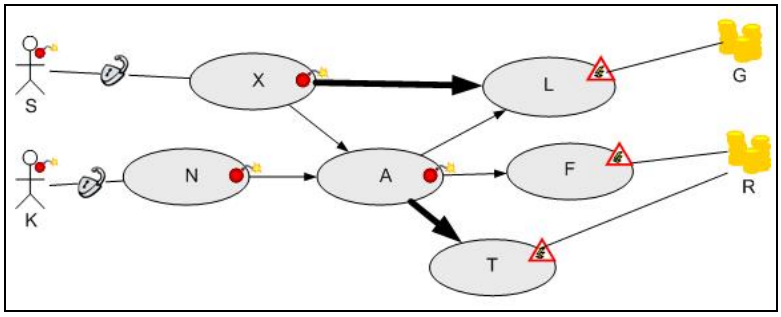


Diagram 2

TP11 - Which diagram shows best that both X and N must happen before A can happen?

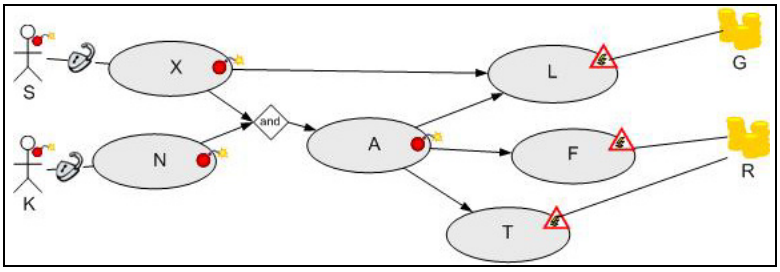


Diagram 1

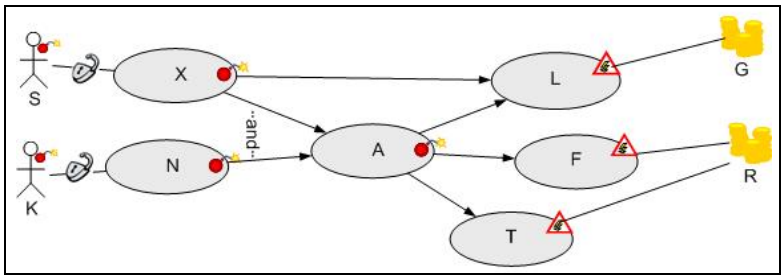


Diagram 3

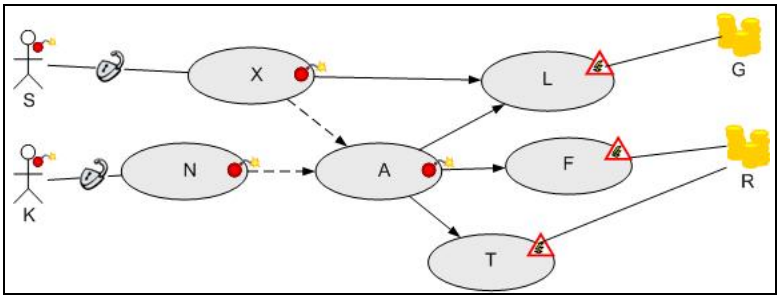


Diagram 2

TP12 - Which diagram shows best that the asset G and R share the same vulnerability?

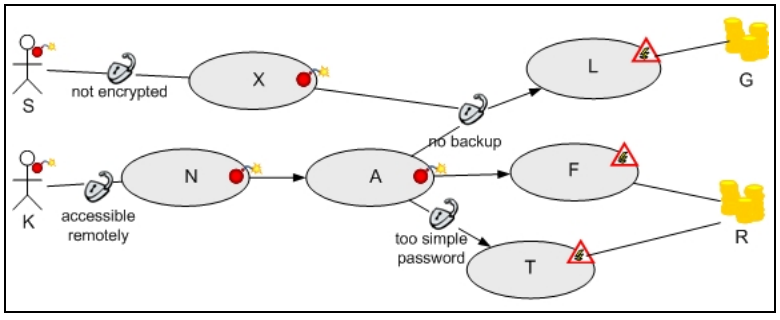


Diagram 1

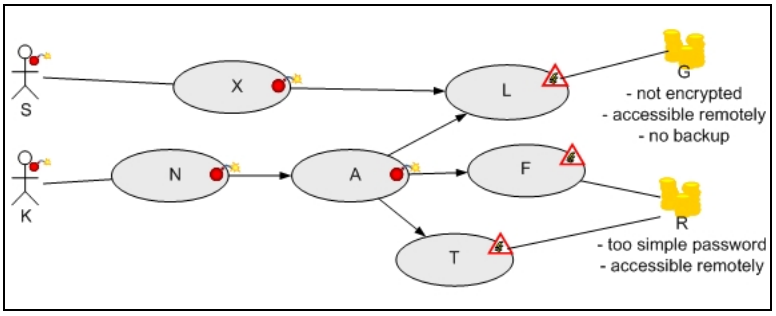


Diagram 2

TP13 - Which diagram shows best that unwanted incident F poses the largest risk for asset R?

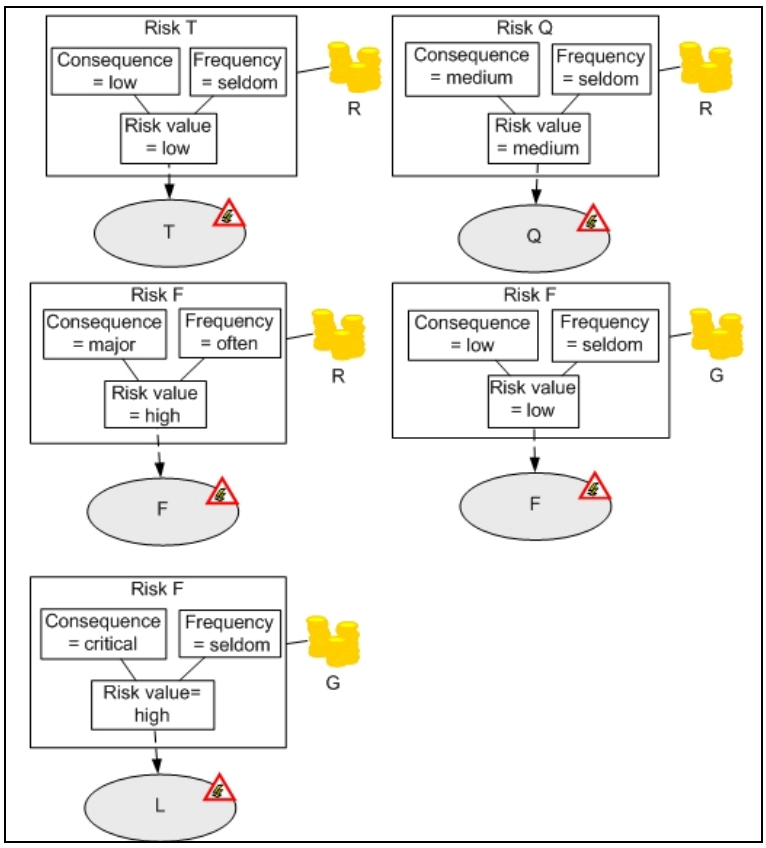


Diagram 1

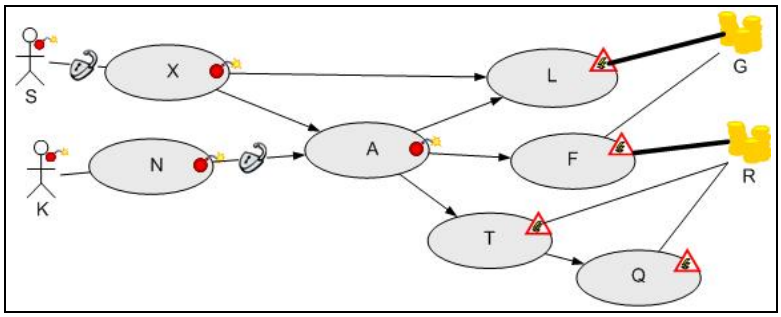


Diagram 2

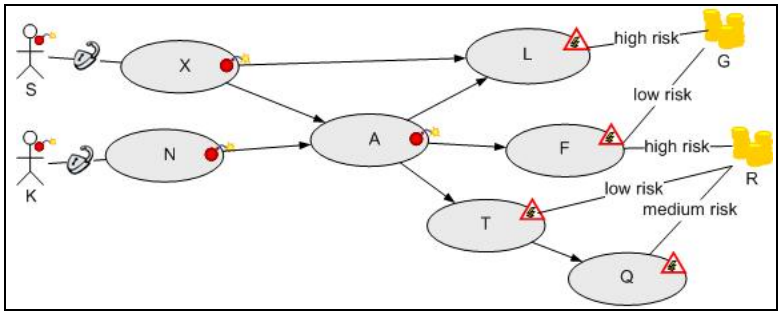


Diagram 3

TP14 - Which diagram shows best that unwanted incident of scenario F represents two different risks?

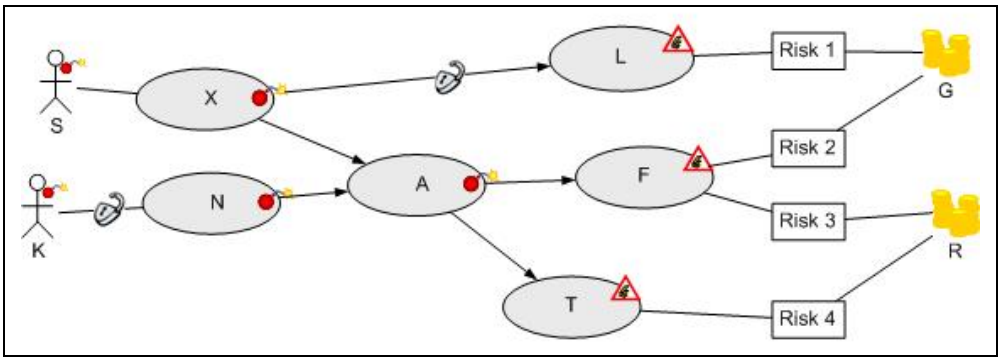


Diagram 1

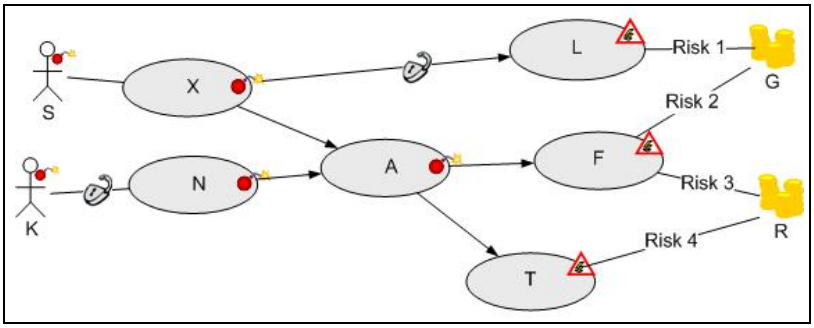


Diagram 2

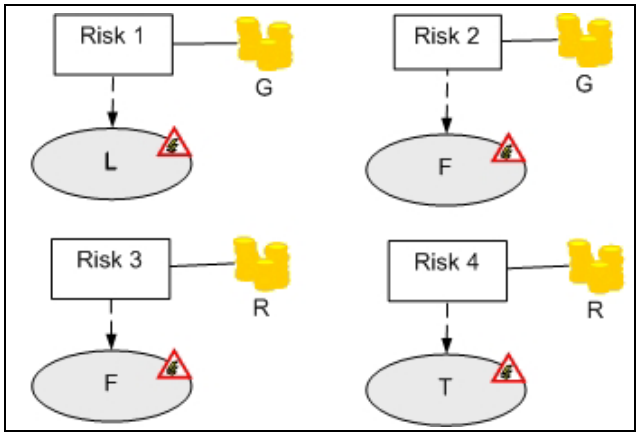


Diagram 3

C.III Prestudy results and discussion

The results from the statistical tests are presented in the tables below. The Chi-Square test is based on the assumption the there will be equally many observations in each category (here: the number of "0" and "1" will be the same in each task). In the tables the difference between observed and expected observations is denoted as "Residual". "df" denotes the degrees of freedom which is the number variables that can vary when the sum of the variables have to add up to the observed frequency. E.g. in our case, once the frequency for the "1's" is set, the frequency for the "0's" is given from the observed frequency minus the number of "1's", meaning that only the first variable is free and therefore the degrees of freedom equals 1. The outcome of the computation in a Pearsons Chi-Square test is the Chi-Square value (abbreviated to Chi-Square), which together with the degrees of freedom (df) gives us the asymptotic significance value from the Chi-Square

distribution. "Asymp. Sig." is short for asymptotic significance (or p-value) which indicates whether the difference between two diagrams is statistically significant (in our test it must be below 0,05 to be significant). As we can see there was a significant difference between the diagram alternatives in five of the seven tasks.

Table 5 – Results from SPSS

Conceptual containment				
		Frequencies		
TP8D1D2		*0*	*1*	*Total*
	Observed N	4	6	10
	Expected N	5	5	
	Residual	-1	1	
TP8D3D1		*0*	*1*	*Total*
	Observed N	8	2	10
	Expected N	5	5	
	Residual	3	-3	
TP8D2D3		*0*	*1*	*Total*
	Observed N	4	6	10
	Expected N	5	5	
	Residual	-1	1	
	Chi-Square	df	Asymp. Sig.	
TP8D1D2	0,4	1	0,527	
TP8D3D1	3,6	1	0,058	
TP8D2D3	0,4	1	0,527	

Likelihood				
		Frequencies		
TP10D1D2		*0*	*1*	*Total*
	Observed N	3	6	9
	Expected N	4,5	4,5	
	Residual	-1,5	1,5	
	Chi-Square	df	Asymp. Sig.	
TP10D1D2	1	1	0,317	

Shared vulnerabilities				
		Frequencies		
TP12D1D2		*0*	*1*	*Total*
	Observed N	1	7	8
	Expected N	4	4	
	Residual	-3	3	
	Chi-Square	df	Asymp. Sig.	

Magnitude of unwanted incident				
		Frequencies		
TP9D1D2		*0*	*1*	*Total*
	Observed N	6	1	7
	Expected N	3,5	3,5	
	Residual	2,5	-2,5	
TP9D3D1		*0*	*1*	*Total*
	Observed N	7	3	10
	Expected N	5	5	
	Residual	2	-2	
TP9D2D3		*0*	*1*	*Total*
	Observed N	0	9	9
	Expected N	4,5	4,5	
	Residual	-4,5	4,5	
	Chi-Square	df	Asymp. Sig.	
TP9D1D2	3,571	1	0,059	
TP9D3D1	1,6	1	0,206	
TP9D2D3	9	1	0,003	

Logical "and"				
		Frequencies		
TP11D1D2		*0*	*1*	*Total*
	Observed N	7	2	9
	Expected N	4,5	4,5	
	Residual	2,5	-2,5	
TP11D3D1		*0*	*1*	*Total*
	Observed N	1	7	8
	Expected N	4	4	
	Residual	-3	3	
TP11D2D3		*0*	*1*	*Total*
	Observed N	2	6	8
	Expected N	4	4	
	Residual	-2	2	
	Chi-Square	df	Asymp. Sig.	
TP11D1D2	2,778	1	0,096	
TP11D3D1	4,5	1	0,034	
TP11D2D3	2	1	0,157	

TP12D1D2	4,5	1	0,034	

Magnitude of risk				
		Frequencies		
TP13D1D2		*0*	*1*	*Total*
	Observed N	3	6	9
	Expected N	4,5	4,5	
	Residual	-1,5	1,5	
TP13D3D1		*0*	*1*	*Total*
	Observed N	9	1	10
	Expected N	5	5	
	Residual	4	-4	
TP13D2D3		*0*	*1*	*Total*
	Observed N	2	7	9
	Expected N	4,5	4,5	
	Residual	-2,5	2,5	

	Chi-Square	df	Asymp. Sig.	
TP13D1D2	1	1	0,317	
TP13D3D1	6,4	1	0,011	
TP13D2D3	2,778	1	0,096	

One incident, multiple risks				
		Frequencies		
TP14D1D2		*0*	*1*	*Total*
	Observed N	4	2	6
	Expected N	3	3	
	Residual	1	-1	
TP14D3D1		*0*	*1*	*Total*
	Observed N	0	10	10
	Expected N	5	5	
	Residual	-5	5	
TP14D2D3		*0*	*1*	*Total*
	Observed N	9	1	10
	Expected N	5	5	
	Residual	4	-4	

	Chi-Square	df	Asymp. Sig.	
TP14D1D2	0,667	1	0,414	
TP14D3D1	10	1	0,002	
TP14D2D3	6,4	1	0,011	

The gray cells in the table below indicate that statistical significant differences were found between the modeling alternatives.

Table 6 – Task conclusions

Task topic	***TaskID***	***Diagram***	***P***	***Hypothesis testing***
Conceptual containment	TP8	TP8D1D2	0,527	Keep H0
		TP8D3D1	0,058	
		TP8D2D3	0,527	
Magnitude of unwanted incident	TP9	TP9D1D2	0,059	Reject H0. D3 (and D1) are preferred over D2.
		TP9D3D1	0,206	
		TP9D2D3	0,003	
Likelihood	TP10	TP10D1D2	0,317	Keep H0
Logical “and”	TP11	TP11D1D2	0,096	Reject H0. D1 preferred over D3
		TP11D3D1	0,034	
		TP11D2D3	0,157	
Shared vulnerabilities	TP12	TP12D1D2	0,034	Reject H0. D2 is preferred
Magnitude of risk	TP13	TP13D1D2	0,317	Reject H0. D3 is preferred over D1, but not significantly over D2.
		TP13D3D1	0,011	
		TP13D2D3	0,096	
One incident, multiple risks	TP14	TP14D1D2	0,414	Reject H0. D1 and D2 are preferred over D3.
		TP14D3D1	0,002	
		TP14D2D3	0,011	

In TP8 none of the modeling alternatives were significantly preferred. The reason may be that they all have weaknesses that will make them difficult to use in practice. The first alternative will, although conforming to so-called best practice, become impossible to use when the diagram scales due to its space consuming design. The other two diagram alternatives are based on arrows to show inclusion, but still this aspect may be difficult to see and understand for the reader.

The third diagram alternative in TP9 was preferred over the two others, which shows that text labels are preferred over both size and color to illustrate magnitude.

Two alternatives for illustrating the likelihood of different paths were tested in TP10. Neither line thickness nor line type were preferred for this purpose.

Logical and-gates are often used in traditional risk modeling. In TP11 we tested three alternatives for illustrating this. We conclude that the one resembling the original and-gate symbol the most was preferred over the two other alternatives.

When several assets share the same vulnerability it may be useful to show it explicitly. In TP12 an alternative to using the original CORAS UML profile, in which vulnerabilities are placed below each asset, was compared to one that distributed vulnerabilities throughout the graph. The original CORAS UML profile version was preferred, probably because the students were more familiar with this notation and also because this notation is easier to read the than the other when one is interested in looking at the assets and their vulnerabilities only.

In TP13 we tested three alternative representations of risks with various magnitudes to find the preferred one. The third diagram alternative was preferred over the first, but not significantly over diagram 2. This means that for this task, a modeling alternative close to the CORAS UML profile was the least preferred modeling alternative. A similar result was found in TP14 where the notation close to the CORAS UML profile (diagram 3) was the least preferred among the models.

C.IV Threats to validity for the prestudy

In general, humans who get involved in a risk analysis are rarely familiar with the analysis concepts or the process used. Concepts that are intuitive to students are often also intuitive to professionals. We therefore believe that our use of students in this explorative investigation has little effect on the validity of the results.

The student received the concept definitions belonging to the original CORAS UML profile [33]. However, many of the modeling alternatives used a new type of notation. This might have been confusing since the UML profile notation was more familiar. Nevertheless, it was often the new modeling notation was preferred over the original notation, something that supports our assumption that the CORAS UML profile may be improved.

The number of subjects was limited (10-11 respondents) and we therefore draw no strong conclusions from this prestudy. Nevertheless, the findings will be used as supportive evidence when interpreting the results from the next investigation where professionals will be used as subjects.

C.V Conclusion

Although the prestudy results are based on a limited number of subjects, the results are used as an exploratory study. The findings form the basis both for the material that will be tested in the main study, as well as for a control of the study set-up.

The main findings from this prestudy were:

- The original CORAS UML notation is not always preferred (particularly for the space consuming representations).
- Text labels seem to be preferred over more graphical means.

- If a symbol for a concept already exists in a related modeling notation, one should investigate this alternative before developing other modeling alternatives (i.e. for the and-gate, and consequently also the or-gate).

Appendix H

Quality evaluation of the CORAS language

This appendix provides the evaluation of the CORAS language. In the following we first model the core modeling scenarios described in *Sect. 6.1*, and then we evaluate the language according to the quality requirements in *Sect. 6.2*. The last sub section provides the summary of the evaluation which is used in *Chapter 9*. The structure of this appendix is as follows:

H.I MODELING THE CORE SECURITY RISK SCENARIOS WITH THE CORAS LANGUAGE . 2

H.I.1 PHASE 1: ESTABLISHING THE CONTEXT 2
H.I.1.1 Evaluating the Modeling Effort *4*
H.I.1.2 Applying the Structured Semantics *4*
H.I.2 PHASE 2: IDENTIFYING RISKS 5
H.I.1.3 Evaluating the Modeling Effort *6*
H.I.1.4 Applying the Structured Semantics *7*
H.I.3 PHASE 3: ESTIMATING RISKS 7
H.I.1.5 Evaluating the Modeling Effort *9*
H.I.1.6 Applying the Structured Semantics *9*
H.I.4 PHASE 4: EVALUATING RISKS 10
H.I.1.7 Evaluating the Modeling Effort *14*
H.I.1.8 Applying the Structured Semantics *15*
H.I.5 PHASE 5: IDENTIFYING TREATMENTS 15
H.I.1.9 Evaluating the Modeling Effort *18*
H.I.1.10 Applying the Structured Semantics *19*

H.II THE QUALITY EVALUATION 20

H.II.1 DOMAIN APPROPRIATENESS 20
H.II.2 PARTICIPANT LANGUAGE KNOWLEDGE APPROPRIATENESS 23
H.II.3 KNOWLEDGE EXTERNALIZABILITY APPROPRIATENESS 23
H.II.4 COMPREHENSIBILITY APPROPRIATENESS 24
H.II.5 TECHNICAL ACTOR APPROPRIATENESS 27
H.II.6 ORGANIZATIONAL APPROPRIATENESS 27

H.III CONCLUSION 29

H.III.1 MODELING THE CORE SECURITY RISK SCENARIOS 29
H.III.2 LANGUAGE QUALITY 30
H.III.3 REQUIREMENTS THAT WERE OMITTED FROM THE EVALUATION 31

REFERENCES 32

H.I Modeling the Core Security Risk Scenarios with the CORAS Language

The core security risk scenarios are an example of security risk analysis information gathered during the analysis process' five phases: context identification, risk identification, risk estimation, risk evaluation and treatment identification.

All the details from the core security risk scenarios will not be repeated in the models, for this we refer to the textual description in Sect. 6.1.

H.I.1 Phase 1: Establishing the Context

The assets from Phase 1 in the core security risk scenarios are modeled in an asset diagram that shows how the assets relate. The two assets within the frame represent direct assets since they may be harmed directly by unwanted incidents related to the web portal service. The arrow from Availability of service to Company reputation means that damage to the availability of the service may damage the company reputation too. Indirect assets are assets that only can be affected if another asset is harmed first, and they are modeled outside the frame.

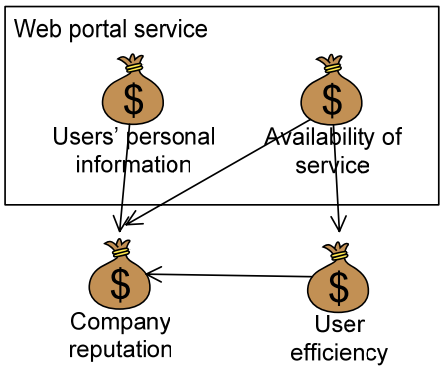


Figure 1 – Asset diagram (direct and indirect assets)

The value definitions used to measure asset damage, likelihood, consequences, risks values etc. are not modeled graphically in the CORAS language. Each asset should be measured according to the type of damage they may be exposed to, but since this is underspecified in the core security risk scenarios we only rank them according to their importance to the stakeholders. An example of how this may be presented is given in Table 1.

Table 1 – Value definitions in a table

	Measurement scale
Asset	Values 1-4, where 1 = highest importance and 4 = lowest importance
Likelihood	Seldom =1 time per 5 years or less Sometimes = more than 1 time per 5 years, and less than 1 time per year Often = 1 time per year or more
Consequence	Minor, moderate, major
Risk value	Low, medium, high
Risk reduction	Measured as decreased risk value (e.g. from high to medium risk)

The reason for not modeling this graphically is that this information is more like a list of definitions which is part of establishing the scope of the analysis, but in it self is very simple to understand. Graphical model should be used to explain complex relationships etc, and not necessarily for every bit of information that arise during a security analysis.

The risk acceptance levels of the company management and the central authorities for each asset are shown in the diagram below. This diagram is a special kind of an asset diagram that illustrates risk acceptance levels with respect to both the assets and the stakeholders (often referred to as clients and other interested parties).

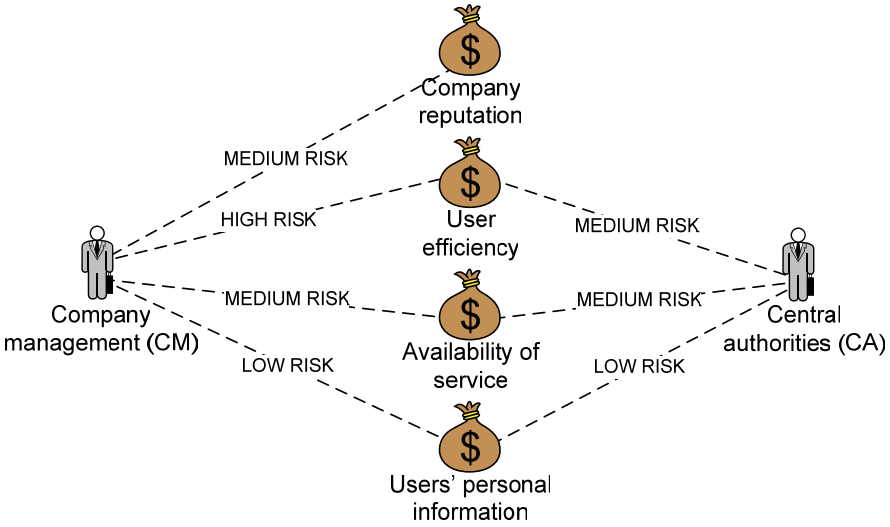


Figure 2 – Risk acceptance level diagram showing maximum accepted risk for each asset

The information in Figure 2 is equivalent to the information in Table 5 – Risk evaluation criteria in *Chapter 6*.

Combining the information in Figure 1 with Figure 2 gives the asset diagram in Figure 3. As we see, the difference between the two interested parties in acceptable risk levels for asset "User efficiency" is modeled by splitting the asset into two representations with identifiers that refer to their stakeholder (CM User efficiency belongs to Company Management and CA User efficiency belongs to Central Authorities). To illustrate that only Company management considers the company's reputations as an asset, also this asset has been given an identifier which reflects the stakeholder.

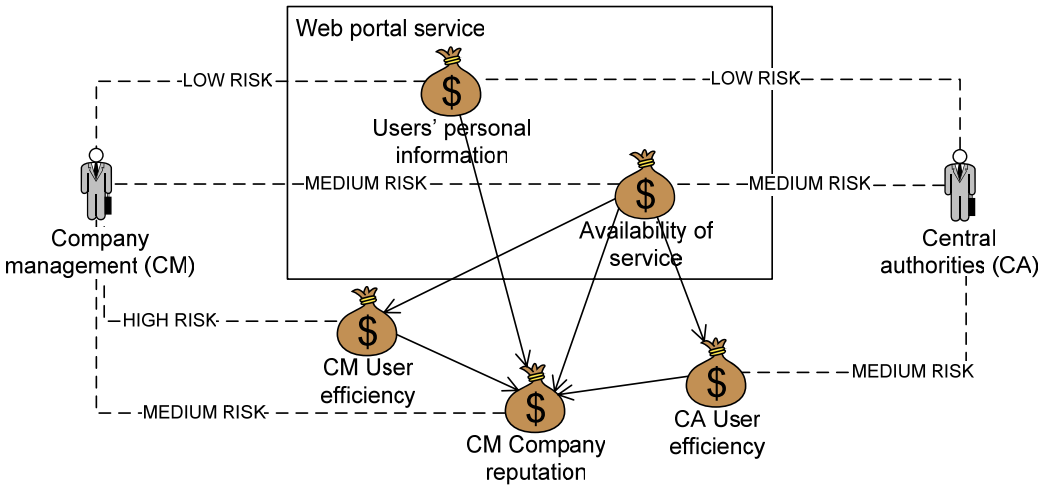


Figure 3 – Refined asset diagram including risk acceptance levels

H.I.1.1 Evaluating the Modeling Effort

The CORAS language was capable of modeling:

- The assets, including dependencies between them
- The stakeholders
- The indirect and direct assets
- The risk acceptance levels of the stakeholder for each asset
- The two stakeholders' conflicting risk acceptance level for the same asset.

Information that was not modeled:

- The full description of the assets
- The full description of the stakeholders
- The scales of the assets, likelihood and consequence values

When the stakeholders accept different risk values for an asset, the asset is split into two different assets with reference to their stakeholder. If only one of the stakeholders value an asset, this is reflected by a specific reference to the stakeholder in the asset's name. We have chosen not to include more than the name of the assets and stakeholders in the models because any extra information will clutter the diagram. The diagrams should be accompanied with textual explanations, which is a more appropriate place to give detailed descriptions of the elements in the diagram. Anything that are not directly necessary in order to read and understand the diagram should be avoided. The decision of not modeling the value scales was taken from a usability perspective. We believe this kind of information is more suitable for presentation in a table than in a graphical model because it only consists of static definitions.

H.I.1.2 Applying the Structured Semantics

To illustrate the use of the structured semantics in *Appendix D* we translate one of the relations in the asset diagram:

Translating the vertices:

- Company management (CM) is a stakeholder.
- Availability of service is an asset.

Translating the relation:

- Company management (CM) wants to protect the value of availability of service, but accepts risk value medium risk or less.

This corresponds to information found in the core security risk scenarios in phase 1 (Sect. 6.1.1.1).

H.I.2 Phase 2: Identifying Risks

The unwanted incidents, threats and vulnerabilities are modeled in threat diagrams. These show the paths from the threats to the left, via various vulnerabilities and threat scenarios which end in unwanted incidents that harms one or more assets. There are three threat diagrams, one for each type of threat.

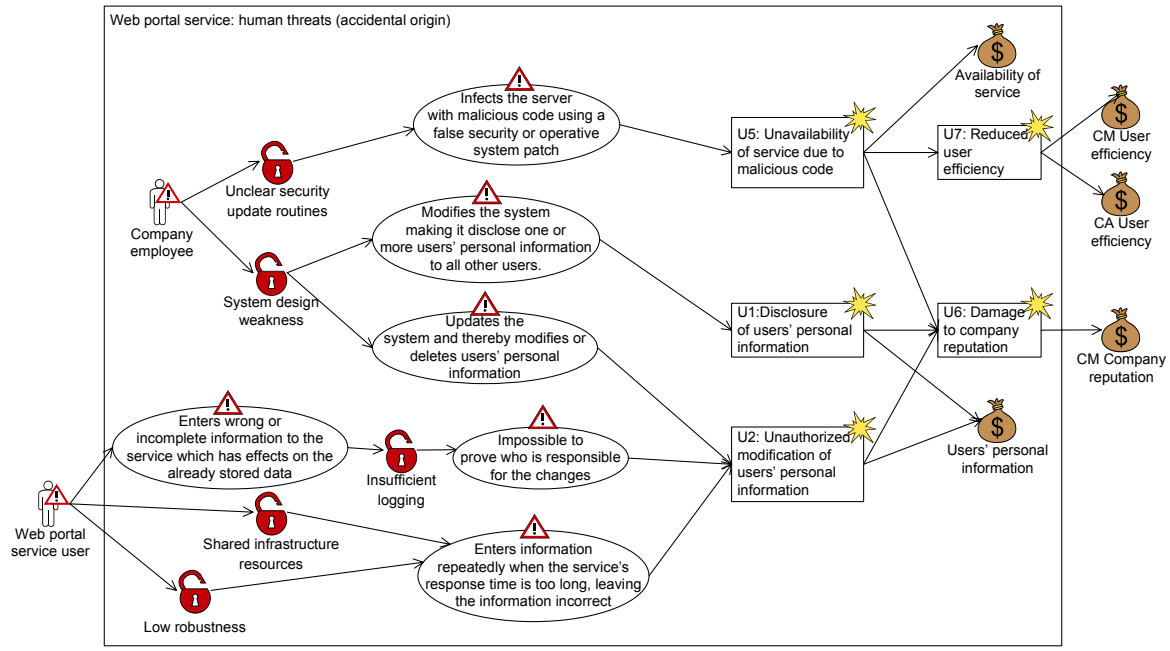


Figure 4 – Threat diagram: human, accidental

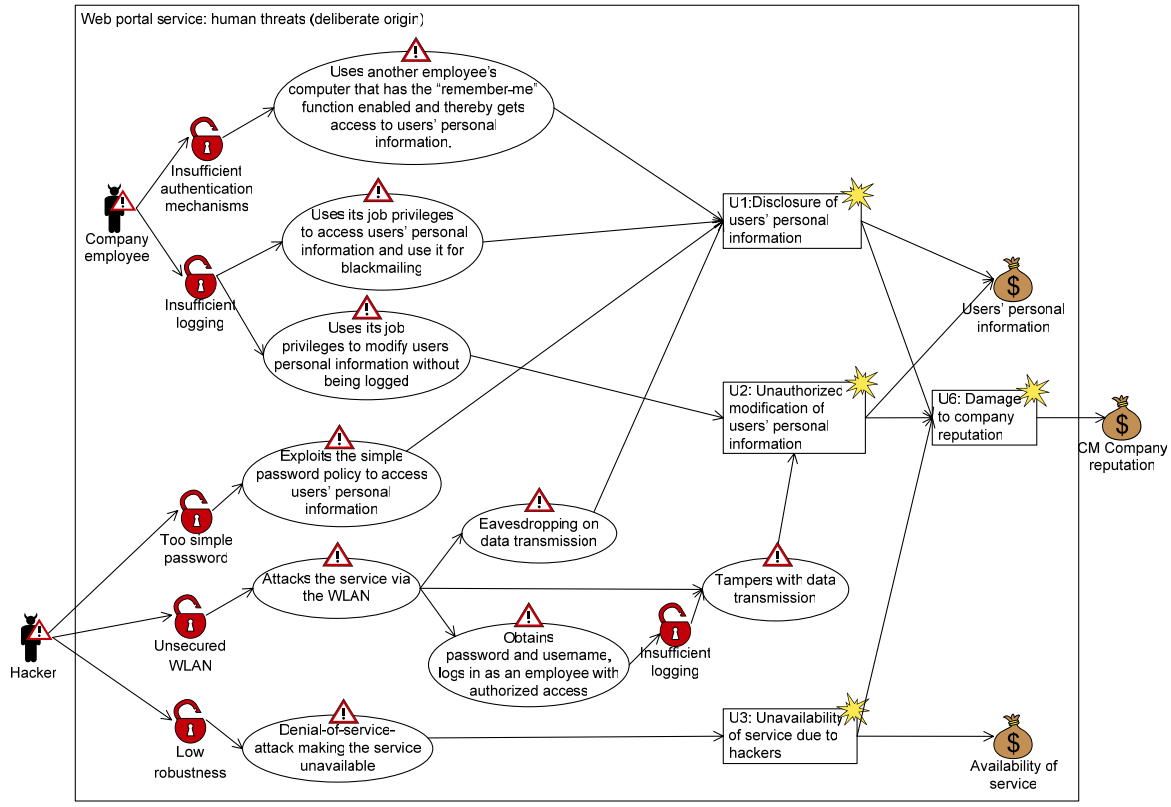


Figure 5 – Threat diagram: human, deliberate

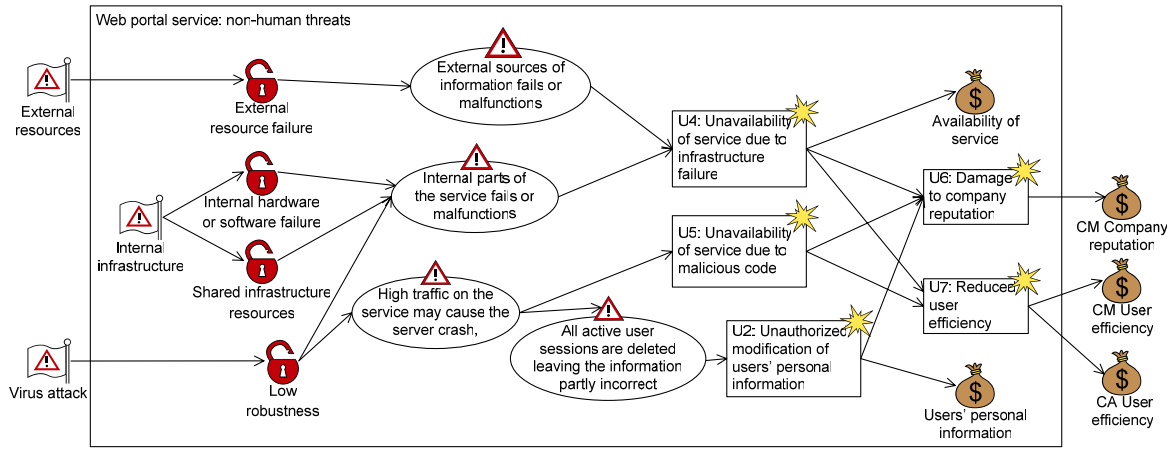


Figure 6 – Threat diagram: non-human threats

H.I.1.3 Evaluating the Modeling Effort

The CORAS language was able to model:

- The threats, divided into human (accidental and deliberate) and non-human threats
- The vulnerabilities, placed in a logical sequence of events with association to the threats that may exploit them
- The threat scenarios with associations to relevant threats, vulnerabilities and unwanted incidents
- The unwanted incidents with associations to the assets they may affect
- A logical region symbolizing the target for each type of threat

This information was not modeled:

- The full description of the vulnerabilities
- The full description of threats
- The full description of assets

As for the asset diagram, we decided to include only the names of the threats, vulnerabilities and assets to avoid messy and confusing diagrams. Any additional information about the elements should be stated in the accompanying textual description, which typically is required for the final analysis report.

H.I.1.4 Applying the Structured Semantics

To illustrate how the structured semantics provides a translation that conforms to the information in the core security risk scenarios we translate one path in the diagram above (from a threat via threat scenarios, vulnerabilities and unwanted incidents to an asset).

Translating the vertices:

- External resources is a non-human threat.
- Threat scenario external sources of information fail or malfunction occurs with undefined likelihood.
- Unwanted incident unavailability of service due to infrastructure failure occurs with undefined likelihood.
- Availability of service is an asset.

Translating the relations:

- External resources exploit external resource failure to initiate external sources of information fail or malfunction with undefined likelihood.
- External sources of information fail or malfunction leads to unavailability of service due to infrastructure failure with undefined likelihood.
- Unavailability of service due to infrastructure failure impacts availability of service with undefined consequence.

This corresponds to information from the core security risk scenarios in phase 2 (Sect. 6.1.2.1).

H.I.3 Phase 3: Estimating Risks

The threat diagram from the previous phase forms the basis for risk estimation. Information about the unwanted incidents' likelihood and consequences towards assets are added to the unwanted incident symbol and the relations between these and assets. The separation between the different kinds of threats is kept.

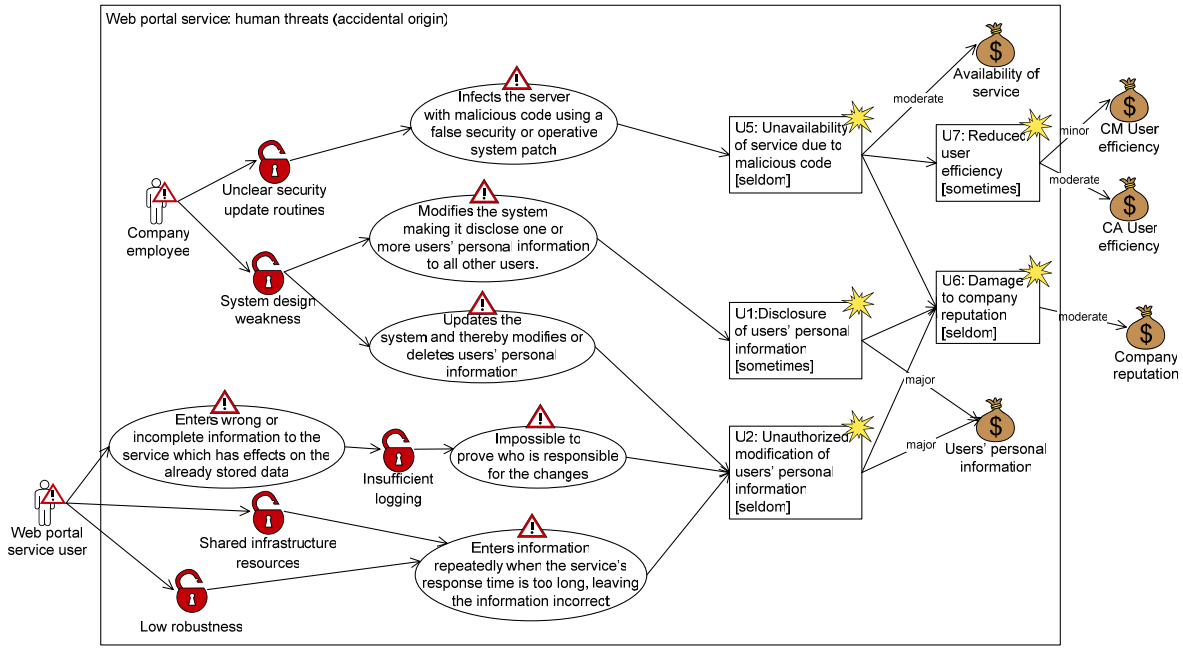


Figure 7 – Threat diagram for human threats (accidental) with likelihood and consequence estimates

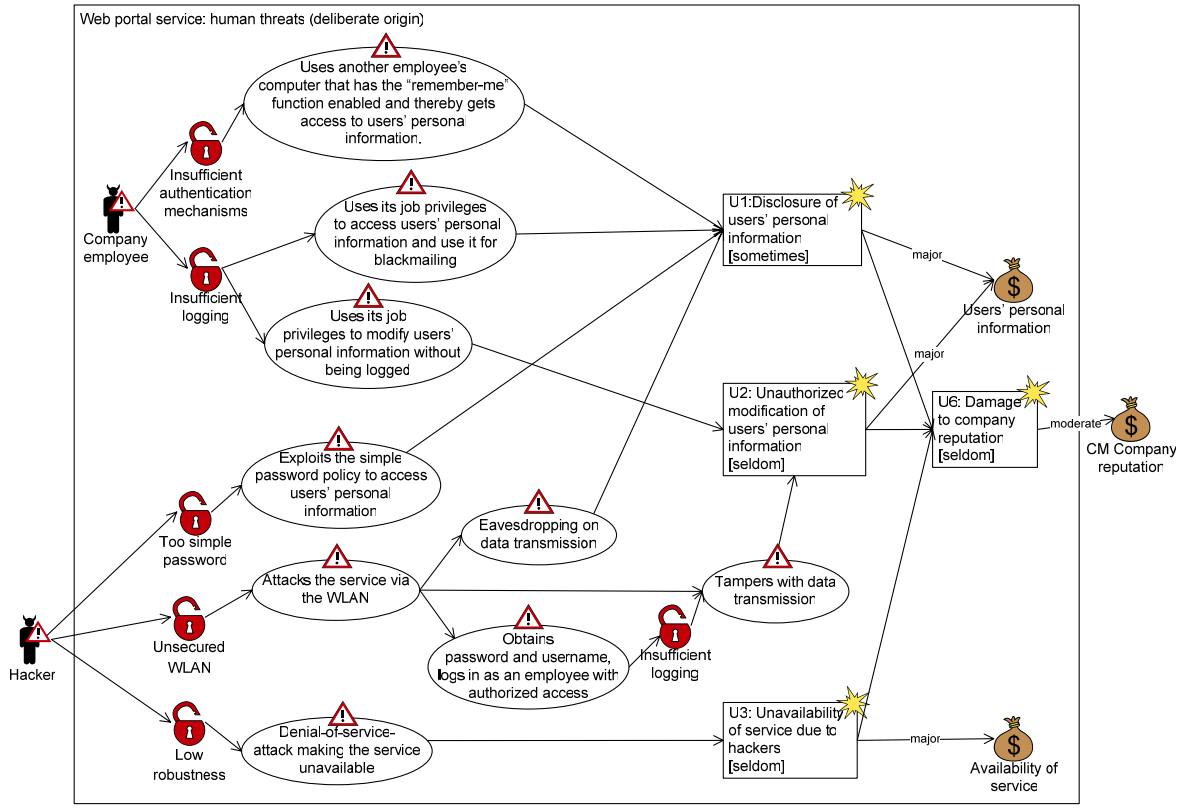


Figure 8 – Threat diagram for human threats (deliberate) with likelihood and consequence estimates

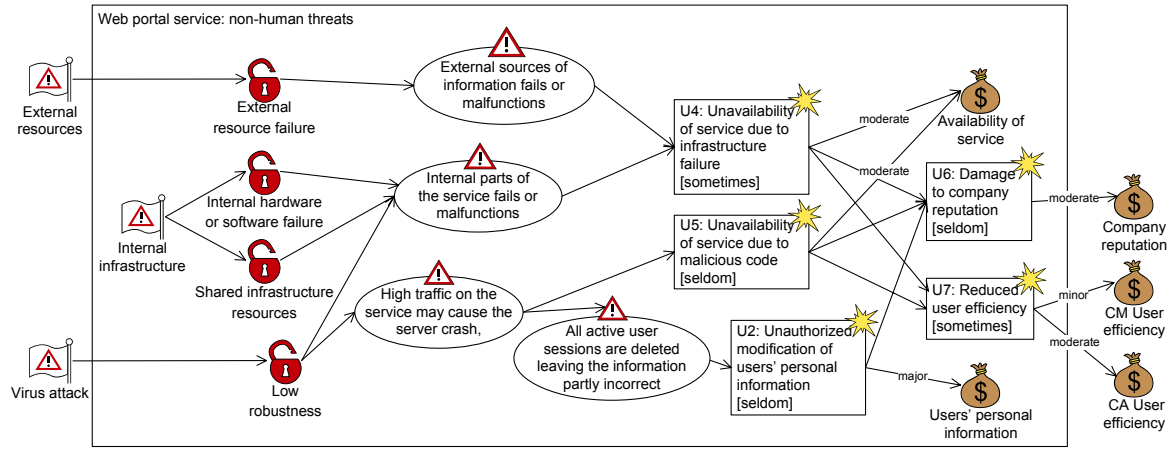


Figure 9 – Threat diagram for non-human threats with likelihood and consequence estimates

H.I.1.5 Evaluating the Modeling Effort

The CORAS language was able to model:

- The consequence of each unwanted incident towards the assets
- The likelihood of each unwanted incident

This information was not modeled:

- None

Modeling the information from this phase using the CORAS language required an extremely small effort. In practice the modeling consisted of adding textual information about likelihood and consequences to the threat diagrams made in the previous phase. This makes the notation very suitable for modeling "on-the-fly" during risk estimation.

H.I.1.6 Applying the Structured Semantics

Except for the likelihood of the unwanted incident and its consequence, we defined the structured semantics for the upper path in the diagram above in the previous phase. We only translate the new information:

Translating the vertices:

- Unwanted incident unavailability of service due to infrastructure failure occurs with likelihood sometimes.

Translating the relation:

- Unavailability of service due to infrastructure failure impacts availability of service with moderate consequence.

This corresponds to information from the core security risk scenarios in phase 3 (risks R4CM, R4CA in Sect. 6.1.3.1).

H.I.4 Phase 4: Evaluating Risks

In the risk evaluation, the each risk's consequence and likelihood values are combined into a risk value. The information about which risks exists, who or what causes them, and what do they harm is modeled in risk diagrams (Figure 10, Figure 11 and Figure 12).

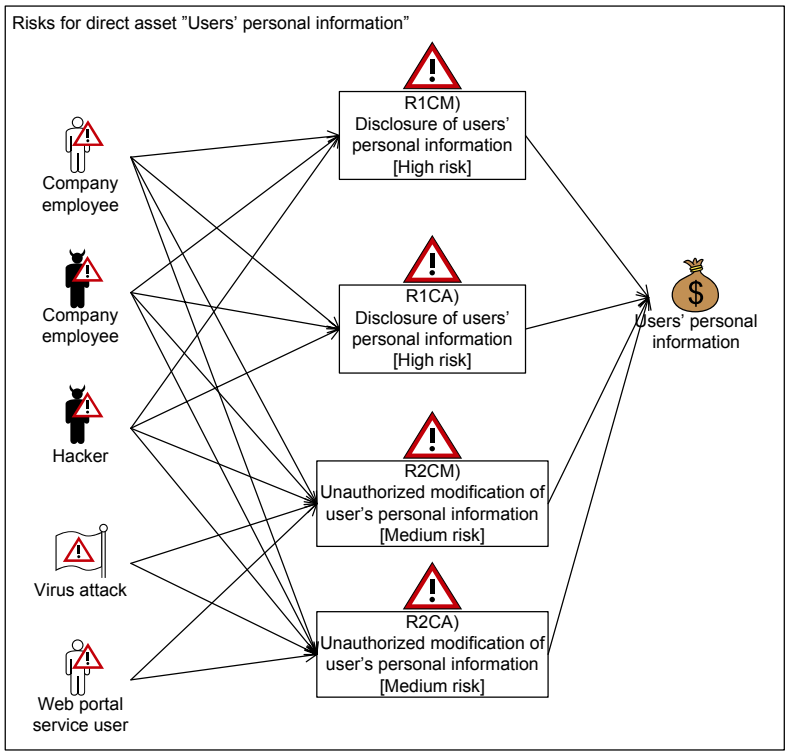


Figure 10 – Risk diagram for the asset "Users' personal information"

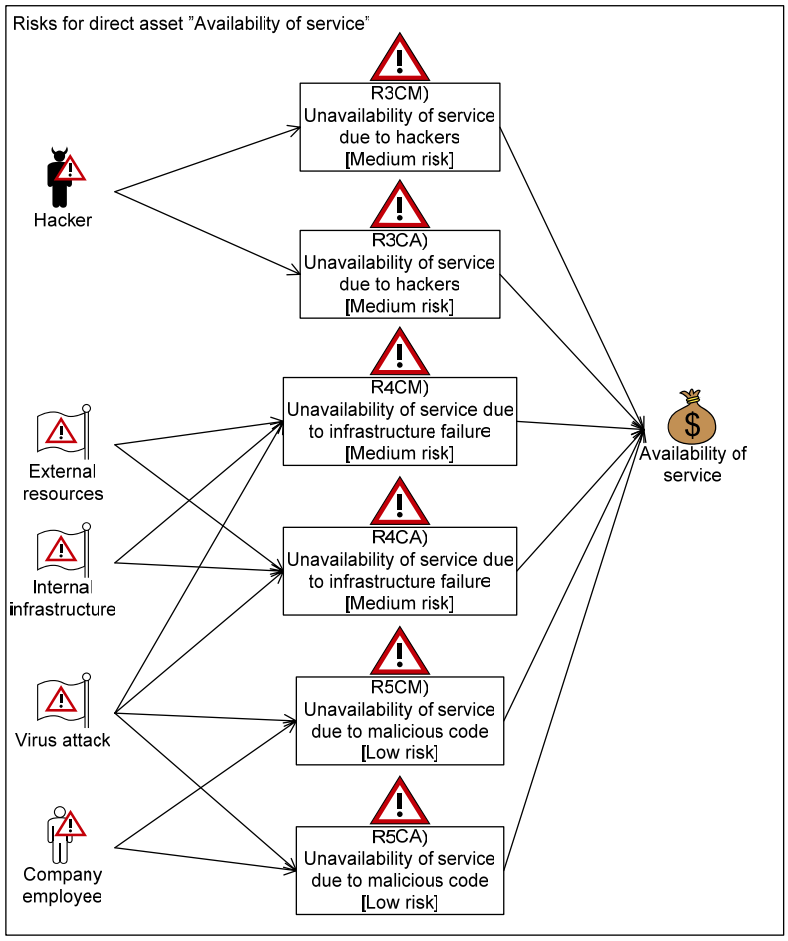


Figure 11 – Risk diagram for the asset "Availability of service"

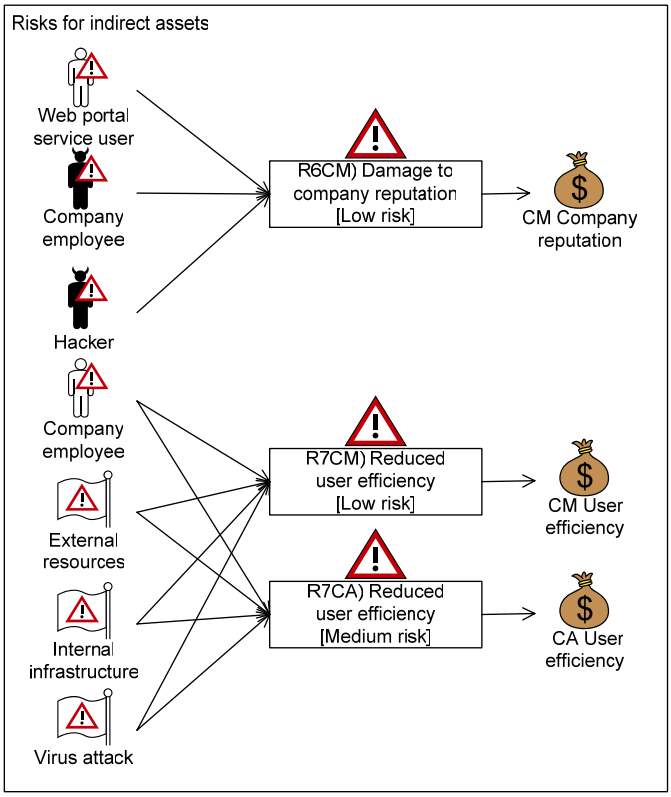


Figure 12 – Risk diagram for the indirect assets

The risk value of each risk is then compared to the risk acceptance levels decided in the context establishment (Figure 3). The risk evaluation diagrams are used to illustrate which risks are acceptable and which are not. Acceptable risks are marked with a grey color and will not be transferred to the treatment identification phase.

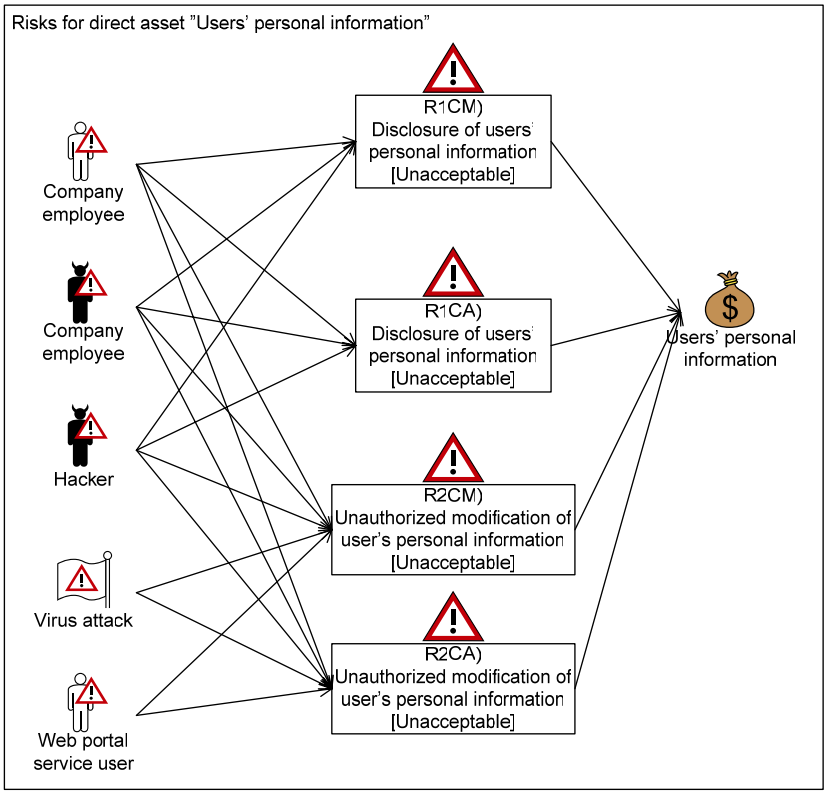


Figure 13 – Risk evaluation of risks towards asset Users' personal information

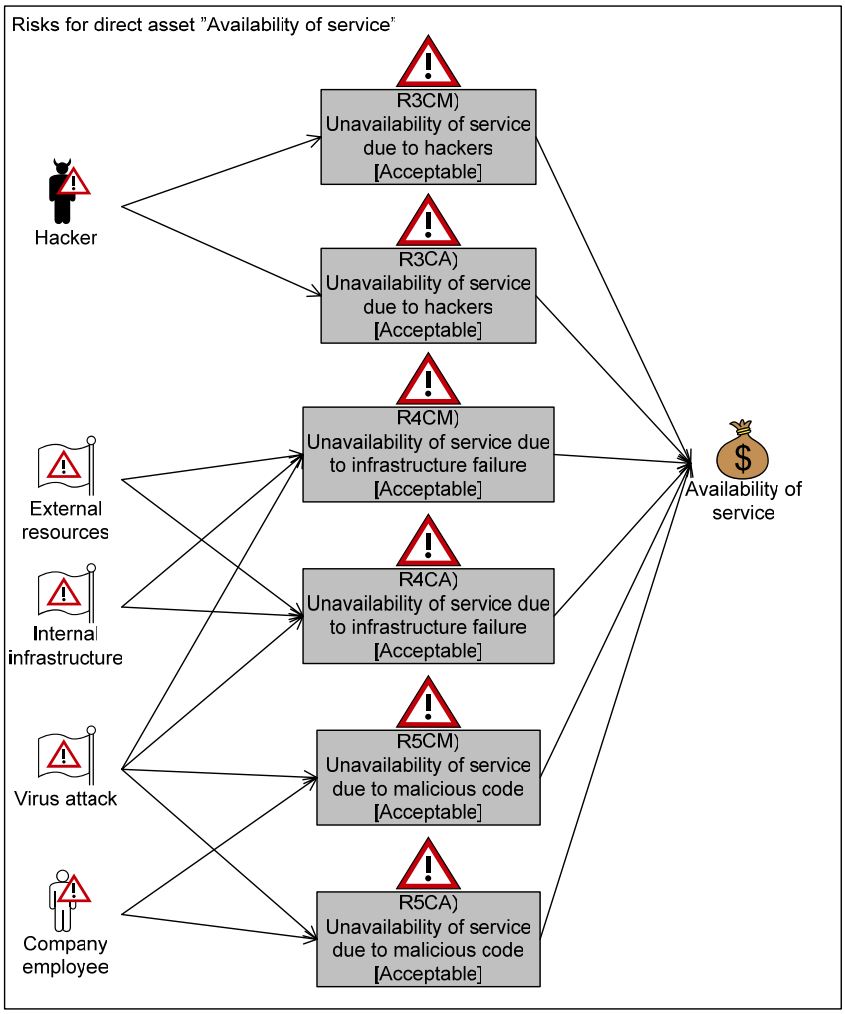


Figure 14 – Risk evaluation for risks towards asset Availability of service

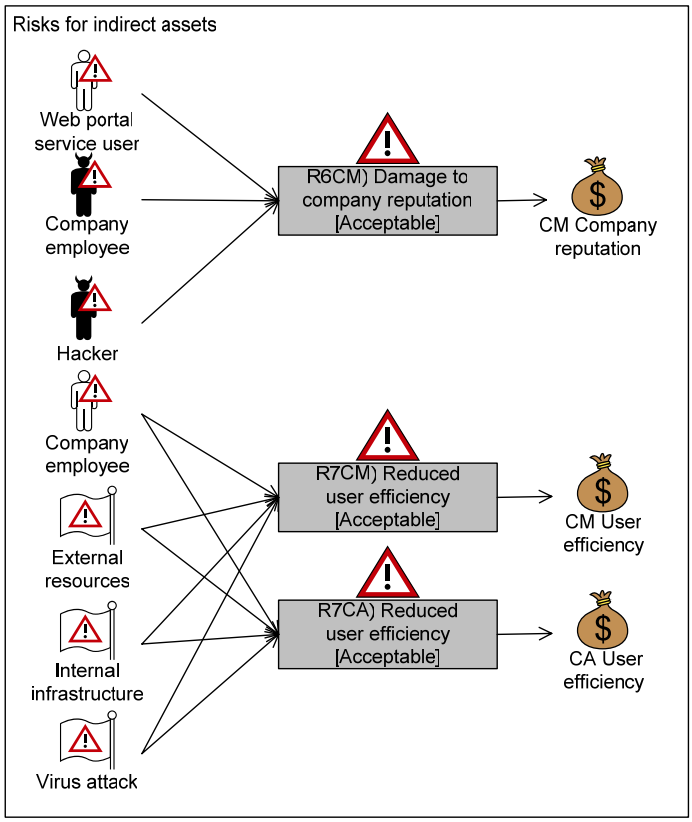


Figure 15 – Risk evaluation for risks towards the indirect assets

H.I.1.7 Evaluating the Modeling Effort

The CORAS language was able to express:

- The risks including their risk value
- A separation of acceptable and unacceptable risks according to the risk acceptance levels of each stakeholder

This information was not modeled:

- None

The risk evaluation phase basically takes the information gathered in the risk identification and estimation and compares it to the risk acceptance levels in phase 1. The CORAS language had no problems modeling this information. The language provides a set of overview diagrams that can be very useful when presenting the analysis findings. The risk diagrams are created by "hiding" the vulnerabilities and threat scenarios between a threat and an unwanted incident in the threat diagrams. The unwanted incident symbol is replaced with a risk symbol with the appropriate risk value. Depending on the modelers choice of tools this can be done either manually (like in Microsoft Visio) or semi-automatically in the CORAS tool. The risk evaluation diagram is a variation of the risk diagram where acceptable risks are grayed out. The latter diagram type is not necessary in order to model the core security risk scenarios, but provides an alternative to the traditional risk matrix.

H.I.1.8 Applying the Structured Semantics

To illustrate the translation of the risk diagrams we translate the upper path in the risk diagram in Figure 10.

Translating the vertices:

- Company employee is an accidental threat.
- The risk disclosure of users' personal information occurs with risk value [high].
- Users' personal information is an asset.

Translating the relations:

- Company employee initiates disclosure of users' personal information.
- Disclosure of users' personal information impacts users' personal information.

By restructuring the sentences above, we get a paragraph that summarizes the translated path: "there is a high risk for unintentional disclosure of the personal information of the users, caused by an employee". The paragraph corresponds to the information found in 6.1.4.1, but also includes information about the threat from 6.1.2.1.

H.I.5 Phase 5: Identifying Treatments

The treatment diagram takes its starting point in the threat diagrams and replaces the consequence between the unwanted incident symbol and the asset with the appropriate risk ID from the risk diagrams. Only unacceptable risks are showed. During the brainstorming session the participants come up with treatment suggestions which are modeled in the diagram like shown in the figures below.

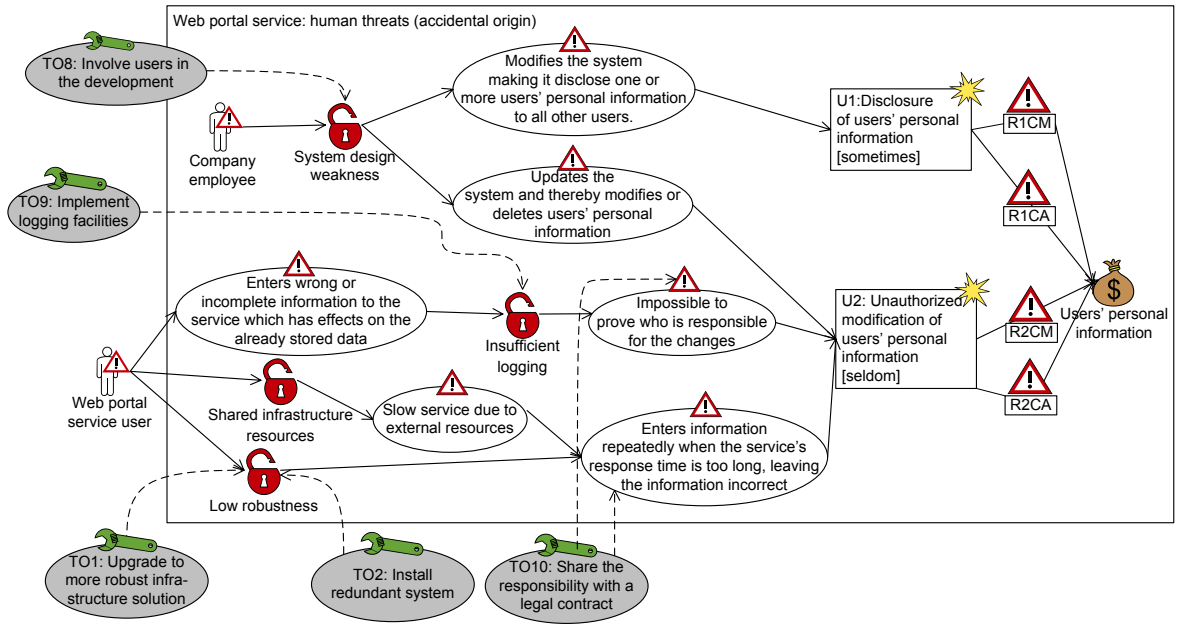


Figure 16 – Treatment diagram for R1CM, R1CA, R2CM, R2CA (human, accidental threats)

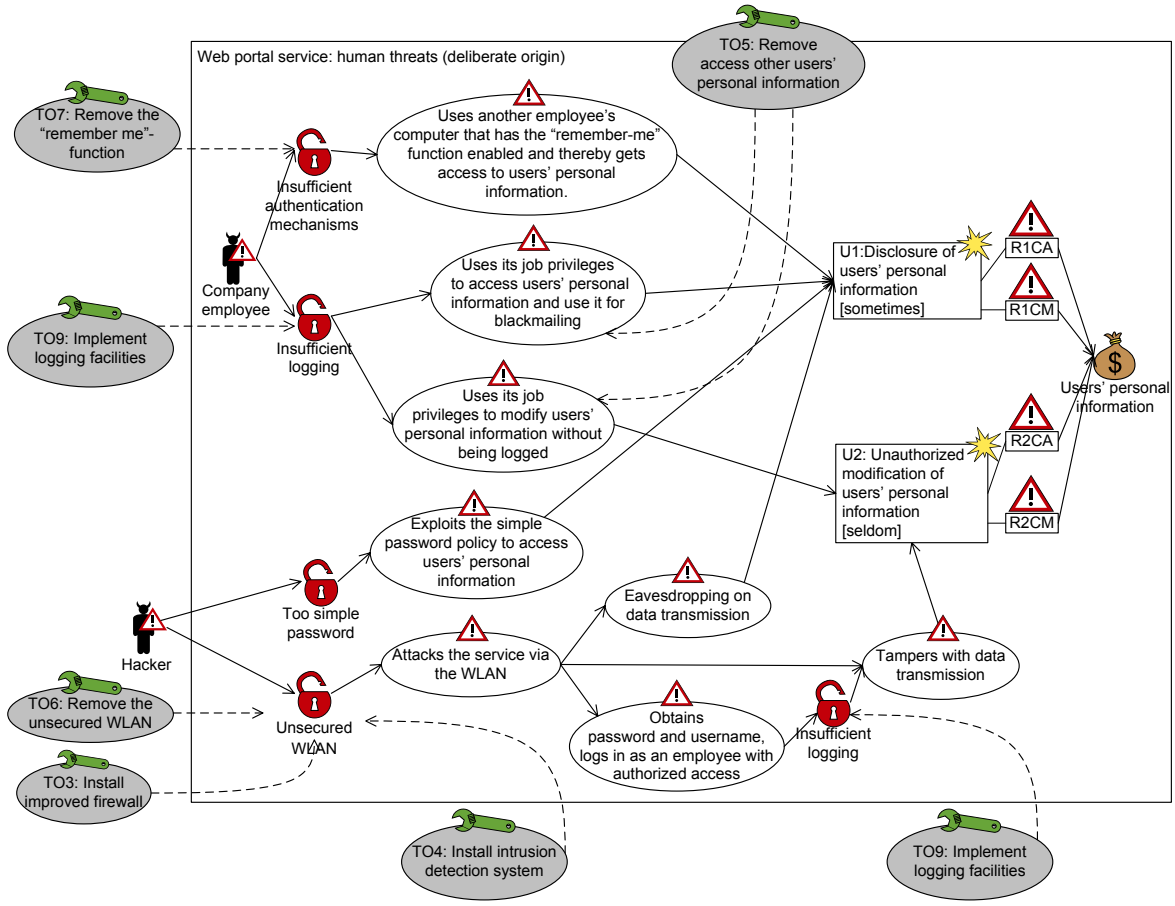


Figure 17 – Treatment diagram for R1CM, R1CA, R2CM, R2CA (human, deliberate threats)

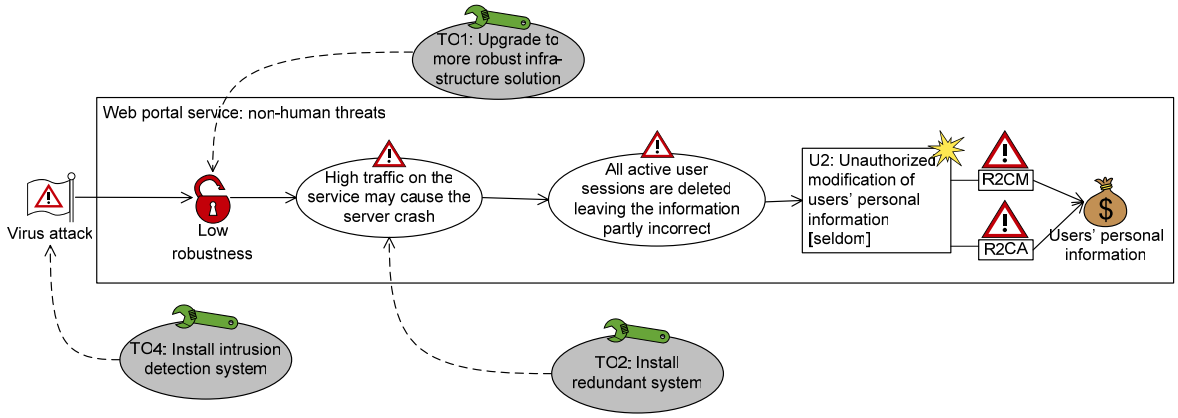


Figure 18 – Treatment diagram for R2CM, R2CA (non-human threat)

To get an overview of the treatments and which risks they are expected to defend the target against, the CORAS language has a treatment overview diagram:

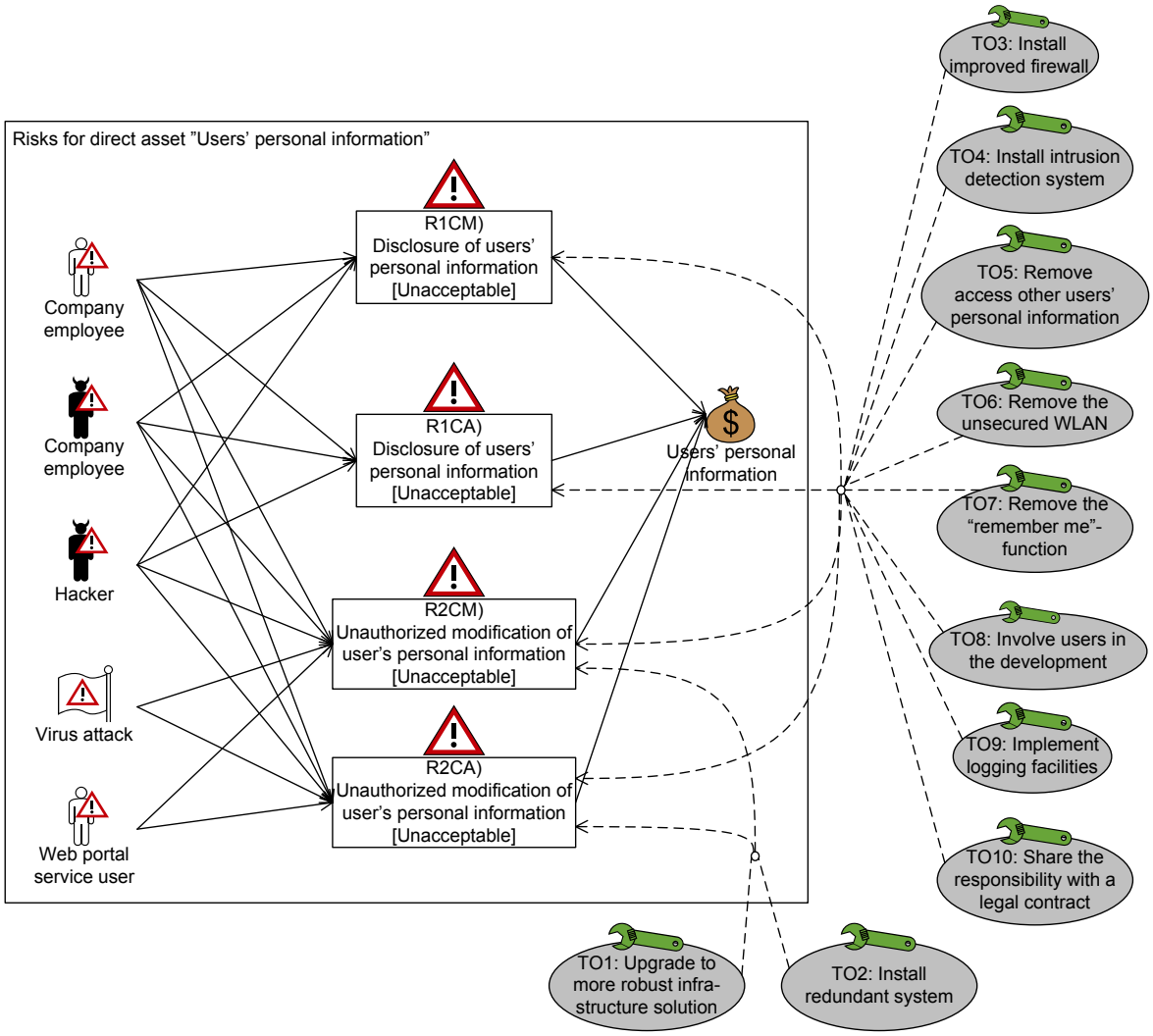


Figure 19 – Treatment overview diagram

The treatments and their expected effects on the consequences and likelihoods are listed in the table below. The CORAS language has no particular diagram for illustrating this. Depending on the combination of treatments, the resulting risk values may vary. In the table below each treatment's effect towards R1CM, R1CA, R2CM, R2CA is specified. It is possible to model each treatment's effect, but we only provide an example in Figure 20 where the effect of treatment TO6 is shown.

Table 2 – Treatment effects and resulting risk values

	R1CM & R1CA		**R2CM & R2CA**	
Treatment	**Treatment effect**	**Resulting risk value**	**Treatment effect**	**Resulting risk value**
TO1	-	-	Reduce likelihood	medium → low
TO2	-	-	Reduce consequence	medium → low
TO3	Reduce likelihood	high → high	Reduce likelihood	medium → medium
TO4	Reduce consequence	high → high	Reduce consequence Reduce likelihood	medium → medium
TO5	Reduce likelihood	high → high	Reduce likelihood	medium → medium
TO6	Reduce likelihood	high → low	Reduce likelihood	medium → low
TO7	Reduce likelihood	high → high	Reduce consequence	medium → medium
TO8	Reduce likelihood	high → medium	Reduce likelihood	medium → medium

TO9	Reduce likelihood	high → medium	Reduce consequence Reduce likelihood	medium → low
TO10	-	-	Reduce consequence	medium → medium

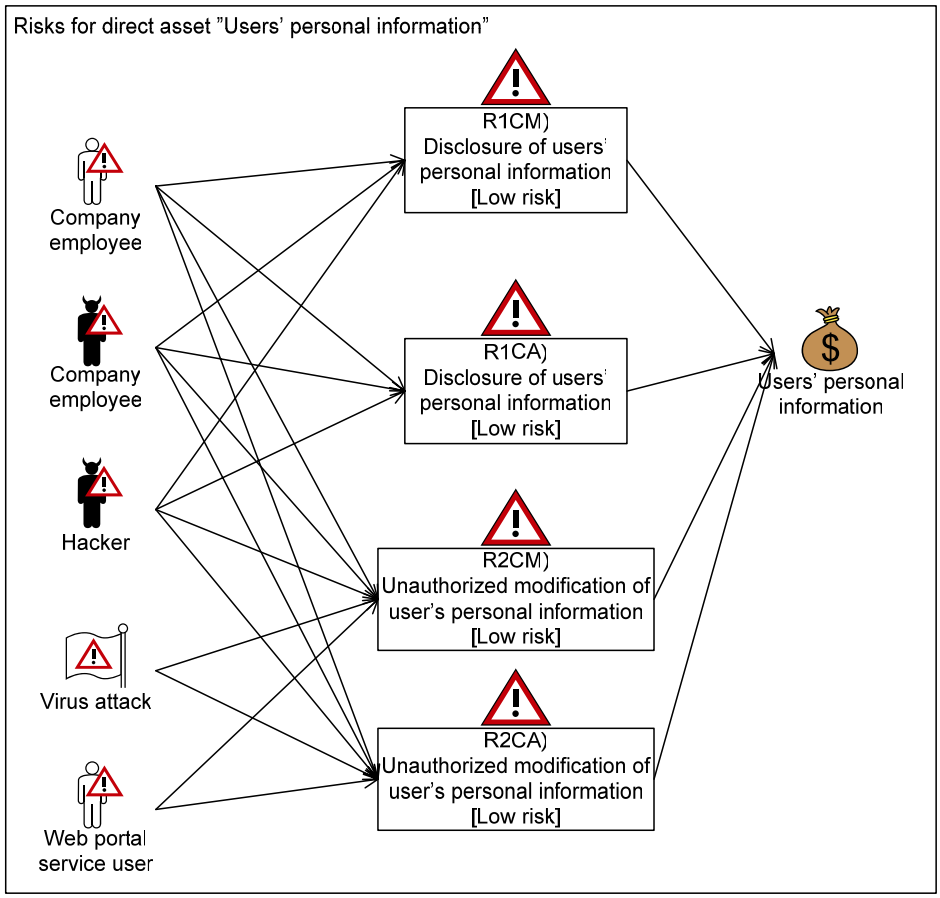


Figure 20 – The resulting risk values from applying TO6

H.I.1.9 Evaluating the Modeling Effort

The CORAS language was able to express:

- The treatments, pointing towards the element in the threat diagram where they should be applied
- An overview of the treatments that may be applied for each risk
- An example of the risk diagram after applying a specific treatment

This information was not modeled:

- The treatments' effect on the unacceptable risks
- The estimated change in risk value each treatment may have on each of the unacceptable risks

The decision of not modeling treatment effects in the diagrams is based on experienced with the UML profile where this was a part of the notation. The treatment effect labels had a tendency of making the diagrams quite unreadable since they often conflicted with existing elements and arrows. The CORAS language has the option for adding this kind of information to the treatment arrows, but we do not recommend using it unless for very simple diagrams.

The treatment overview diagram is an extra diagram type, suitable for presentation purposes in presentations, reports etc., but not required in order to model the core security risk scenarios. The CORAS language has no customized diagram for modeling the expected treatment effects on the risk values, but this may be illustrated using a risk diagram. In our opinion, it not necessary to spend time on modeling this information since it is more efficient to present it in a table.

H.I.1.10 Applying the Structured Semantics

The upper path in the treatment diagram in Figure 16 is translated using the structured semantics:

Translating the vertices:

- Company employee is an accidental threat.
- System design weakness is a vulnerability.
- Modifies the systems making it disclose one or more users' personal information to all other users is a threat scenario.
- Disclosure of users' personal information is an unwanted incident.
- Users' personal information is an asset.
- Involve users in the development is a treatment (treatment scenario).

Translating the relation:

- Company employee exploits system design weakness to initiate modifies the systems making it disclose one or more users' personal information to all other users with undefined likelihood.
- Modifies the systems making it disclose one or more users' personal information to all other users leads to disclosure of users' personal information with likelihood sometimes.
- Disclosure of users' personal information impacts users' personal information.
- Involve users in the development treats system design weakness.

By restructuring the sentences above, the following paragraph is the result of the translation: "in order to reduce the risk of an employee making a mistake that leads to disclosure of user data, one should involve the users (i.e. employees with permission to modify the system) in the further development of the system to mitigate any design weaknesses".

Since the treatment diagram builds on a threat diagram, the paragraph above corresponds to information found all the previous phases, including treatment information from 6.1.5.1.

H.II The Quality Evaluation

The requirements from *Chapter 6* based on are evaluated in the following tables, indicating whether it is a must or should requirement and the score is assigned according to the scale above. Each evaluation is also documented with a rationale.

The scores will help evaluating each section of requirements and give indications as to where one has what are the weaknesses and strengths of the CORAS language. Since the requirement vary level of detail and the "should/must" category does not reflect this (i.e. a small, detailed requirement may be less important compared to a high-level requirement), we will only use the average score numbers from the evaluation as indicators of the language's weak and strong parts.

Within each of the appropriateness factor categories, we do not find all the requirements equally important and the evaluation reflects this by categorizing the two types of requirements as either a "must have" or "should have", where a "must have" is more important than a "should have".

This is the grade scale that will be used in the evaluation:
0 - None or little support for the requirement
1 - There is some, though insufficiently support for the requirement
2 - The requirement is sufficiently supported
3 - The requirement is well supported

H.II.1 Domain Appropriateness

The domain appropriateness is divided into three main areas: modeling perspectives, concepts and symbols which are evaluated separately.

Table 3 – Domain appropriateness: modeling perspectives

		Must	Should	Rationale
C-1	Structural modeling	3		The UML class diagram notation on which the CORAS language is inspired of is very suitable from a structural modeling perspective, and especially useful in the asset diagram.
C-2	Behavioral modeling	3		The UML use case diagram notation on which the CORAS language is inspired of has been further developed to show details of the processes including dependencies on paths (and/or gates), vulnerabilities and consequence and frequency/likelihood/ probability information.
Total score:		**6**	**-**	
Avg.:		**3**	**-**	

Table 4 – Domain appropriateness: concepts

		Must	Should	Rationale
C-3	Asset	3		Asset: the concept is supported.
C-4	Vulnerability	1		Vulnerability: the concept is supported.
C-5	Risk	3		Risk: the concept is supported.

C-6	Stakeholder	3		Stakeholder: the concept is supported. Since the concept stakeholder is heavily used in other domains like system engineering we have chosen to use "client" for the stakeholder who is paying for the analysis and decides risk evaluation criteria, "other parties" for stakeholders that also decides risk evaluation criteria, but do not pay for the actual analysis (e.g. authorities). The people who participate in the security analysis brainstorming are called "participants"
C-7	Threat	3		Threat: the concept is supported. To specify more in detail what kind of threat we have adopted the classification from ISO/IEC13335 [5] where threats are divided into human (deliberate and accidental origin) and environmental (which we just call non-human)
C-8	Unwanted incident	3		Unwanted incident: the concept is supported.
C-9	Treatment	3		Treatment: the concept is supported.
C-10	Likelihood/ frequency/ probability	3		Likelihood/frequency/probability: the concepts are supported.
C-11	Consequence	3		Consequence: the concept is supported.
C-12	Threat scenario	3		Threat scenario: the concept is supported.
C-13	The relations in the CORAS conceptual model	 3 3 3 3 3 3 3 3 3 2 2		This means the following relations: Stakeholder-asset: the relation is supported. Asset – vulnerability: the relation is supported. Threat-vulnerability: the relation is supported (Threat and treatment diagrams). Threat-unwanted incident: the relation is supported (Threat and treatment diagrams). Threat-asset: the relation is supported (Threat, risk, treatment and treatment overview diagram). Threat-risk: the relation is supported (Risk diagram). Unwanted incident-asset: the relation is supported. Treatment-risk: the relation is supported (Treatment overview diagram). Treatment-unwanted incident / threat / vulnerability: the relations are supported (Treatment diagram). Risk-asset: the relation is supported (Risk overview diagram). Risk-consequence/likelihood: partly supported, the likelihood and consequence of unwanted incidents are shown in threat diagrams, but the risk diagram only shows whether the risk is acceptable or not (based on the likelihood and consequence). Risk-unwanted incident: is not supported. The relation between risk and unwanted incident is shown by using identical names and similar identifiers.
		Avg.: 34/12= 2,8		
C-14	And/or operators	3		AND/OR operators: the concepts are supported.
C-15	L must be independent of the target that is assessed	3		The language is target independent.
C-16	Region	3		Physical or logical region has been introduced to bridge the gap between the risk related concepts and the target concepts.
Total score:		**41,8**	**-**	
Avg. :		**3**	**-**	

Table 5 – Domain appropriateness: symbols

		Must	Should	Rationale
C-17	Asset	3		Asset: special symbol exists.
C-18	Vulnerability	3		Vulnerability: special symbol exists.
C-19	Aggregated vulnerability		-	Aggregated vulnerability: no special symbol. There has not been any need for this symbol in any of the field trials, we therefore leave it out of the evaluation
C-20	Risk	3		Risk: special symbol exists.
C-21	Risk theme		-	Risk theme: no special symbol. There has not been any need for this symbol in any of the field trials, we therefore leave it out of the evaluation
C-22	Aggregated risk		-	Aggregated risk: no special symbol. There has not been any need for this symbol in any of the field trials, we therefore leave it out of the evaluation
C-23	Stakeholder	3		Stakeholder: special symbol exists.
C-24	Threat	3		General threat: only special symbols exists (the concepts has been specialized into three threat types)
C-25	Human threat		3	Human threat: special symbol exists (both for human accidental and deliberate)
C-26	Non-human threat		3	Non-human threat: special symbol exists.
C-27	Threat scenario	3		Threat scenario: special symbol exists.
C-28	Unwanted incident	3		Unwanted incident: special symbol exists.
C-29	Treatment	3		Treatment (scenario): special symbol exists.
C-30	Treatment type		-	Treatment type: Preventing treatment: no special symbol. There has not been any need for this symbol in any of the field trials; we therefore leave it out of the evaluation. Repairing treatment: no special symbol. There has not been any need for this symbol in any of the field trials, we therefore leave it out of the evaluation
C-31	Likelihood, frequency, probability		-	Likelihood/frequency/probability: no special symbols. There has not been any need for this symbol in any of the field trials, we therefore leave it out of the evaluation
C-32	Consequence		-	Consequence: no special symbol. There has not been any need for this symbol in any of the field trials, we therefore leave it out of the evaluation
C-33	Association type	3		Association type: Undirected association: exists in the form of a dashed line between assets and clients / other interested parties. Directed association: There exist a dashed arrow for treatments and a solid one between threats, vulnerabilities, threat scenarios, unwanted incidents, risks and assets.
C-34	Operators	3		Operators: And-operator: special symbol exists. Or-operator: special symbol exists.
C-35	Region		3	Region: special symbol exists.
Total score:		**33**	**6**	
Avg.:		**3**	**3**	

H.II.2 Participant Language Knowledge Appropriateness

Table 6 – Participant language knowledge appropriateness

		Must	Should	Rationale
S-1	L's external representation must imitate the real world	3		The threat diagrams in CORAS let the modeler model natural sequences of events from initiator via event to "victim" (asset), and the directed associations are now pointing the "correct" direction to represent cause-events (left to right). Due to a more obvious way of structuring the diagrams using the three types of threats, and the new region element, symbolizing physical/logical parts of the target, we experience less chaotic diagrams.
S-2	The symbols used in L must be based on the most common interpretation of the concept-symbol	3		The symbols have been redesigned in a more uniform, intuitive way and we believe they now satisfy this requirement.
C-36	L must be understandable for people unfamiliar with modeling and without specific training	3		According to investigations [2, 4] there are still terms that are found more difficult to understand than other (e.g. risk, likelihood, frequency). The language still has similarities with UML diagrams and therefore the reader can benefit from UML knowledge, but we have omitted the non-intuitive constructs in UML like the arrows pointing against the reading direction and we have also extended the use case inspired notation with more elements (vulnerability, likelihood and consequence estimates, regions).
Total score:		**9**	**-**	
Avg.:		**3**	**-**	

H.II.3 Knowledge Externalizability Appropriateness

Table 7 – Knowledge externalizability appropriateness

		Must	Should	Rationale
S-3	L must help externalize tacit knowledge	3		The language uses the natural way of describing how a threat harms an asset, i.e. a logical sequence of events starting with the threat that initiates a threat scenario which leads to an unwanted incident that harms an asset. There is a clear left-to-right reading direction. It is now easy to see which vulnerabilities a threat exploits since they are distributed along the sequence of events.
S-4	It must be easy to model as part of actual work	2		The models can be difficult to create and complete from scratch "on-the-fly" as the information collected in a brainstorming often needs to be restructured and corrected before it can be modeled correctly, but using the new modeling tool and also the guidelines for how to model makes it much easier to model as part of the actual work.
C-37	It must be possible to model fault trees with L	3		Fault tree gates (AND/OR) have now been included in the language.
C-38	It should be possible to model event trees with L		1	Event tree diagrams: even though similar to fault trees, an event tree appears differently in its notation. To model event trees one needs a different type of modeling approach than the CORAS language provides. The intention, or meaning of an event tree may however be expressed, but it will not look like a traditional event tree.
Total score:		**8**	**1**	
Avg.:		**2,7**	**1**	

H.II.4 Comprehensibility Appropriateness

Comprehensibility appropriateness includes the language's internal and external representation.

Table 8 – Comprehensibility appropriateness: internal representation

		Must	Should	Rationale
C-39	L must build on most-common I of risk specific concepts	2		There are still some terms that are more difficult than others are, like risk and frequency measures (See *Appendices B* and *C*).
S-5	The concepts in L must be general rather than specialized	2		In this case the language has evolved away from this requirement where both threat and stakeholder have become more specialized since the last version. The reason was that to avoid misunderstandings and make it more concrete and realistic we felt the specializations were necessary The stakeholder concepts is "overloaded" and frequently used in other domains, the threat concept is too general to comprehend for novices to security analysis and needs to be brought down to a more pragmatic level.
S-6	The concepts in L must be composable	-		After several major field trials we have yet to see the need for composing elements, as a consequence we will therefore omit this requirement.
S-7	L must be flexible in precision	3		As with the previous version of the language, this version has no restrictions on flexibility with respect to precision. The modeler is free to choose his or her preferred level of precision. Nevertheless, to use the semantic mapping from diagrams to precise English one has to adhere to a set of basic modeling rules, which are described in the modeling guideline.
S-8	If the number of concepts has to be large, the phenomena must be organized hierarchically	3		The language has only a limited number of concepts and we therefore they do not need any hierarchical organization.
S-9	The concepts of L must be easily distinguishable from each other.	2		As mentioned in the evaluation of the UML profile (*Chapter 7*), the concepts threat and risk have been found difficult to distinguish by the model readers [4]. By now being able to model risk, and also use the specialized threat symbols we hope to mitigate this misunderstanding, since the original concept definitions have been kept.
S-10	The use of concepts must be uniform throughout the whole set of statements that can be expressed within L	3		There are no concepts that are used in a non-uniform manner, i.e. changes meaning according to how or where it is used.
S-11	L must be flexible in the level of detail	3		The CORAS language does not require the modeler to model in a particular level of details. With the new risk diagram and treatment overview diagram it is easier to make overview diagrams with little details.
Total score:		**18**	**-**	
Avg.:		**2,3**	**-**	

Table 9 – Comprehensibility appropriateness: external representation

		Must	Should	Rationale
C-40	It should be possible to associate the overall threat picture to target descriptions		3	The region can be used to link the risk documentation to target documentation.
S-12	L must contain constructs that can represent the intention of the underlying conceptual model	-		Covered by the requirements in the domain appropriateness section.
S-13	Symbol discrimination in L must be easy.	3		The icons have been redesigned with focus on creating a uniform set of icons, using the most intuitive interpretation of the concepts and with focus on usability. We therefore feel that the comprehensibility of the symbols now is optimal.
S-14	It must be easy to distinguish which of the symbols in L any graphical mark in M is a part of	2		See the requirement specification for details about each bullet point below: • It is sometimes difficult to avoid crossing lines from e.g. treatment to treated entity. The problem of crossing lines also arise related to incident scenarios-assets, threats-incident scenarios, unwanted incident-assets etc, this is a general problem of the inspiration source in UML use cases and leaves it to the modeler to draw aesthetically pleasing models. • The symbol for threat scenario is also used for risk, but the first uses the symbol on top of an oval and the second on top of a rectangle. We therefore feel they are sufficiently different, and they appear in the same diagram only during treatment identification. • The symbols are within the same writing system: pictorial • The number of colors used in symbols are limited and the diagrams do not appear too colorful • The symbols have been designed for both black-white and color printing.
S-15	The use of emphasis in L must be in accordance with the relative importance of the statements in the given M	-		An empirical investigation [3] has shown that the graphical means mentioned in this requirement were not preferred over textual information labels. We therefore omit the requirement in the rest of the evaluation. Nevertheless, we have changed the color of the treatment symbol from red to green since red was used in several symbols and the new modeling guideline provides instructions regarding modeling directions.
S-16	Composition of symbols should be made in an aesthetically pleasing way		2	There exists a new modeling guideline, but it does not address the topic of aesthetically pleasing models in an explicit manner. There are also aspects of this requirement that are more appropriate for traditional software design models, and not diagrams with a reading direction like the CORAS diagrams (e.g. place nodes with a high degree of importance in the centre of the drawing)
S-17	L should not have empty symbols.		3	There are no empty symbols.

S-18	***M*** must adhere to the most common principles from gestalt psychology	3		Various principles from gestalt psychology have been investigated [3] and some are also implemented: The color used for threat scenario and incident scenario (now unwanted incident) was too similar and their icons were both "redish", making them difficult to distinguish at first sight. This has now been addressed; threat scenario is still red, while unwanted incident is yellow. The size of an object had no particular meaning in the previous version of the CORAS language, but due to later investigations [3] the size of an object will not play any role. Closed contours were not used to represent aggregation, but since the concept of aggregation has shown little practical usefulness it will not be emphasized any further in this evaluation. Conceptual containment is represented by an include-arrow, and not within a close contour as the gestalt principles state. The conceptual containment was important when the incident scenarios included threat scenarios, but in the current version of the language, this type of inclusion is removed. The different links now are both wavy, straight and dashed which makes them easier to distinguish.
S-19	The most common modeling tasks should be as efficient as possible	3		The efficiency of making the different diagrams has improved considerably since the last version of the language. Modeling asset diagrams must still be done from scratch, but uses a much simpler and easier notation. Modeling threat diagrams also must be done from scratch, but when the diagram is complete it is used with only adding information in both risk estimation and treatment identification. In addition the risk diagram and the treatment overview diagram can be generated automatically in the tool.
C-41	There must be a reasonable number of diagram types	3		Basically there is only two main types of diagrams, the asset and the threat diagram, the rest of the diagrams are based on the threat diagram. Blow this table we discuss each diagram type.
C-42	L must have a precise semantics	**3**		To ensure that the interpretation of the diagrams can be understood by anyone reading them, the CORAS language has a structured semantics. The semantics provides a schematic mapping from each of the diagrams to precise English.
Total score:		**17**	**8**	
Avg.:		**2,8**	**2,7**	

Asset diagram:
After the revision the diagram has improved considerably with respect to understandability. The amount of details has been reduced and assets are now shown by their name and the asset symbol. The way of associating stakeholders/clients/other interested parties to assets is similar to the provisos version because this was satisfactory. A new aspect is the possibility to show which assets that in case of damage may harm also other assets, i.e. direct and indirect assets. We have so far left the vulnerabilities out of the diagram since they are more useful to show in threat diagrams. An idea for future work may be to generate "asset-vulnerability" diagrams automatically from the final threat diagrams.

Threat diagram:
With the inclusion of vulnerabilities and regions, the threat diagrams have become a richer and more precise diagram. The new threat symbols have also made it more intuitive to divide the threat diagrams in several types of diagrams when the information that is to be modeled

exceeds one diagram. The ability to set likelihood and consequence estimates directly into the diagram have also shown very useful in field trials.
There will always be issues related to how to structure the diagrams in a best possible manner, but except for the advices provided in the modeling guideline this is up to the individual modeler's style and preferences.

Risk diagram:
A diagram showing the threats, risks and assets in one overview is new compared to the UML profile. The diagram can be generated semi-automatically from threat diagrams (currently it does not generate whether a risk is acceptable/unacceptable). The risk diagram is no longer space consuming and very useful for giving overview presentations of the risks.

Treatment diagram:
This diagram is basically the final threat diagram where unacceptable risks are indicated on the relation between the unwanted incidents and assets and acceptable risks are removed (including the paths leading up to them if they are not leading to unacceptable risks as well). The treatment arrows are easier to distinguish since they have become both dashed and wavy. Still the diagram may easily become cluttered if many treatments are to be annotated to the diagram. As for the threat diagram, avoiding this becomes the responsibility of the modeler that has to decide when to split into more diagrams.

Treatment overview diagram:
This new diagram was included to give and overview of how to deal with the identified risks and is meant to be used in summaries or final presentations of the analysis results. The diagram can be generated semi-automatically (often need to rearrange the elements to make it look better), and shows which treatments that are planned for each risk. The diagram has given the modeler an opportunity to give high level descriptions of treatments suitable when presenting treatment strategies.

H.II.5 Technical actor appropriateness

Table 10 – Technical actor interpretation appropriateness

	Must	Should	Rationale
S-20	3		Formal syntax
S-21		3	There exists a schematic mapping, or pattern matching from diagrams to precise English.
Total score:	**3**	**3**	
Avg.:	**3**	**3**	

H.II.6 Organizational Appropriateness

Table 11 –Organizational appropriateness

	Must	Should	Rationale
S-22	2		Using the CORAS language will help reducing the effort needed to understand the overall risk picture and this understanding may again help the organization to control its risks. The language and its guidelines provide means to identifying and analyzing risks, as well as a consistent and complete way of documenting them. The experience from field trials and the feedback we have received there suggests that the CORAS language is indeed very helpful in the security analysis process. However, due to the lack of hard, empirical evidence for the fulfillment of this requirement we will not give it full score.

C-43	3		According to the feedback from field trials, the CORAS diagrams have eased the effort of explaining risks.
C-44	3		The language exists in an open source modeling tool that can be downloaded freely, as well as a Microsoft Visio stencil.
Total score:	**8**	**-**	
Avg.:	**2,7**	**-**	

H.III Conclusion

This section summarizes the results from modeling the test case with the CORAS language and the evaluation of CORAS using SEQUAL

H.III.1 Modeling the Core Security Risk Scenarios

For each phase in the security analysis process we have evaluated whether or not the CORAS language was able to express the core security risk scenarios and in this section we summarize the results.

Phase 1 – Context establishment:
In this phase, the CORAS language was capable of modeling all the important information from the core security risk scenarios. The decision of not modeling value definitions was taken from a usability perspective since we believe this kind of information is more suitable for presentation in a table than in a model. We have chosen not to include more than the name of the assets and stakeholders in the models because additional information will most likely clutter the diagram. The diagrams should be accompanied with textual explanations, which is a more appropriate place to give detailed descriptions of the elements in the diagram. In our opinion, anything that are not directly necessary in order to read and understand the diagram should be avoided.

Phase 2 – Risk identification:
All the information in phase 2 was successfully modeled with the CORAS language. As for the asset diagram, we decided to include only the names of the threats, vulnerabilities and assets to avoid messy and confusing diagrams. Any additional information about the elements should be stated in the accompanying textual description, which typically is required for the final analysis report.

Phase 3 – Risk estimation:
Modeling the information from this phase using the CORAS language required an extremely small effort. In practice the modeling consisted of adding textual information about likelihood and consequences to the elements in the threat diagram. This makes the diagram very suitable for modeling "on-the-fly" during a brainstorming session. There was nothing in the core security risk scenarios that was unable to express.

Phase 4 – Risk evaluation:
The risk evaluation phase basically takes the information gathered in the risk identification and estimation and compares it to the risk acceptance levels in phase 1. The CORAS language had no problems modeling this information. The language provides a set of overview diagrams that can be very useful when presenting the analysis' findings. The risk diagrams are created by "hiding" the vulnerabilities and threat scenarios between a threat and an unwanted incident in the threat diagrams. The unwanted incident symbol is replaced with a risk symbol with the appropriate risk value. Depending on the modelers choice of tools this can be done either manually (like in Microsoft Visio) or semi-automatically in the CORAS tool. The risk evaluation diagram is a variation of the risk diagram where acceptable risks are grayed out. The latter diagram type is not necessary in order to model the core security risk scenarios, but provides an alternative to the traditional risk matrix.

Phase 5 – Risk treatment:
The CORAS language has no trouble in illustrating the treatments in the core security risk scenarios, including where they should be applied. Treatment overview diagram is an extra diagram type, suitable for presentation purposes in reports etc., but not required in order to model the core security risk scenarios. The CORAS language has no particular diagram for modeling the expected effect of each treatment with respect to likelihood and consequence. Such information may be added to the treatment arrow, but experiences with the UML profile has shown that it makes the diagram quite messy [1]. The final risk value after applying a treatment can be illustrated by updating the risk diagram with the new risk value.

H.III.2 Language Quality

The table below summarizes the average scores for each quality aspect discussed above. The scores provides an *indication* of weak and strong aspects of the CORAS language, but since the *number* of requirements within each category and their *level of details* vary, and no weighting has been included to adjust the score for this, we only use this as an indication of how well the language meets our requirements. An "-" means that there are no "should-requirements".

Table 12 – Evaluation scores

Category	Must-requirements			Should-requirements		
	Score	# Req.	Avg.	Score	# Req.	Avg.
Domain appropriateness:						
- modeling perspectives	6	2	3		-	-
- concepts	41,8	14	3		-	-
- symbols	30	10	3	9	3*	3
Participant language knowledge appropriateness	9	3	3		-	-
Knowledge externalizability appropriateness	8	3	2,7	1	1	1
Comprehensibility appropriateness:						
- internal representation	18	7**	2,3		-	-
- external representation	17	6***	2,8	8	3	2,7
Technical actor interpretation appropriateness	3	1	3	3	1	3
Organizational appropriateness	8	3	2,7		-	-

* C-19, C-21, C-22, C-30, C-31, C-32 were omitted
** S-6 was omitted
*** S-12, S-15 were omitted

As we can see from the scores, the CORAS language has improved considerably from its starting point as a UML profile. There has especially been progress with respect to participant language knowledge appropriateness, knowledge externalizability appropriateness and comprehensibility appropriateness (external representation). However, some aspects may still be further improved, mainly related to comprehensibility appropriateness.

The lowest score for the CORAS language was given for the comprehensibility appropriateness of the language's internal representation (its concepts and relationships between these). As the investigations of security analysis terminology has shown, there are still some terms that are more difficult to understand, for instance risk and frequency measures (See details in *Appendices B* and *C*). The language has also evolved away from the

requirement of using general rather than specialized concepts since "threat" has become more specialized since the previous version. Threat is now specialized into human or non-human threats, and human threats are either categorized as having deliberate origin (e.g. sabotage) or accidental (e.g. blunders). The reason was actually to avoid misunderstandings by making it more concrete and realistic. The threat concept was too general to comprehend for novices to security analysis and it was necessary to bring it down to a more pragmatic level.

The criticism regarding the comprehensibility appropriateness of the external representation of the CORAS language can generalized to almost all modeling notations, UML included, and is related to drawing aesthetically pleasing models. When the CORAS diagrams grow in size they can easily become cluttered and difficult to read. This can only be avoided by carefully dividing the diagrams into smaller and more manageable pieces of information. Crossing lines may also become a problem, e.g. from treatment to treated entity, unwanted incidents to assets or from threats to threat scenarios. Also in this case the modeler has to use his or her experience to know when the maximum level of tolerable complexity in a diagram is reach and a splitting is necessary.

The main difficulty of achieving full score in knowledge externalizability appropriateness is related to the nature of the brainstorming session and its free and creative nature. It is highly recommended to document the information produced during the brainstorming immediately, preferably by modeling it "on-the-fly". This have however shown challenging as it sets high demands to the analysis secretary who has to be prepared to model new information, change and update previous models during the entire brainstorming session.

One of the requirements for organization appropriateness is that the language must help the organization to achieve its goals, in the security analysis setting seen as controlling the organizations security risks. To measure this one may monitor the organization before and after the entry of the CORAS language, but this is beyond the scope of our work. We have however found it to ease the explanation of the organization’s security risks, and this is an important aspect in the effort of controlling risks.

As already explained, the should-requirements are not that important to the language quality, but more of the type "nice-to-have". The low score on knowledge externalizability appropriateness for the should-requirements is related to expressing event trees. Although it is an important notation within security analysis, it requires additional constructs beyond the current syntax of the CORAS language. The intention or meaning of the event trees can however be expressed, but then in a form that no longer resembles traditional event trees.

H.III.3 Requirements that Were Omitted from the Evaluation

The following requirements were omitted from the evaluation, because either they are overlapping with other requirements, or they cannot be justified based on practical experience from industrial field trials:

C-19, C-21, C-22, C-30, C-31, C-32, S-6, S-12, S-15

These are identical to the requirements that were omitted in the evaluation of the UML profile [1] and should be considered removed in the next revision of the framework.

References

[1] Hogganvik, I., Lund, M. S., and Stølen, K., "Quality Evaluation of the CORAS UML profile", SINTEF ICT, Tech.rep. SINTEF A2199, 2007.

[2] Hogganvik, I. and Stølen, K., "Empirical Investigations of the CORAS Language for Structured Brainstorming", SINTEF ICT, Tech.rep. STF90 A05041, January, 2005.

[3] Hogganvik, I. and Stølen, K., "Investigating Preferences in Graphical Risk Modeling", SINTEF ICT, Tech.rep. SINTEF A57, 2006.

[4] Hogganvik, I. and Stølen, K., "Risk Analysis Terminology for IT-systems: does it match intuition?" in Proc. Int. Symposium on Empirical Software Engineering (ISESE'05), pp. 13-23, 2005.

[5] ISO/IEC13335-1, Information technology – Security techniques – Management of information and communications technology security - Part 1:Concepts and models for information and communications technology security management, 2004.