The CORAS Model-based Method for Security Risk Analysis

Folker den Braber, Gyrd Brændeland, Heidi E. I. Dahl,
Iselin Engan, Ida Hogganvik, Mass S. Lund,
Bjørnar Solhaug, Ketil Stølen, Fredrik Vraalsen

SINTEF, Oslo
September 2006

Contents

1 Introduction **1**

2 A Guided Tour to the CORAS Method **3**
2.1 Step 1: Introductory meeting 5
2.2 Step 2: High-level analysis 7
2.3 Step 3: Approval 10
2.4 Step 4: Risk identification 12
2.5 Step 5: Risk estimation 16
2.6 Step 6: Risk evaluation 17
2.7 Step 7: Risk treatment 19
2.8 Conclusions 20

3 Basic Concepts **23**
3.1 Introduction 23
3.2 Target of Evaluation 23
3.3 Risk 24
3.4 Treatment 27
3.5 Combined Diagram 27

4 The CORAS Risk Modelling Language **29**
4.1 Syntax and Semantics 29
4.2 Assets Overview Diagrams 30
4.3 Threat Diagrams 31
4.4 Risk Overview Diagrams 33
4.5 Treatment Diagrams 34
4.6 Treatment Overview Diagrams 37
4.7 Conclusion 39

5 Using CORAS to Assess Security in a Business Environment **41**
5.1 Step 1: Introductory Meeting 42
5.1.1 Client Presents System or Organisation 43
5.1.2 Characterise Focus and Scope of the Analysis 44
5.1.3 Plan the Analysis 45
5.2 Step 2: High-Level Analysis 46
5.2.1 Risks Analysts Present Target of Analysis 46
5.2.2 Identify Assets 49
5.2.3 High-Level Security Analysis 49
5.3 Step 3: Approval Meeting 50

5.3.1 Documentation of the Target of Analysis 50
5.3.2 Asset Values 51
5.3.3 Risk Evaluation Criteria 51
5.4 Step 4: Risk Identification 52
5.4.1 Structured Brainstorming 53
5.5 Step 5: Risk Estimation 55
5.6 Step 6: Risk Evaluation 56
5.7 Step 7: Risk Treatment 57
5.7.1 Identify Treatments 57
5.7.2 Evaluate Treatments 59
5.8 Finalisation Meeting 59

6 Using CORAS in Component Oriented System Development 61
6.1 Introduction 61
6.2 The Component Model 62
6.3 An Integrated Approach 63
6.4 Requirements 64
6.4.1 Requirements Definition 64
6.4.2 Requirements to Protection Definition 66
6.5 Components and their Assets 66
6.5.1 Component Identification 66
6.5.2 Component Asset Identification 67
6.6 Interactions 68
6.6.1 Component Interaction 68
6.6.2 Component Risk Interaction 69
6.7 Specification 70
6.7.1 Component Specification 70
6.7.2 Component Protection Specification 71
6.8 Summary and conclusion 71

7 The CORAS Tool 73
7.1 Introduction 73
7.2 The CORAS Tool 73

A The CORAS Guideline 77

Bibliography 85

Chapter 1

Introduction

Have you ever asked yourself any of the following questions:

- Should I worry when using my credit card on the internet?
- How safe is my internet bank account?
- Who has access to my personal health care records?
- Am I the only one reading my email?
- What are the consequences of a single personal mistake for my company?

Today, most business processes and communications as well as a lot of everyday situations involve IT-technology. Apart from requirements on functionality, this development of IT-systems has increased the need for security. Security issues are reaching the main headlines on a regular basis. Viruses, worms, misconfiguration and program bugs are common problems in a world where new releases and updates are almost as frequently announced as spam email pops up in our inboxes.

There are many aspects to the concept of IT-security. One of them is security risk analysis, which is an inevitable and crucial activity for every system developer, system user or system owner, in order to get control over and knowledge about the security level of the actual system. Security analyses are costly and time consuming and cannot be carried out from scratch every time a system is updated or modified, so a method is needed in which the analysis results can easily be accessed and maintained. As a solution for this, we propose the CORAS model-based method for security risk analysis (or simply CORAS).

CORAS is one of many methods for conducting security analyses, but at the moment of writing it is the only graphical or model-based approach. This gives CORAS several advantages:

- It presents precise descriptions of the target system, its context and all relevant security features in an easily accessible format.
- The graphical presentation of the information improves communication and interaction between parties involved in the analysis.
- CORAS also facilitates documentation of risk assessment results and the assumptions on which these results depend.

Structure

The CORAS model-based method for security risk analysis may be separated into three different components:

1. The CORAS risk modelling language, which includes both the graphical syntax of the CORAS diagrams and a textual syntax and semantics.
2. The CORAS method, a step-by-step description of the security analysis process, with a guideline for constructing the CORAS diagrams.
3. The CORAS tool[1], a tool for documenting, maintaining and reporting risk analysis results.

This document is structured as follows: In Chapter 2 we give a guided tour of CORAS, applying the method and language to an example case. Chapter 3 introduces the basic concepts, which are used in Chapter 4 to describe the syntax and semantics of the CORAS risk modelling language. The next two chapters, Chapter 5 and 6 explores two applications of CORAS: security assessment in a business environment and component oriented system development. Chapter 7 presents the CORAS tool. And finally, Appendix A is a guideline for the CORAS method.

Background

CORAS is based on traditional security analysis techniques like the structured brainstorming technique HazOp [24], fault tree analysis (FTA) [12] and the failure mode and effects analysis (FMEA) [1], combining them with system development techniques such as the UML [26], which is the de facto standard modelling language for information systems.

CORAS furthermore takes into account international standards for risk management, such as the Australian/New Zealand Standard for Risk Management, AS/NZS 4360:2004 [27], the ISO/IEC 17799 Code of Practice for Information Security Management [13], the ISO/IEC 13335 Guidelines for the management of IT-Security [16], and system documentation in the form of the Reference Model for Open Distributed Processing [15].

[1] The tool may be downloaded from `http://coras.sourceforge.org`.

Chapter 2

A Guided Tour to the CORAS Method

This chapter presents a case-driven guided tour of the CORAS method for model based security analysis. 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 chapter devotes a separate section to each of them. The chapter focuses in particular on the use of the CORAS risk modelling language as a means for communication and interaction during the seven steps.

The seven steps 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.

- 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 being made and other preconditions 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 evaluation. 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 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.

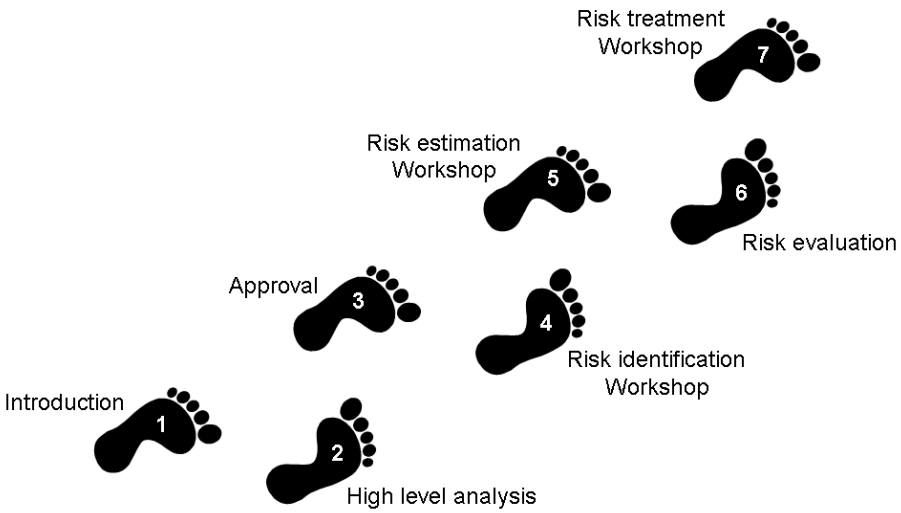

Figure 2.1: The seven steps of the CORAS method

In the following sections we will make a presentation of the seven steps by means of an example; the CORAS method applied to a telemedicine case[1].

> The example is distinguished from general descriptions of the method, language etc. by being formatted with with wider margins.

Let us introduce the example that will follow us throughout this chapter:

> In one region of the country, an experimental telemedicine system has been set up. A dedicated network between the regional hospital

[1] This example is inspired by security analyses of real telemedicine systems conducted within the CORAS project [15, 24]. However, changes have been made so that it no longer represents any real case.

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 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 (referred to as "the analysts") to do the job. In cooperation with the ministry, they organise the meetings and workshops that a security analysis consists of.

2.1 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 may 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 already mentioned 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 PHCCs are participating.

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 2.2. 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.

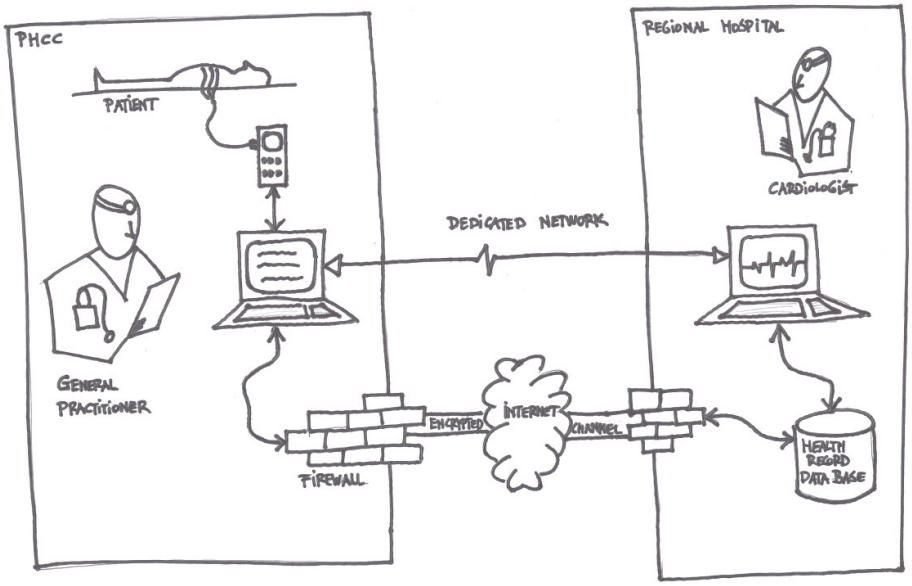

Figure 2.2: Picture of target

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 wellbeing, 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.

2.2 Step 2: High-level analysis

The second step is called the high-level analysis, and as the name indicates this involves conducting an initial, preliminary analysis of the target. Hence, also 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 analysis work during the remaining four steps of the analysis is directed towards these assets. The outcome of the high-level analysis helps the analysts to identify the target aspects 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 meantime in two diagrams, a UML class diagram (Figure 2.3) and a UML collaboration diagram (Figure 2.4). Furthermore, the medical doctor's description of use has been captured as a UML activity diagram (Figure 2.5). During this presentation the participants representing the client make corrections and eliminate errors, so that the result is a target description that is agreed upon. In the class- and collaboration diagrams the security analysis leader has also indicated what he understands as the focus areas of the analysis.

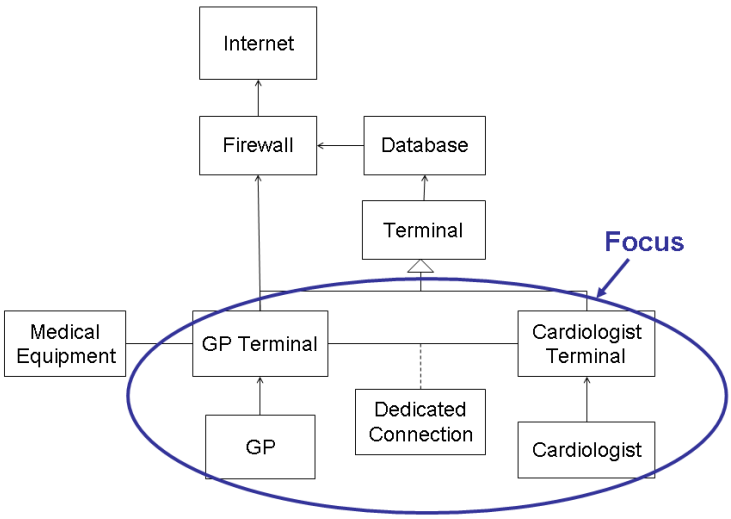

Figure 2.3: Class diagram

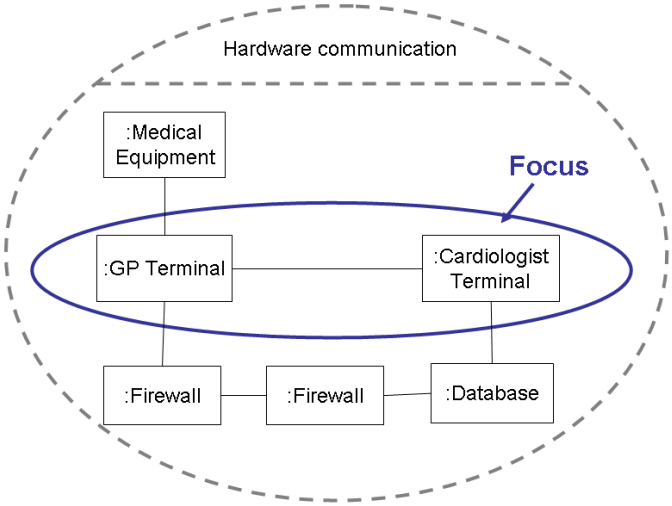

Figure 2.4: Collaboration diagram

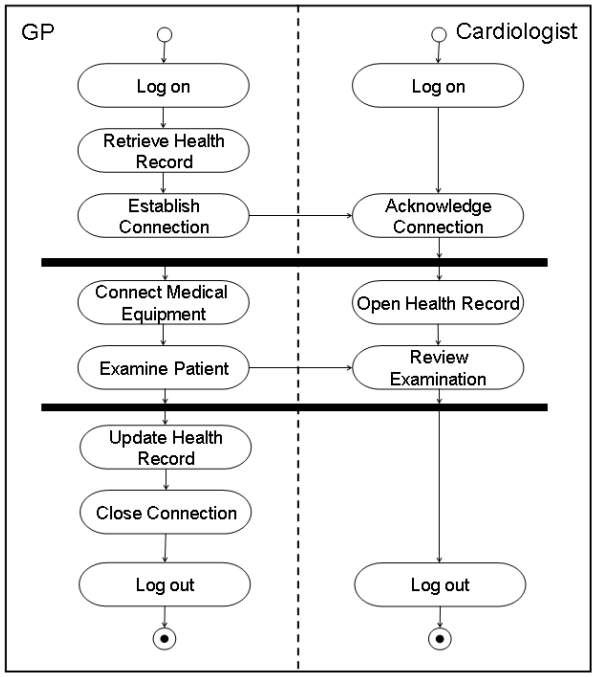

Figure 2.5: Activity diagram

After agreeing on a target description, the analysis moves on to asset identification. An asset is something in or related to the target that the client assigns great value to. Based on the discussion at the introductory meeting, the analysis leader has prepared an initial asset diagram showing the parties, and their main assets of relevance in the target (Figure 2.6). The National Ministry of Health is the client (i.e. the party that is initiating and paying for the analysis). There are 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".

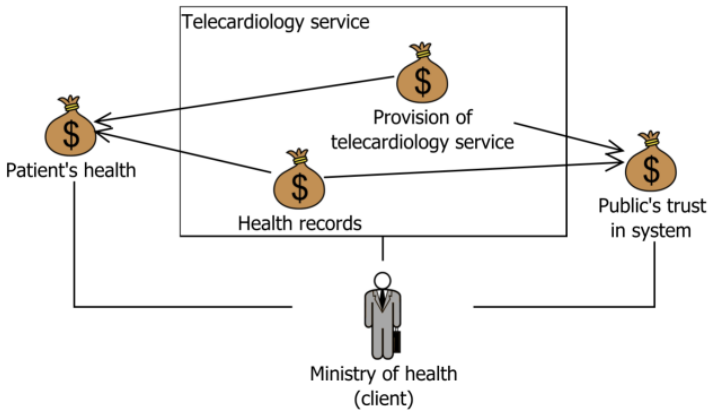

Figure 2.6: Asset diagram

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. The client is concerned about hackers, eavesdroppers, system failure and whether the security mechanisms are sufficiently secure. 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 2.1.

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
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

Table 2.1: High-level risk table

2.3 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 be done per 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) that is agreed upon and that everyone commit to. The approval also involves defining consequence scales (for unwanted incidents) 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.). 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 in system" is to be kept outside the scope.

Risks are events that harm assets when they occur, or, in other words, reduce the value of at least one of the identified assets. However, often the client accepts some risks, either because of shortage of resources or because of conflicting concerns. 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 2.2).

Asset	**Importance**	**Type**
Health records	2	Direct asset
Provision of telecardiology service	3	Direct asset
Public's trust in system	3	Indirect asset
Patient's health	1	Indirect asset

Table 2.2: Asset table

Having finished the asset table, they go on to define scales for the likelihood 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 incidents may have different impact depending on which asset is harmed, they make separate consequence scales for each of the direct assets. Table 2.3 shows the consequence scale defined for the asset "Health records" and Table 2.4 gives the likelihood scale defined for the target as such.

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 2.3: Consequence scale for "health records"

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 identify and evaluate possible treatments for it. They define these criteria by means of a risk

Likelihood value	**Description**[2]
Certain	Five times or more per year (50-*: 10y = 5-*: 1y)
Likely	Two to five times per year (21-49: 10y = 2,1-4,9: 1y)
Possibly	Once a year (6-20: 10y = 0,6-2: 1y)
Unlikely	Less than once per year (2-5: 10y = 0,2-0,5: 1y)
Rare	Less than once per ten years (0-1:10y = 0-0,1:1y)

Table 2.4: Likelihood scale

evaluation matrix for each asset. The security analysis leader draws the matrix for "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 each cell in the matrix as "acceptable" or "must be evaluated". The resulting risk evaluation matrix is show below (Table 2.5), and the participants decides to let this matrix yield for the other assets as well.

		Consequence				
		Insignificant	*Minor*	*Moderate*	*Major*	*Catastrophic*
Frequency	*Rare*					
	Unlikely	Acceptable				
	Possible					
	Likely				Must be evaluated	
	Certain					

Table 2.5: Risk evaluation matrix

After completing this task for all direct assets the analysts and the participants have the framework and vocabulary they need to start identifying threats, vulnerabilities, unwanted incidents and risks, and can move on to the next step.

2.4 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 risk modelling language. We will now exemplify how we model risks with the CORAS

language, using the symbols presented in Figure 2.7.

Figure 2.7: Symbols from the CORAS risk modelling language

The analysis leader challenges the participants to work with questions like: What worries you most 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 2.2-2.5) 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).

The initial threat scenario diagrams (Figure 2.8-2.10) have 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 type of threat they describe (human accidental, human deliberate and non-human threats).

The threat diagram in Figure 2.8 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. A confidentiality breach of the health records can also harm

the public's trust in the system. The system allows for irregular handling of health records where an employee accidentally may cause a leakage of records. This may compromise the confidentiality of the patient's health information and in the outmost consequence affect the patient's health.

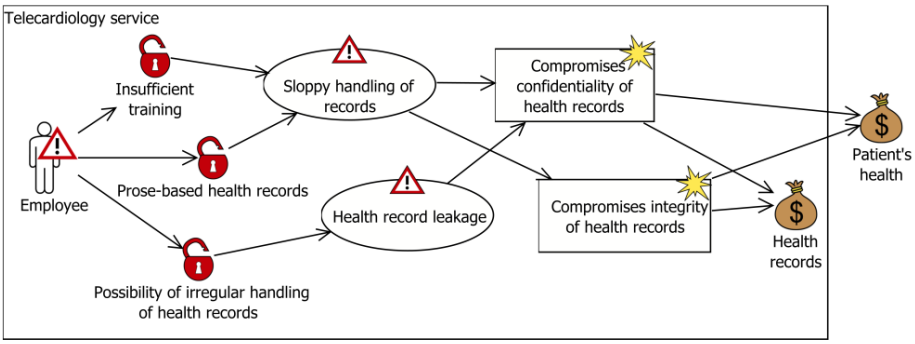

Figure 2.8: Initial threat diagram: accidental actions

In the threat diagram describing deliberate harmful actions caused by humans, the participants have identified two main threats: hacker and eavesdropper (Figure 2.9). A hacker may exploit insufficient security mechanisms to break in to 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.

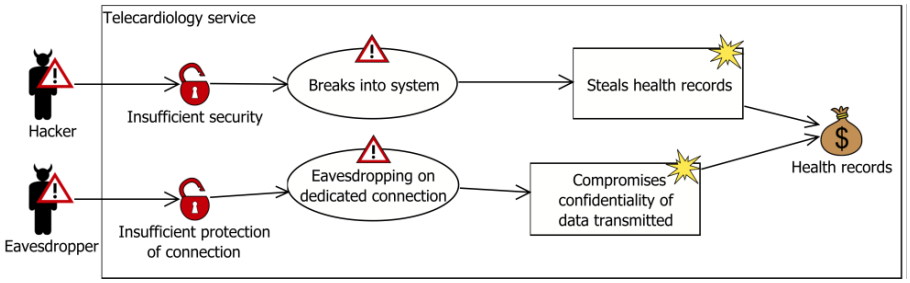

Figure 2.9: Initial threat diagram: deliberate actions

In addition to human threats, the participants worry about threats like system failure and network failure (Figure 2.10). 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.

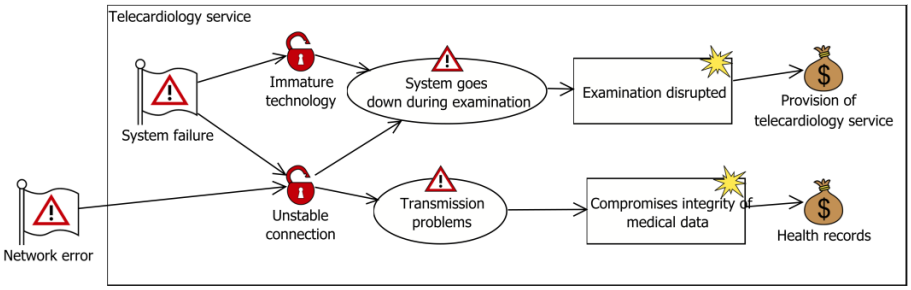

Figure 2.10: Initial threat diagram: non-human threats

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 receives much attention among the participants and develops into Figure 2.11. 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 for 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 doing 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 give set a patient's diagnosis, also the ability of providing a telecardiology service.

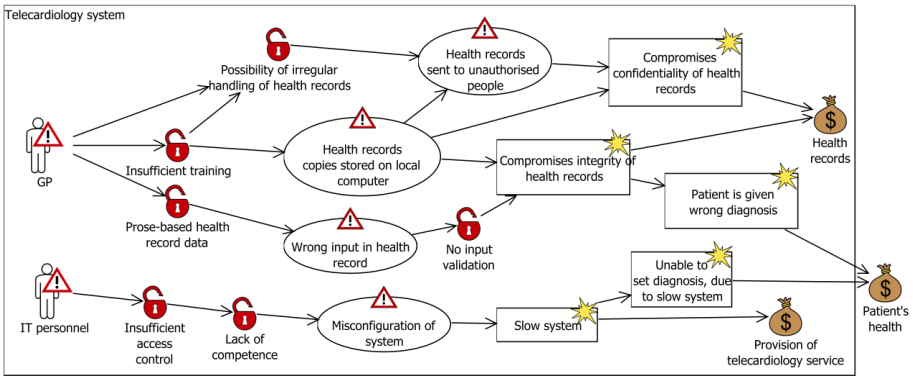

Figure 2.11: Final threat diagram: accidental actions

2.5 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 not. The participants in the workshop provide likelihood estimates for each threat scenario in the threat diagrams. For scenarios the likelihood of which are difficult to estimate, the analysis leader gives suggestions based on historical data like security incident statistics or personal experience. The likelihoods 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 2.12.
>
> There are different ways of computing the likelihood of an incident that may be caused by more than one threat scenario. If the esti-

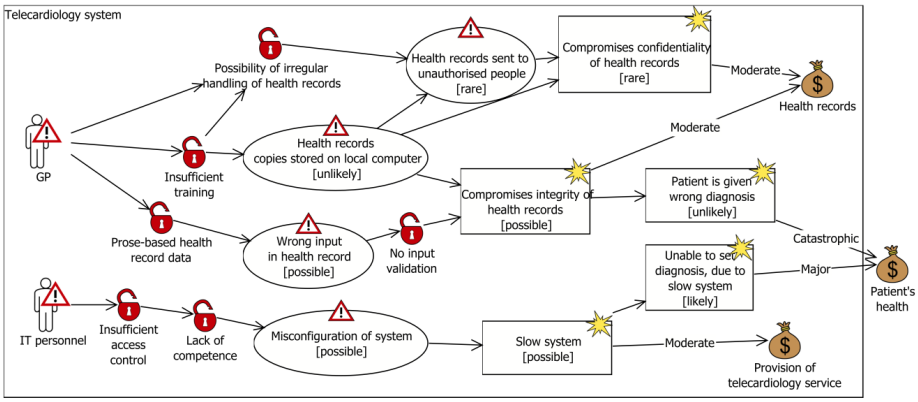

Figure 2.12: Threat diagram with likelihood and consequence estimates

mates 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 records copies stored on local computer" can both lead to "Compromises confidentiality of health records". In Table 2.6 we show how the combined likelihood is estimated. It is of course important that the combined estimates reflect reality, meaning that the combined estimates should be presented to the participants for validation.

Threat scenario	Likelihood	Unwanted incident	Combined likelihood
Health records sent out to unauthorised people	Rare (0-1:10y)	Compromises confidentiality of health records	(0-1:10y) + (2-5:10y) = (2-6:10y) Some overlap between unlikely and possible, but fits best in the unlikely interval.
Health records copies stored on local computer	Unlikely (2-5:10y)		

Table 2.6: Combined likelihood estimates

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".

2.6 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' 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 2.7 shows the risks placed in the risk evaluation matrix.

		Consequence				
		Insignificant	*Minor*	*Moderate*	*Major*	*Catastrophic*
Likelihood	*Rare*			CC1		
	Unlikely					PR1
	Possible			CI1, SS2		
	Likely				SS1	
	Certain					

Table 2.7: Risk evaluation matrix with risks

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.

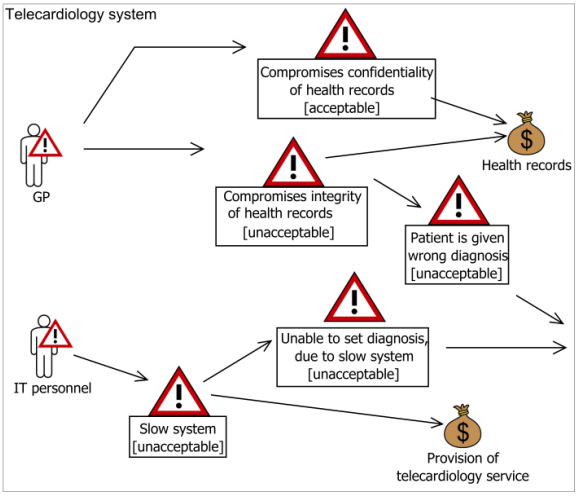

Figure 2.13: Risk overview diagram

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 2.13. Since the risk of compromising the confidentiality of health records is within the acceptable levels it will not be assessed in the treatment identification.

2.7 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 found to be unacceptable are all evaluated to find means to reduce them. A treatment should contribute to reduced likelihood and/or consequence of an incident. 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.

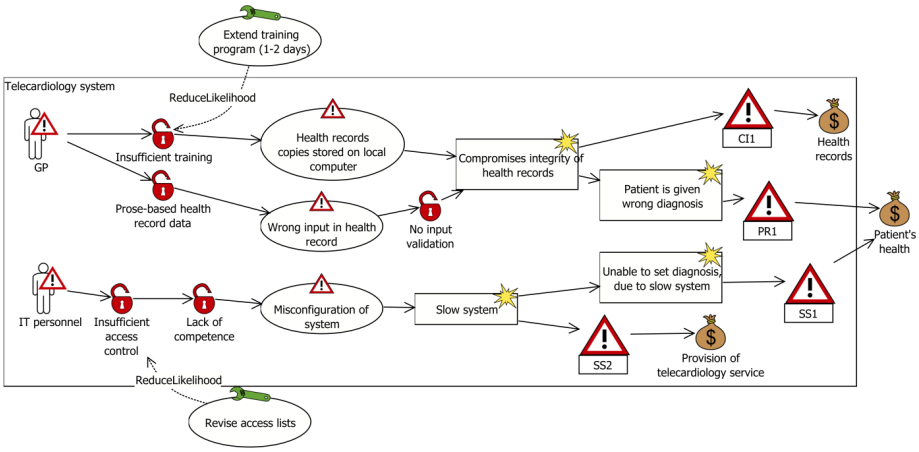

Figure 2.14: Treatment diagram

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[3]

[3]The alternative options are reduce likelihood, reduce consequence, avoid or retain risk.

indicating where they will be implemented. Finally, the following treatments are suggested and annotated to the treatment diagram in Figure 2.14:

- 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.

When the final results from the analysis are to be presented to the client and others with interest in the results, an overview of the risks and the proposed treatments is useful. In our case the treatment overview diagram of Figure 2.15 is used for this presentation.

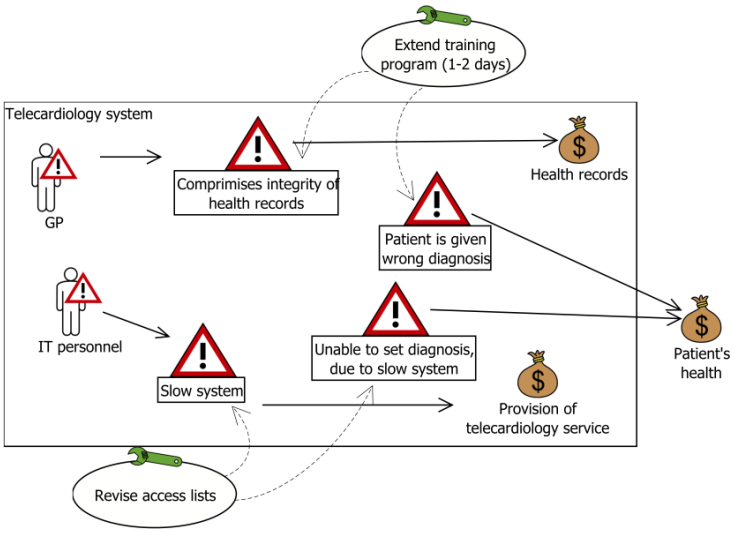

Figure 2.15: Treatment overview diagram

2.8 Conclusions

This chapter has demonstrated the use of the CORAS method in the security analysis of a telemedicine system. The main focus has been on the CORAS 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 overview 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.
- treatment overview diagram suitable for presentation of the final treatments.

For a more thorough presentation of the CORAS diagrams, see Chapter 4.

Chapter 3

Basic Concepts

3.1 Introduction

The activities involved in a security risk analysis comprise people of various backgrounds, and it is of crucial importance that these people are able to discuss and communicate about security risks and related concepts without ambiguities, uncertainties, misunderstandings, etc. For this reason, a conceptual framework must be in place that precisely defines all the relevant notions and that captures the relations that exist between the concepts. The introduction of a clear and understandable set of concepts is important also in order to ensure that the documentation of the security risk analysis is as clear and intelligible as possible for those who are to use this documentation on later occasions.

Several strategies have been deployed in order to develop a a conceptual framework the notions of which have a common understanding among the users. Firstly, the notions are to a large extent based on accepted standards for security and risk management, in particular [14] and [27]. Such a basis ensures that there are major research communities sharing our understanding and use of the terminology. Secondly, we aim at defining notions such that the use and intuitive understanding of our terminology harmonise with everyday use in English prose. Thirdly, the terminology has been extensively tested over several years of research both through scientific publications and through security analyses in field trials. Finally, the terminology has been tested empirically by interviewing several people of various background on their intuitive understanding of the CORAS concepts [11].

The notions introduced in this chapter will be precisely defined in English prose. As a rule, they will also be modelled in UML 2.0 class diagrams in which the relevant relations between the concepts are precisely accounted for.

3.2 Target of Evaluation

The notions introduced in this section are related to all the various steps of the CORAS method for security risk analysis, but in particular steps 1 through 3. During these steps, the system or organisation that is subject to the analysis, including the affected parties and the assets of the system, must be identified and documented.

Target of evaluation: The information system or organisation that is the subject of the security risk analysis. Often referred to as the target.

Given the target of evaluation, there may be various organisations, systems, users, etc. that are related to and have some interest in the target. Those on behalf of which a security risk analysis is conducted, and hence have valuables within the target that should be protected, are referred to as the parties of the analysis:

Party: An entity related to the target of evaluation and for which the security analysis is held.

The client of a security analysis, i.e. the one paying for the analysis and for whom the analysis is conducted, is usually the most central party of the target. It is the client's assets that are to be protected and the risks towards which that should be identified and mitigated if necessary.

A security risk analysis is conducted for the purpose of protecting the valuables of the parties of the analysis. The notion of asset denotes such a valuable.

Asset: Something to which a party of the target directly assigns value and, hence, for which this party requires protection.

An asset can be both physical and intangible. An example of the former is the physical hardware on which sensitive information is stored, and an example of the latter company brand.

Figure 3.1 shows the UML class diagram capturing the aspects of a target of analysis discussed so far. Notice that we assume a party to have at least one asset, for without assets there is nothing to risk and hence no point in conducting a security analysis.

Also notice that an asset is uniquely linked to a single party. This is because individual parties value entities differently: Something of high value to one party may be of insignificant value to another. Our definition of an asset furthermore implies that if a party is removed from the target, all the assets associated with this party are removed too.

Figure 3.1: Asset and party

3.3 Risk

The notions of this section are foremost associated with steps 4 through 6 of the CORAS method for security analysis. Assuming that the target of evaluation is properly accounted for and documented, the risk related aspects of the target are to be identified.

Having identified the parties and assets of the target, it is time to investigate the different ways in which the system can be harmed. Firstly, there are the weaknesses of the system that allows for the target to be harmed to begin with:

Vulnerability: A weakness, flaw or deficiency by the system that opens for a threat to harm or reduce the value of assets.

A vulnerability can be understood as something that is "missing" in the target, e.g. a company network without a firewall to protect against unauthorised attacks.

Threat: A potential cause of an unwanted incident.

A threat is what causes harm to assets of the target, and may be both human and non-human. A typical example of a human threat is a hacker that breaks into the target system, whereas a non-human threat can be hardware failure, software bug or natural threats such as flood. As to the human threat, it is further categorised into intentional and unintentional threat. An intentional threat may for example be an employee that deliberately leaks sensitive information to the press in return for money, whereas an unintentional threat may be an employee that accidentally forwards an e-mail containing sensitive information to someone that is not authorised to access this information.

The specific damaging event that is introduced or caused by a threat is defined as follows:

Unwanted incident: An event that may harm or reduce the value of assets.

Figure 3.2 shows the class diagram relating the introduced concepts. The notion of asset, and hence also the notion of party, is crucial to this picture; if there are no assets in the target of evaluation, there can be no vulnerabilities, threats or unwanted incidents. Notice further that what counts as a threat, vulnerability and unwanted incident depends on the party in question.

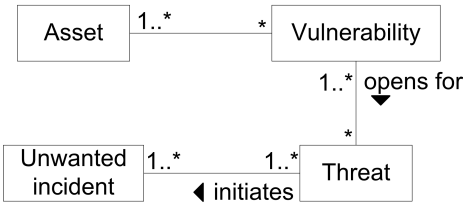

Figure 3.2: Vulnerability, threat and unwanted incident

The diagram of Figure 3.2 shows that a vulnerability must have at least one associated asset. A vulnerability, however, can exist without there being any threat present to introduce an unwanted incident because of this vulnerability.

Figure 3.3 shows the classification of threats to a given target.

The notions introduced so far forms the conceptual basis for the notion of risk which is defined as follows:

Risk: A risk is the chance of the occurrence of an unwanted incident.

The value or level of a risk varies depending on two factors, viz. how often the unwanted incident occurs and the impact or damage it causes when it occurs. The rate and impact of an unwanted incident are defined as follows, respectively:

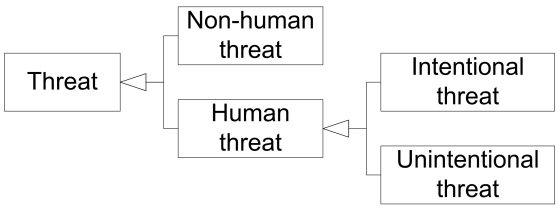

Figure 3.3: Threat

Likelihood: The frequency or probability for an unwanted incident to occur.

Consequence: The impact of an unwanted incident on an asset in terms of reduction or loss of asset value.

During a security analysis, the participants of the analysis choose the way in which to represent likelihood and consequence by convenience. The likelihood can be represented qualitatively, e.g. by operating with the values "unlikely", "possible" and "certain", it can be represented by stating how many times the incident occurs per time unit, e.g "five times a year", or it can be defined as a probability usually as a number ranging from 0 to 1, alternatively as a percentage. The class diagram for the notion of likelihood is given in Figure 3.4.

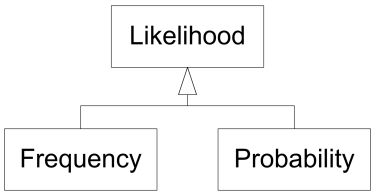

Figure 3.4: Likelihood

The consequence may likewise be represented either quantitatively or qualitatively. A quantitative representation can e.g. be to specify the monetary loss that is caused by the unwanted incident, whereas a qualitative representation can be to define a consequence scale with values such as "minor", "moderate" and "major".

The value of a risk is now defined as follows:

Risk value: The level or value of a risk as derived from the likelihood and the consequence of an unwanted incident.

Figure 3.5 shows the class diagram capturing the notion of risk. We see that for each risk, there is exactly one asset and one unwanted incident. This means that if there is one unwanted incident that when it occurs harms several assets, the number of risks represented by this incident equals the number of affected assets. Alternatively one could associate one risk with each unwanted incident, but then it would be more difficult to assess the consequence of the risk.

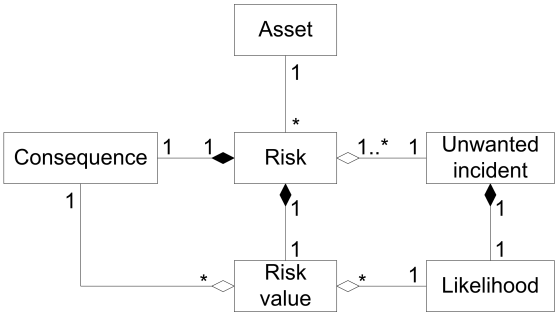

Figure 3.5: Risk

3.4 Treatment

This section is associated mainly with the last and seventh step of the CORAS method, i.e. risk mitigation or treatment. The notion of treatment is defined as follows:

Treatment: The selection and implementation of appropriate options for dealing with risk. A means that is directed towards one or more risks with the objective of reducing risk value.

A treatment can be directed towards the unwanted incident, the threat introducing the incident, the vulnerability associated with the affected asset or a combination of these. In any case, the ultimate goal is to reduce the risk value, and by the above definitions we know that this can be done by reducing the consequence and/or the likelihood of the risk.

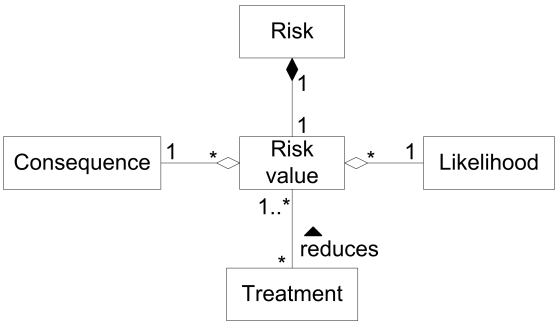

Figure 3.6: Treatment

3.5 Combined Diagram

Before we conclude we present here the class diagram combining all the notions introduced in this chapter. Notice that we for the purpose of readability have omitted the specialisations of threat and likelihood as shown, respectively, in Figure 3.3 and Figure 3.4 above.

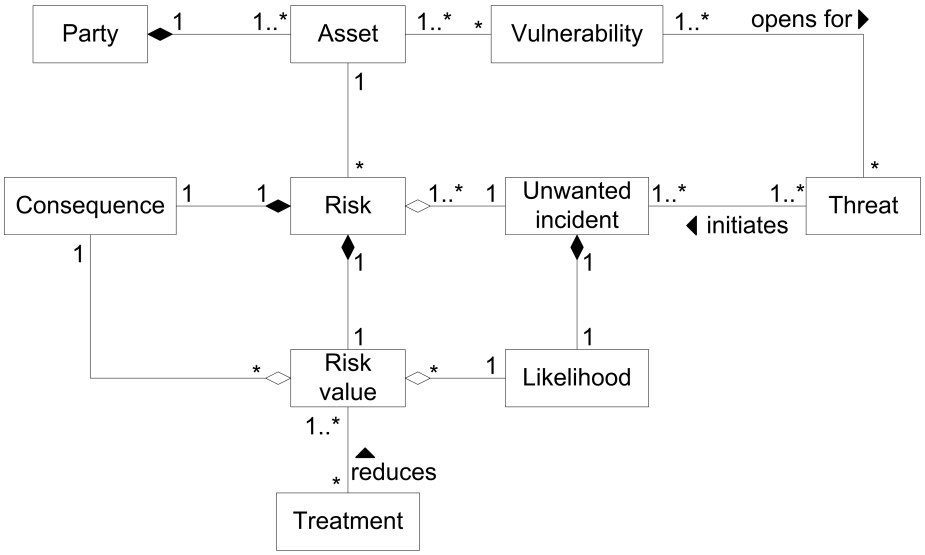

Figure 3.7: The basic concepts of the CORAS method

Chapter 4

The CORAS Risk Modelling Language

In this chapter, we use the concepts introduced in Chapter 3 to construct CORAS diagrams, visual presentations of the information gathered and generated in a risk analysis. In order to give the diagrams a precise meaning, we propose a textual syntax (as opposed to the graphical syntax of the diagrams) and a semantics in English.

4.1 Syntax and Semantics

To formulate the textual syntax for the CORAS risk modelling language, we use an extended BNF notation (EBNF), where braces {_} (respectively $\{\_\}^+$) means an ordered sequence of zero (respectively one) or more repetitions of the enclosed element, and square brackets [_] denotes optional features. As usual, the vertical bar _|_ represents options, and we use a shorthand notation for options within relations: we write $a \mid b \rightarrow c \mid d$ instead of $a \rightarrow c \mid a \rightarrow d \mid b \rightarrow c \mid b \rightarrow d$.

We will use the following naming convention: We use initials to denote instances of the elements of the syntax, e.g. p for parties, a for assets etc. The only exception is treatment scenarios, which we will denote trs. When there are more than one of a kind we use subscripts, e.g. t_1, t_2 for two threats, or $\{v_i\}$ for an arbitrary number of vulnerabilities. When we want to refer to the identifier associated with an element, we use the suffix -id, e.g. rid for the identifier of a risk.

The semantics of a language is a translation of a syntax into an already well-known language. In this chapter, we will translate the textual syntax we propose for the CORAS language into English. We write this as a function

$$[\![\_]\!] : \text{textual syntax} \longrightarrow \text{English}. \tag{4.1}$$

Given a diagram or the corresponding textual expressions, the semantic function will return a text which is a precise description of the meaning the diagram.

In the next sections, we will describe the different types of CORAS diagrams. At the end of each section, we will propose a textual syntax and semantics.

4.2 Assets Overview Diagrams

The assets overview diagram is an overview of the assets and how harm to one asset may affect others. It has two basic components, parties and assets, shown in Figure 4.1.

Figure 4.1: Components of the assets overview diagram

In addition to these basic building blocks we have two different relations which are displayed in the assets overview diagram: value and affect. We want to display the relationship inherent in the definition of an asset; i.e. that it is something a party values. We also want to include assets not necessarily valued directly by the parties of the analysis, but which may be affected by harm to other assets.

To summarise, the assets overview diagram has value relations between parties and assets, and affect relations between assets. Figure 4.2 shows an example.

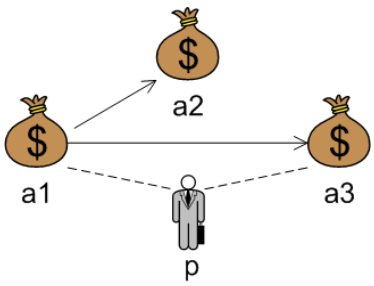

Figure 4.2: Assets overview diagram

Syntax of Assets Overview Diagrams

$$
\begin{aligned}
\langle value \rangle &::= \langle party \rangle \rightarrow \langle asset \rangle\,; \\
\langle affect \rangle &::= \langle asset \rangle \rightarrow \langle asset \rangle\,; \\
\langle party \rangle &::= \langle identifier \rangle\,; \\
\langle asset \rangle &::= \langle identifier \rangle\,; \\
\langle identifier \rangle &::= \text{A natural language description;}
\end{aligned}
$$

Semantics of Assets Overview Diagrams

$$
\begin{aligned}
[\![p \rightarrow a]\!] &:= [\![p]\!] \text{ wants to protect the value of } [\![a]\!]. \\
[\![a_1 \rightarrow a_2]\!] &:= [\![a_2]\!] \text{ may be harmed via } [\![a_1]\!]. \\
[\![p]\!] &:= \text{ party } p. \\
[\![a]\!] &:= \text{ asset } a.
\end{aligned}
$$

4.3 Threat Diagrams

A threat diagram is a complete overview of the chains of events initiated by threats, having consequences for the assets. Its basic components are the seven elements of Figure 4.3: deliberate, accidental and non-human threats, vulnerabilities, threat scenarios, unwanted incidents and assets. Threat scenarios and unwanted incidents may be assigned a likelihood.

Figure 4.3: Components of the threat diagram

Threat diagrams have two different relations: cause and impact. A threat exploits one or more vulnerabilities, causing a threat scenario or an unwanted incident. These may then result in new threat scenarios or unwanted incidents, either as direct consequences or by the threat exploiting other vulnerabilities. Each cause relation may be assigned a likelihood. An unwanted incident which has a consequence for an asset is said to impact it.

To summarise, threat diagrams have a binary cause relation between threats, threat scenarios and unwanted incidents, and an impact relation between unwanted incidents and assets. Figure 4.4 is an example of a threat diagram, incorporating all the possible elements and relations.

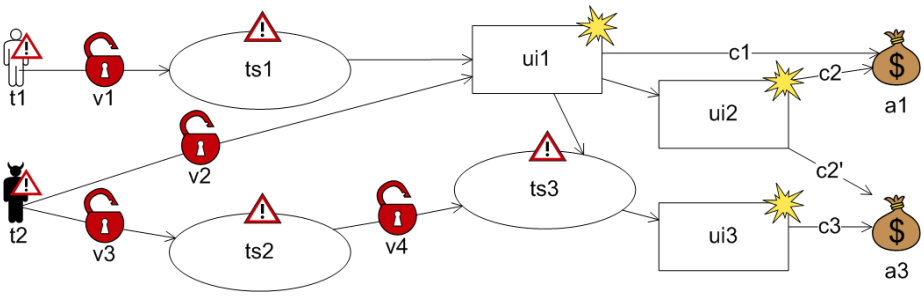

Figure 4.4: Threat diagram

Syntax of Threat Diagrams

$$
\begin{aligned}
\langle relation \rangle &::= \langle cause \rangle \mid \langle impact \rangle\,; \\
\langle cause \rangle &::= \langle threat \rangle \\
&\qquad \xrightarrow{\{\langle vulnerability \rangle\}^{+}[,\langle likelihood \rangle]} \\
&\qquad\quad \langle threat\ scenario \rangle \mid \langle unwanted\ incident \rangle\,; \\
\langle cause \rangle &::= \langle threat\ scenario \rangle \mid \langle unwanted\ incident \rangle \\
&\qquad \xrightarrow{\{\langle vulnerability \rangle\}[,\langle likelihood \rangle]} \\
&\qquad\quad \langle threat\ scenario \rangle \mid \langle unwanted\ incident \rangle\,; \\
\langle impact \rangle &::= \langle unwanted\ incident \rangle \xrightarrow{\langle consequence \rangle} \langle asset \rangle\,; \\
\langle threat \rangle &::= \langle identifier \rangle\,; \\
\langle vulnerability \rangle &::= \langle identifier \rangle\,; \\
\langle threat\ scenario \rangle &::= (\langle identifier \rangle\,[,\langle likelihood \rangle]); \\
\langle unwanted\ incident \rangle &::= (\langle identifier \rangle\,[,\langle likelihood \rangle]); \\
\langle asset \rangle &::= \langle identifier \rangle\,; \\
\langle likelihood \rangle &::= \langle qualitative \rangle \mid \langle quantitative \rangle\,; \\
\langle consequence \rangle &::= \langle qualitative \rangle \mid \langle quantitative \rangle\,; \\
\langle qualitative \rangle &::= \langle identifier \rangle\,; \\
\langle quantitative \rangle &::= \text{A number;} \\
\langle identifier \rangle &::= \text{A natural language description;}
\end{aligned}
$$

Semantics of Threat Diagrams

$$
\begin{aligned}
[\![t \xrightarrow{\{v_i\},l} ts]\!] &:= [\![t]\!] \text{ exploits } [\![v_i]\!] \text{ to initiate } [\![ts]\!], \text{ with a } [\![l]\!]. \\
[\![t \xrightarrow{\{v_i\},l} ui]\!] &:= [\![t]\!] \text{ exploits } [\![v_i]\!] \text{ to initiate } [\![ui]\!], \text{ with a } [\![l]\!]. \\
[\![ts_1 \xrightarrow{\{v_i\},l} ts_2]\!] &:= \text{A threat which has initiated } [\![ts_1]\!], \text{ exploits } [\![v_i]\!] \\
&\qquad \text{to initiate } [\![ts_2]\!], \text{ with a } [\![l]\!]. \\
[\![ts \xrightarrow{\{v_i\},l} ui]\!] &:= \text{A threat which has initiated } [\![ts]\!], \text{ exploits } [\![v_i]\!] \\
&\qquad \text{to initiate } [\![ui]\!], \text{ with a } [\![l]\!]. \\
[\![ui \xrightarrow{\{v_i\},l} ts]\!] &:= \text{A threat which has initiated } [\![ui]\!], \text{ exploits } [\![v_i]\!] \\
&\qquad \text{to initiate } [\![ts]\!], \text{ with a } [\![l]\!]. \\
[\![ui_1 \xrightarrow{\{v_i\},l} ui_2]\!] &:= \text{A threat which has initiated } [\![ui_1]\!], \text{ exploits } [\![v_i]\!] \\
&\qquad \text{to initiate } [\![ui_2]\!], \text{ with a } [\![l]\!]. \\
[\![ui \xrightarrow{c} a]\!] &:= [\![ui]\!] \text{ has } [\![c]\!] \text{ affecting } [\![a]\!]. \\
[\![t]\!] &:= \text{ threat } t. \\
[\![v]\!] &:= \text{ vulnerability } v. \\
[\![ts]\!] &:= \text{ threat scenario } tsid \text{ with } [\![l_{ts}]\!].
\end{aligned}
$$

$$
\begin{aligned}
[\![ui]\!] &:= \text{ unwanted incident } uiid \text{ with } [\![l_{ui}]\!]. \\
[\![a]\!] &:= \text{ asset } a. \\
[\![l]\!] &:= \text{ likelihood } l. \\
[\![c]\!] &:= \text{ consequence } c.
\end{aligned}
$$

4.4 Risk Overview Diagrams

A risk overview diagram is a summary of a threat diagram, displaying the risks posed to the assets by the threats. It has five basic components: deliberate, accidental and non-human threats, risks and assets (see Figure 4.5). Each risk is assigned a risk value indicating its severity.

Figure 4.5: Components of the risk overview diagram

The risk overview diagram has only one relation: an impact from a threat to an asset, showing the risks that threat poses to the asset. In the graphical presentations of the diagrams, we have chosen to represent this relation by multiple arrows to emphasise the causal relationship between consecutive risks.

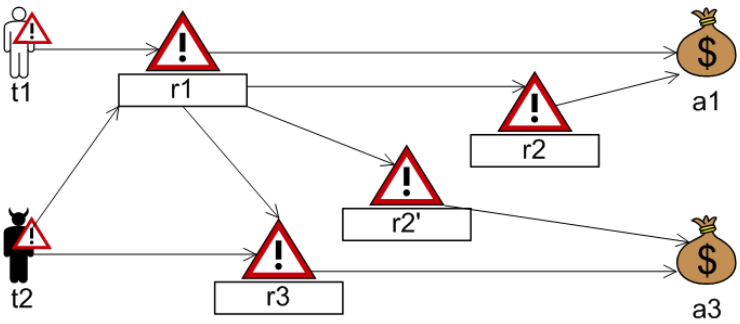

Figure 4.6: Risk overview diagram

Syntax of Risk Overview Diagrams

$$
\begin{aligned}
\langle relation \rangle &::= \langle impact \rangle\,; \\
\langle impact \rangle &::= \langle threat \rangle \xrightarrow{\{\langle risk \rangle\}^+} \langle asset \rangle\,; \\
\langle threat \rangle &::= \langle identifier \rangle\,; \\
\langle risk \rangle &::= (\langle identifier \rangle\,, \langle risk\ value \rangle); \\
\langle risk\ value \rangle &::= \langle qualitative \rangle \mid \langle quantitative \rangle \mid
\end{aligned}
$$

$$
\begin{aligned}
& \text{riskfunction}(\langle \mathit{consequence} \rangle, \langle \mathit{likelihood} \rangle); \\
\langle \mathit{asset} \rangle &::= \langle \mathit{identifier} \rangle; \\
\langle \mathit{consequence} \rangle &::= \langle \mathit{qualitative} \rangle \mid \langle \mathit{quantitative} \rangle; \\
\langle \mathit{likelihood} \rangle &::= \langle \mathit{qualitative} \rangle \mid \langle \mathit{quantitative} \rangle; \\
\langle \mathit{qualitative} \rangle &::= \langle \mathit{identifier} \rangle; \\
\langle \mathit{quantitative} \rangle &::= \text{A number}; \\
\langle \mathit{identifier} \rangle &::= \text{A natural language description};
\end{aligned}
$$

Semantics of Risk Overview Diagrams

$$
\begin{aligned}
[\![t \xrightarrow{r} a]\!] &:= [\![t]\!] \text{ poses a } [\![r]\!] \text{ to } [\![a]\!]. \\
[\![t]\!] &:= \text{threat } t. \\
[\![r]\!] &:= \text{risk } r. \\
[\![rv]\!] &:= \text{risk value } rv. \\
[\![rf(l,c)]\!] &:= \text{risk function } rf \text{ of } [\![l]\!] \text{ and } [\![c]\!]. \\
[\![(rid, rv)]\!] &:= \text{risk } rid \text{ with } [\![rv]\!]. \\
[\![(rid, rf(l,c))]\!] &:= \text{risk } rid \text{ with } [\![rf(l,c)]\!]. \\
[\![a]\!] &:= \text{asset } a. \\
[\![l]\!] &:= \text{likelihood } l. \\
[\![c]\!] &:= \text{consequence } c.
\end{aligned}
$$

4.5 Treatment Diagrams

A treatment diagram is a complete overview of proposed treatments of elements of a threat diagram. It is based on a threat diagram, replacing the consequences of the impact relations with the corresponding risks from the risk overview diagram, and adding proposed treatment scenarios. Its basic components are shown in Figure 4.7.

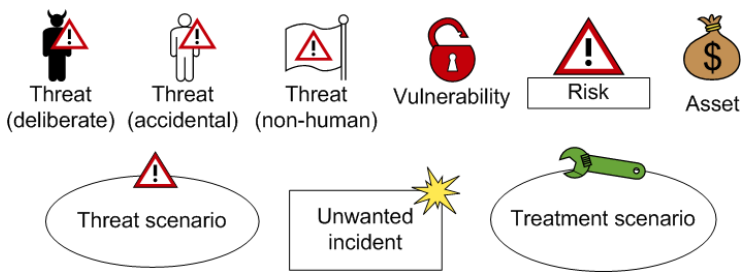

Figure 4.7: Components of the treatment diagram

We also have an additional relation, cure, from a treatment scenario to any other kind of element. Cure has an efficiency parameter, describing the expected effect of the treatment scenario. See Figure 4.8 for an example of a treatment diagram.

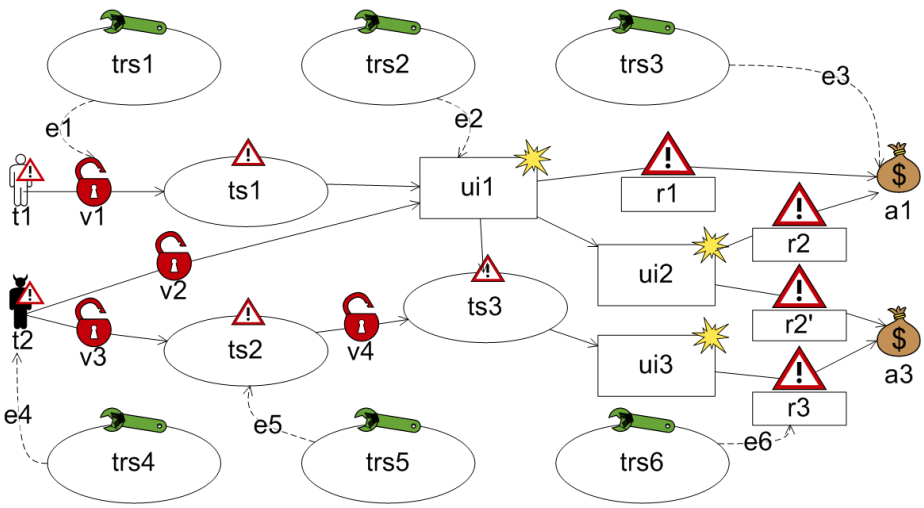

Figure 4.8: Treatment diagram

Syntax of Treatment Diagrams

$$
\begin{aligned}
\langle relation\rangle &::= \langle cause\rangle \mid \langle impact\rangle \mid \langle cure\rangle\,;\\
\langle cause\rangle &::= \langle threat\rangle\\
&\qquad \xrightarrow{\{\langle vulnerability\rangle\}^{+}[,\langle likelihood\rangle]}\\
&\qquad \langle threat\ scenario\rangle \mid \langle unwanted\ incident\rangle\,;\\
\langle cause\rangle &::= \langle threat\ scenario\rangle \mid \langle unwanted\ incident\rangle\\
&\qquad \xrightarrow{\{\langle vulnerability\rangle\}[,\langle likelihood\rangle]}\\
&\qquad \langle threat\ scenario\rangle \mid \langle unwanted\ incident\rangle\,;\\
\langle impact\rangle &::= \langle unwanted\ incident\rangle \xrightarrow{\langle risk\rangle} \langle asset\rangle\,;\\
\langle cure\rangle &::= \langle treatment\ scenario\rangle \xrightarrow{\langle effect\rangle} \langle threat\rangle \mid \langle vulnerability\rangle \mid\\
&\qquad \langle threat\ scenario\rangle \mid \langle unwanted\ incident\rangle \mid \langle risk\rangle \mid \langle asset\rangle\,;\\
\langle threat\rangle &::= \langle identifier\rangle\,;\\
\langle vulnerability\rangle &::= \langle identifier\rangle\,;\\
\langle threat\ scenario\rangle &::= (\langle identifier\rangle\,[,\langle likelihood\rangle]);\\
\langle unwanted\ incident\rangle &::= (\langle identifier\rangle\,[,\langle likelihood\rangle]);\\
\langle asset\rangle &::= \langle identifier\rangle\,;\\
\langle treatment\ scenario\rangle &::= \langle identifier\rangle\,;\\
\langle risk\rangle &::= (\langle identifier\rangle\,,\langle risk\ value\rangle);\\
\langle risk\ value\rangle &::= \langle qualitative\rangle \mid \langle quantitative\rangle \mid\\
&\qquad \mathrm{riskfunction}(\langle consequence\rangle\,,\langle likelihood\rangle);\\
\langle effect\rangle &::= \langle qualitative\rangle \mid \langle quantitative\rangle\,;\\
\langle likelihood\rangle &::= \langle qualitative\rangle \mid \langle quantitative\rangle\,;
\end{aligned}
$$

$$\langle consequence \rangle ::= \langle qualitative \rangle \mid \langle quantitative \rangle\,;$$
$$\langle qualitative \rangle ::= \langle identifier \rangle\,;$$
$$\langle quantitative \rangle ::= \text{A number;}$$
$$\langle identifier \rangle ::= \text{A natural language description;}$$

Semantics of Treatment Diagrams

$$\llbracket t \xrightarrow{\{v_i\},l} ts \rrbracket := \llbracket t \rrbracket \text{ exploits } \llbracket v_i \rrbracket \text{ to initiate } \llbracket ts \rrbracket\text{, with a } \llbracket l \rrbracket.$$
$$\llbracket t \xrightarrow{\{v_i\},l} ui \rrbracket := \llbracket t \rrbracket \text{ exploits } \llbracket v_i \rrbracket \text{ to initiate } \llbracket ui \rrbracket\text{, with a } \llbracket l \rrbracket.$$
$$\llbracket ts_1 \xrightarrow{\{v_i\},l} ts_2 \rrbracket := \text{A threat which has initiated } \llbracket ts_1 \rrbracket\text{, exploits } \llbracket v_i \rrbracket \text{ to initiate } \llbracket ts_2 \rrbracket\text{, with a } \llbracket l \rrbracket.$$
$$\llbracket ts \xrightarrow{\{v_i\},l} ui \rrbracket := \text{A threat which has initiated } \llbracket ts \rrbracket\text{, exploits } \llbracket v_i \rrbracket \text{ to initiate } \llbracket ui \rrbracket\text{, with a } \llbracket l \rrbracket.$$
$$\llbracket ui \xrightarrow{\{v_i\},l} ts \rrbracket := \text{A threat which has initiated } \llbracket ui \rrbracket\text{, exploits } \llbracket v_i \rrbracket \text{ to initiate } \llbracket ts \rrbracket\text{, with a } \llbracket l \rrbracket.$$
$$\llbracket ui_1 \xrightarrow{\{v_i\},l} ui_2 \rrbracket := \text{A threat which has initiated } \llbracket ui_1 \rrbracket\text{, exploits } \llbracket v_i \rrbracket \text{ to initiate } \llbracket ui_2 \rrbracket\text{, with a } \llbracket l \rrbracket.$$
$$\llbracket ui \xrightarrow{ri} a \rrbracket := \llbracket ui \rrbracket \text{ poses a } \llbracket ri \rrbracket \text{ to } \llbracket a \rrbracket.$$
$$\llbracket trs \xrightarrow{e} t \rrbracket := \llbracket trs \rrbracket \text{ reduces the likelihood of } \llbracket t \rrbracket \text{ with an } \llbracket e \rrbracket.$$
$$\llbracket trs \xrightarrow{e} v \rrbracket := \llbracket trs \rrbracket \text{ reduces the likelihood of a threat exploiting } \llbracket v \rrbracket \text{ with an } \llbracket e \rrbracket.$$
$$\llbracket trs \xrightarrow{e} ts \rrbracket := \llbracket trs \rrbracket \text{ reduces the likelihood of } \llbracket ts \rrbracket \text{ with an } \llbracket e \rrbracket.$$
$$\llbracket trs \xrightarrow{e} ui \rrbracket := \llbracket trs \rrbracket \text{ reduces the likelihood of } \llbracket ui \rrbracket \text{ with an } \llbracket e \rrbracket.$$
$$\llbracket trs \xrightarrow{e} r \rrbracket := \llbracket trs \rrbracket \text{ reduces the risk value of } \llbracket r \rrbracket \text{ with an } \llbracket e \rrbracket.$$
$$\llbracket trs \xrightarrow{e_l} (rid, rf(l,c)) \rrbracket := \llbracket trs \rrbracket \text{ reduces the } \llbracket l \rrbracket \text{ of } \llbracket (rid, rf(l,c)) \rrbracket \text{ with an } \llbracket e_l \rrbracket.$$
$$\llbracket trs \xrightarrow{e_c} (rid, rf(l,c)) \rrbracket := \llbracket trs \rrbracket \text{ reduces the } \llbracket c \rrbracket \text{ of } \llbracket (rid, rf(l,c)) \rrbracket \text{ with an } \llbracket e_c \rrbracket.$$
$$\llbracket trs \xrightarrow{e} a \rrbracket := \llbracket trs \rrbracket \text{ reduces the importance of } \llbracket a \rrbracket \text{ with an } \llbracket e \rrbracket.$$
$$\llbracket t \rrbracket := \text{ threat } t.$$
$$\llbracket v \rrbracket := \text{ vulnerability } v.$$
$$\llbracket ts \rrbracket := \text{ threat scenario } tsid \text{ with likelihood } l_{ts}.$$

$$
\begin{aligned}
[\![ui]\!] &:= \text{ unwanted incident } uiid \text{ with likelihood } l_{ui}. \\
[\![r]\!] &:= \text{ risk } r. \\
[\![rv]\!] &:= \text{ risk value } rv. \\
[\![rf(l,c)]\!] &:= \text{ risk function } rf \text{ of } [\![l]\!] \text{ and } [\![c]\!]. \\
[\![(rid, rv)]\!] &:= \text{ risk } rid \text{ with } [\![rv]\!]. \\
[\![(rid, rf(l,c))]\!] &:= \text{ risk } rid \text{ with } [\![rf(l,c)]\!]. \\
[\![a]\!] &:= \text{ asset } a. \\
[\![trs]\!] &:= \text{ treatment scenario } trs. \\
[\![l]\!] &:= \text{ likelihood } l. \\
[\![c]\!] &:= \text{ consequence } c. \\
[\![e]\!] &:= \text{ effect } e.
\end{aligned}
$$

4.6 Treatment Overview Diagrams

The last diagram of a CORAS risk analysis is the treatment overview diagram. It is a summary of proposed treatments affecting the risk values of the risks to the assets. It is based on a risk overview diagram, adding the treatment scenarios and cure relations from the treatment diagram. Note that the (optional) effects of the treatment scenarios are not the same here as in the treatment diagram, as they are influenced by the paths from the element treated to the risk. Figure 4.9 shows the six basic components of a treatment overview diagram.

Figure 4.9: Components of the treatment overview diagram

Here, the treatment scenarios cure either the threats and assets they treated in the treatment diagram, or the risk which are direct consequences or the elements they cure. As the risks following such a risk also are direct consequences of the original element, we omit the arrows from the treatment scenario to these risks. Figure 4.10 is an example of a treatment overview diagram.

Syntax of Treatment Overview Diagrams

$$
\begin{aligned}
\langle relation \rangle &::= \langle impact \rangle \mid \langle cure \rangle\,; \\
\langle impact \rangle &::= \langle threat \rangle \xrightarrow{\{\langle risk \rangle\}^+} \langle asset \rangle\,; \\
\langle cure \rangle &::= \langle treatment\ scenario \rangle \xrightarrow{[\langle effect \rangle]} \langle threat \rangle \mid \langle risk \rangle \mid \langle asset \rangle\,; \\
\langle threat \rangle &::= \langle identifier \rangle\,; \\
\langle asset \rangle &::= \langle identifier \rangle\,; \\
\langle risk \rangle &::= (\langle identifier \rangle\,, \langle risk\ value \rangle); \\
\langle risk\ value \rangle &::= \langle qualitative \rangle \mid \langle quantitative \rangle \mid
\end{aligned}
$$

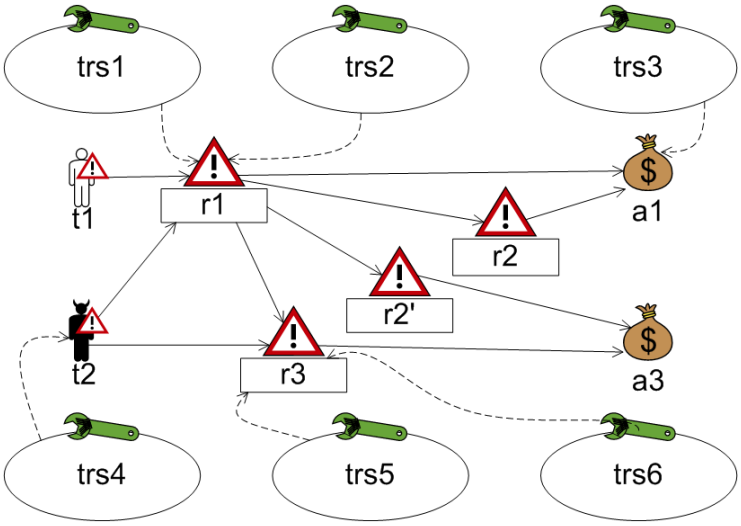

Figure 4.10: Treatment overview diagram

$$
\begin{aligned}
& \text{riskfunction}(\langle \mathit{consequence} \rangle, \langle \mathit{likelihood} \rangle); \\
\langle \mathit{treatment\ scenario} \rangle &::= \langle \mathit{identifier} \rangle\,; \\
\langle \mathit{effect} \rangle &::= \langle \mathit{qualitative} \rangle \,|\, \langle \mathit{quantitative} \rangle\,; \\
\langle \mathit{consequence} \rangle &::= \langle \mathit{qualitative} \rangle \,|\, \langle \mathit{quantitative} \rangle\,; \\
\langle \mathit{likelihood} \rangle &::= \langle \mathit{qualitative} \rangle \,|\, \langle \mathit{quantitative} \rangle\,; \\
\langle \mathit{qualitative} \rangle &::= \langle \mathit{identifier} \rangle\,; \\
\langle \mathit{quantitative} \rangle &::= \text{A number;} \\
\langle \mathit{identifier} \rangle &::= \text{A natural language description;}
\end{aligned}
$$

Semantics of Treatment Overview Diagrams

$$
\begin{aligned}
[\![t \xrightarrow{r} a]\!] &:= [\![t]\!] \text{ poses a } [\![r]\!] \text{ to } [\![a]\!]. \\
[\![trs \xrightarrow{e} t]\!] &:= [\![trs]\!] \text{ reduces the likelihood of } [\![t]\!] \\
&\quad \text{with an } [\![e]\!]. \\
[\![ts \xrightarrow{e} r]\!] &:= [\![trs]\!] \text{ reduces the risk value of } [\![r]\!] \\
&\quad \text{with an } [\![e]\!]. \\
[\![trs \xrightarrow{e} (rid, rf(l,c))]\!] &:= [\![trs]\!] \text{ reduces the } [\![l]\!] \text{ of } [\![(rid, rf(l,c))]\!] \\
&\quad \text{with an } [\![e]\!]. \\
[\![trs \xrightarrow{e} (rid, rf(l,c))]\!] &:= [\![trs]\!] \text{ reduces the } [\![c]\!] \text{ of } [\![(rid, rf(l,c))]\!] \\
&\quad \text{with an } [\![e]\!]. \\
[\![trs \xrightarrow{e} a]\!] &:= [\![trs]\!] \text{ reduces the importance of } [\![a]\!] \\
&\quad \text{with an } [\![e]\!]. \\
[\![t]\!] &:= \text{ threat } t.
\end{aligned}
$$

$$
\begin{aligned}
[\![r]\!] &:= \text{ risk } r. \\
[\![rv]\!] &:= \text{ risk value } rv. \\
[\![rf(l,c)]\!] &:= \text{ risk function } rf \text{ of } [\![l]\!] \text{ and } [\![c]\!]. \\
[\![(rid, rv)]\!] &:= \text{ risk } rid \text{ with } [\![rv]\!]. \\
[\![(rid, rf(l,c))]\!] &:= \text{ risk } rid \text{ with } [\![rf(l,c)]\!]. \\
[\![a]\!] &:= \text{ asset } a. \\
[\![trs]\!] &:= \text{ treatment scenario } trs. \\
[\![e]\!] &:= \text{ effect } e. \\
[\![l]\!] &:= \text{ likelihood } l. \\
[\![c]\!] &:= \text{ consequence } c.
\end{aligned}
$$

4.7 Conclusion

This chapter has given an introduction to the five core CORAS diagrams: assets overview, threat, risk overview, treatment and treatment overview diagrams. Using it as a reference, you should be able to construct your own diagrams and check whether a given diagram is correct or not. The textual syntax and semantics at the end of each section gives an exact interpretation of each diagram.

Chapter 5

Using CORAS to Assess Security in a Business Environment

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 [7].

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 [14]. The individual participants 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 [4]. 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, the

parties need to identify and address risks that may arise from the collaboration.

In order 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 non-technical aspects. The collaboration of different experts, like computer scientists and lawyers, is necessary when analysing what may go wrong in a co-operation [10,22]. 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, i.e. the organisations, processes and people which interact with the system.

The example case is based on a risk analysis which was performed in the TrustCoM IST project [29] using CORAS [9]. 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.

The remainder of this chapter is structured in accordance with the seven steps of the CORAS method for security analysis.

> As in Chapter 2, the example is separated from the rest of the text by being formatted like this paragraph.

5.1 Step 1: 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. The questions that should be answered at this initial point of time are of the following type:

- 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 of the analysis?

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 security 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 at some future point of time. 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 security analysis is necessary.

The introductory meeting should include the risk analysts (leader and secretary) and the client of the analysis. The latter is typically represented by a person with decision making powers with respect to the system or organisation being analysed, as well as system users and experts on the system or organizations such as IT personnel or lawyers. The meeting may also involve other parties or groups 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.

5.1.1 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, 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 (system integrator) 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 5.1 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).

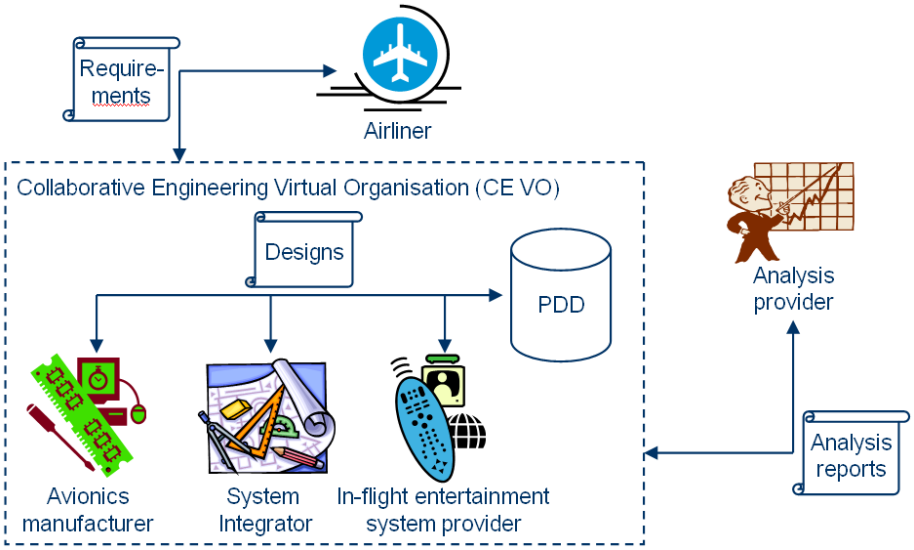

Figure 5.1: Actors in the CE scenario

5.1.2 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 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. To limit the size of the analysis, other aspects such as data integrity issues,

Role	Name	Organisation	Background/Expertise
Security analysis leader	Thomas	CORAS Ltd.	Security analysis, security
Security analysis secretary	Frank	CORAS Ltd.	Security 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 5.1: Risk analysis roles

e.g. malicious modification or deletion of information because of industrial sabotage or for example virus attacks, are left outside the scope of this particular analysis.

5.1.3 Plan the Analysis

Finally, the rest of the risk analysis should be planned in more detail, including identifying participants and meeting times and venues.

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 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 parties, domain experts, and other interested groups or persons 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 analysis team becomes large, it may be beneficial to split it into smaller groups during parts of the process, e.g. the during 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 guide and control the group.

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 5.1.

The security analysis for SI was performed over the course of ten weeks. Because the participants were spread across several countries, the main part of the analysis was performed during a two day

Date	Tasks	Participants
29 Nov	- Target identification - Asset identification	- Analysis leader - Analysis secretary - Legal expert
11 Jan	- High level analysis	- Analysis leader - Analysis secretary - Legal expert
27 Jan	- Approval - Risk identification	- Whole security analysis team
28 Jan	- Risk estimation and evaluation - Risk treatment	- Whole security analysis team
2 Feb	- Cleanup of results -Risk analysis report	- Analysis leader - Analysis secretary - Legal expert

Table 5.2: Risk analysis plan

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 5.2.

5.2 Step 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 security 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.

5.2.1 Risks 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 corrections 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 5.2.

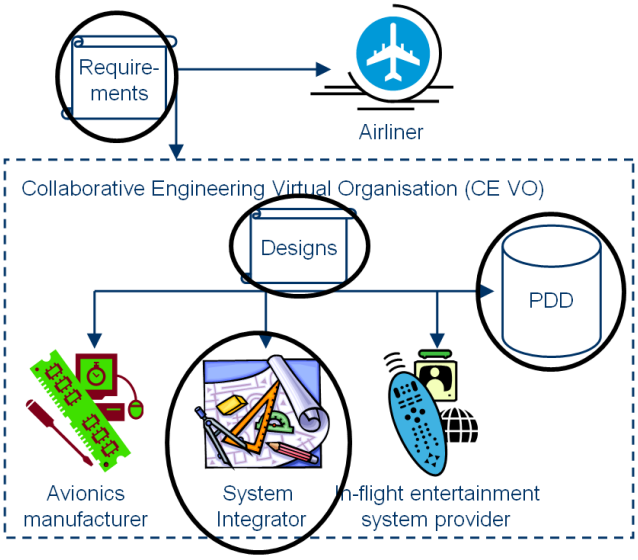

Figure 5.2: Target of analysis

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 5.3.

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 provision of services or the purchase of goods without establishing a consortium
- Service Level Agreements (SLAs), i.e. (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 5.4.

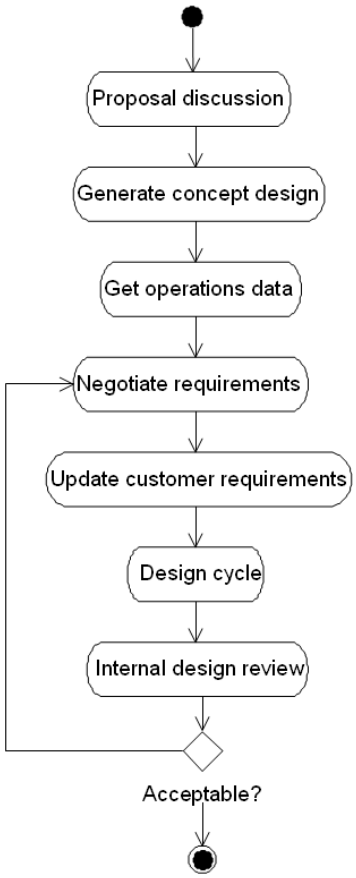

Figure 5.3: High-level CE VO design process

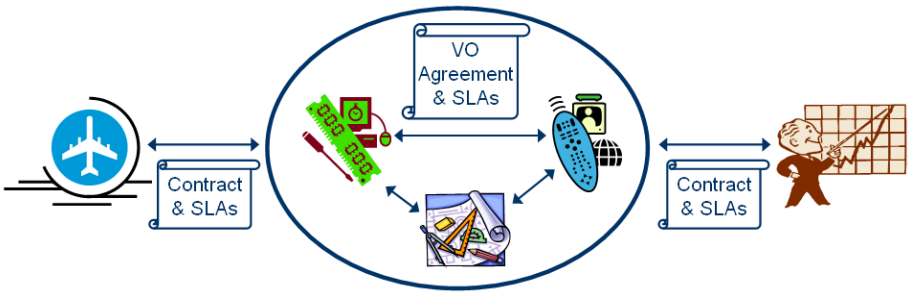

Figure 5.4: Contracts in the CE VO scenario

5.2.2 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.

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 4–6 most important assets suffice.

> As mentioned above 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 5.5.

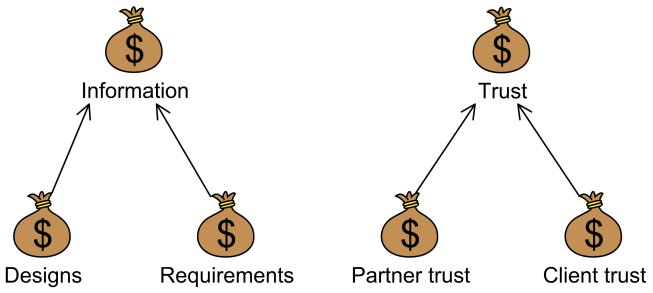

Figure 5.5: Asset diagram

5.2.3 High-Level Security Analysis

Sometimes it may be difficult to determine exactly what should and should not be included in the security 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 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.

Threat (deliberate) / Threat (accidental) / Threat (non-human)	Threat scenario / Unwanted incident / Asset	Vulnerability
Who/what causes it?	- How? - What is the incident? - What does it harm?	What makes this possible?
...	...	...

Table 5.3: High-level security analysis table

This high-level analysis can be seen as a first iteration of the security analysis. The results of the high-level analysis are documented in a table such as the one in Table 5.3.

5.3 Step 3: 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 high-level analysis. The resulting documentation should be sent to the client for perusal prior to the meeting.

5.3.1 Documentation of the 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 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's/customer's trust in SI	Other	Very high

Table 5.4: CE VO analysis asset table

5.3.2 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 5.4.

5.3.3 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, i.e. what are the chances of this risk occurring, and consequence, i.e. 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 and amount of money lost respectively. 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 and 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 5.5.

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 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 5.6. 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.

Value type	Value	
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

Table 5.5: Value definition table from CE VO analysis

		Consequence				
		Insignificant	Minor	Moderate	Major	Catastrophic
Likelihood	Rare					
	Unlikely					
	Possible					
	Likely					
	Certain					

Accept risk

Evaluate risk

Table 5.6: Risk matrix from CE VO analysis

5.4 Step 4: 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.

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.

5.4.1 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 pre-defined questionnaires and checklists.

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.

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 5.3 above 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 organisations or individuals. 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 5.6, 5.7 and 5.8 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.

Figure 5.6: Hacker steals designs and sells them to competitor

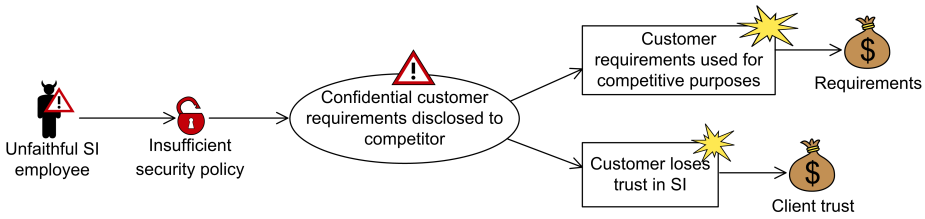

Figure 5.7: Unfaithful employee discloses customer requirements

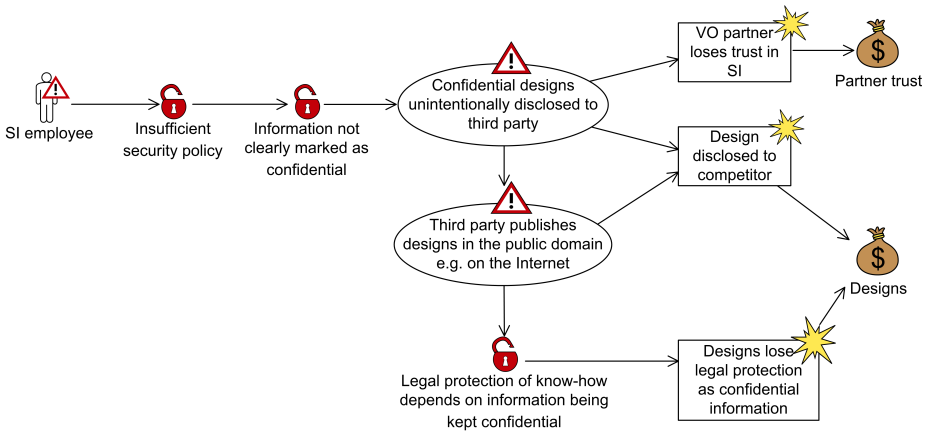

Figure 5.8: Loss of legal protection of know-how

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

Table 5.7: Consequence and likelihood table

5.5 Step 5: Risk Estimation

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 the risk evaluation described in the next section.

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 or historical data is available, more sophisticated methods may be used, for instance Fault Tree Analysis 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 5.7 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 5.9. 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").

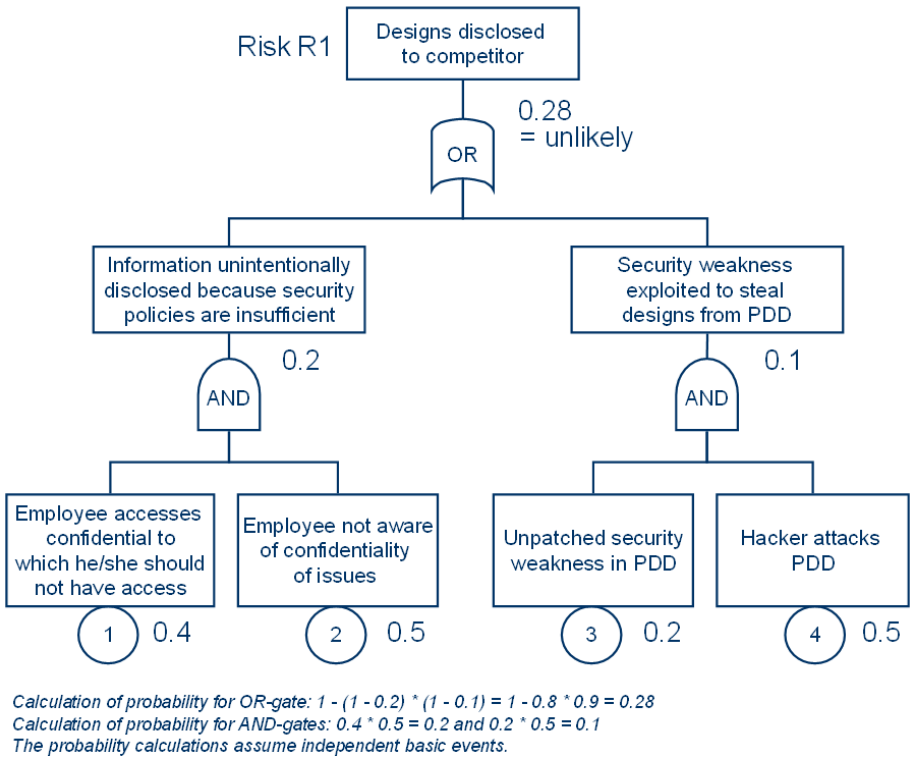

Figure 5.9: Fault tree

5.6 Step 6: Risk Evaluation

The goal of the risk evaluation is to prioritise the risks and identify which ones are in need of treatment by comparing against the risk evaluation criteria as established during 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, as this depends 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 5.10.

We then apply the risk evaluation criteria specified earlier during the approval meeting.

After estimating the likelihood and consequence of the risks, they are plotted into the pre-established risk matrix, as shown in Table 5.8.

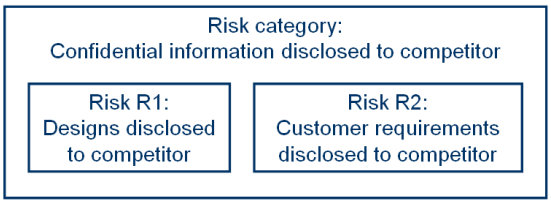

Figure 5.10: Risk category

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.

		Consequence				
		Insignificant	Minor	Moderate	Major	Catastrophic
Likelihood	Rare					
	Unlikely			**R1, R2**	**R3**	
	Possible			**R5**	**R4**	
	Likely					
	Certain					

Accept risk

Evaluate risk

Table 5.8: Risks plotted into risk matrix

5.7 Step 7: Risk Treatment

This phase aims at treating the non-acceptable risks by developing and implementing specific cost-effective strategies and action plans for reducing the risk level.

5.7.1 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
- Risk transfer, 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 5.11 and Figure 5.12. The figures show some threat scenarios from Figure 5.7 and Figure 5.8 and some options for treating them.

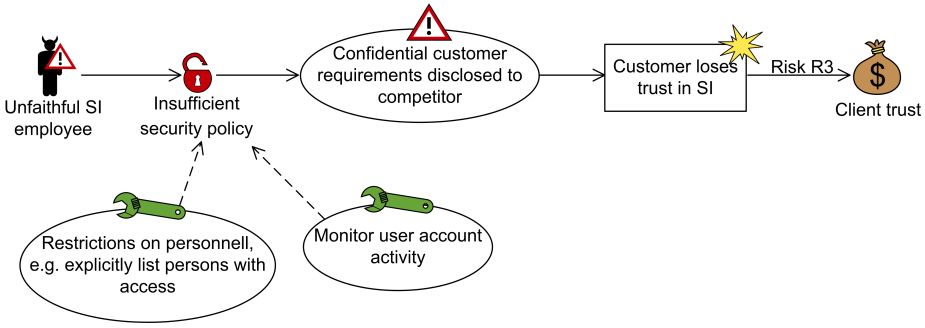

Figure 5.11: Treatments for risk R3

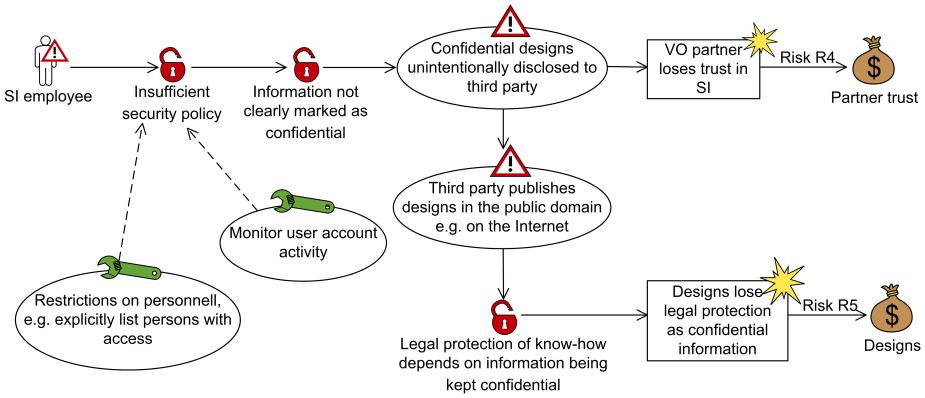

Figure 5.12: Treatments for risks R4 and R5

The aim in the CE VO analysis was to develop an integrated set of treatments, where legal and 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

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

Table 5.9: CE VO treatment evaluation

in the figures below, which reduce the likelihood that some of the vulnerabilities from Figure 5.7 and Figure 5.8 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.

5.7.2 Evaluate Treatments

To determine the best expenditure of the resources available for risk reducing measures, the identified 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 5.9 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.

5.8 Finalisation Meeting

For the risk analysis to have value, the findings of the risk analysis also need to be communicated to the relevant parties 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.

Chapter 6

Using CORAS in Component Oriented System Development

In this chapter, we propose an integrated process for component-based system development and security risk analysis. The integrated process is evaluated in a case study involving an instant messaging component for smart phones. We specify the risk behaviour and functional behaviour of components using the same kinds of description techniques. We represent main security risk analysis concepts, such as asset, party, threat and risk, at the component level.

6.1 Introduction

If the brakes on your car fail it might not be caused by sloppy work at the garage, but rather by a virus from the latest music file downloaded by your car's filesharing software. Well, maybe not today but this is not a far fetched scenario. There are cars on the market today running dynamic component platforms centralising a wide range of functionalities, such as entertainment functions, positioning systems and car controls. They allow continuous installation of new services or upgrades. With strict "time-to-market" requirements and short lifetime expectancies for software technology, products such as cars, smart phones and mobile devices in general are increasingly sold as upgradable products.

But how do we know if we can trust a new component that we download into a system? The increased application of component technology makes systems more flexible but also gives rise to new security concerns and calls for documentation of security risks at the component level. A component-oriented development process must reflect this.

The purpose of security risk analysis is to decide upon the necessary level of asset protection against security risks, such as a confidentiality or integrity breach. For convenience we will often use security analysis as a short term for security risk analysis.

We propose an integrated process for component-based system development and security analysis that builds on already existing techniques for specifica-

tion and security analysis. The CORAS method conveniently combines security analysis methods with UML-inspired systems modelling. CORAS has no particular support for component-oriented development. We believe, however, that UML models are also well suited to document and structure security analysis results at the component level. In CORAS and other current approaches to security analysis in system development, UML models are mainly used as input for identifying threats and structuring the security analysis process. We wish to use system development and modelling techniques such as UML not only as input for the security analysis, but also for the purpose of documenting analysis results, in order to support composition of such results.

We evaluate the integrated process in a case study involving an instant messaging (im) component for smart phones. The im component should allow users to interact in chat sessions and exchange media files with buddies, organised in a peer-to-peer fashion. It should be possible to deploy and run the service on smart phones, laptops etc. running a dynamic component platform.

The case we present is inspired by a tutorial on developing a chat service using OSGi (The Open Source Gateway initiative) - a standardised computing environment for networked services [30]. It is a fictitious example, but nevertheless represents a realistic case for component development that is of practical relevance. Due to lack of space we focus on a few selected parts that illustrate the characteristics of component-based security analysis and the structure of an integrated process. We refer to Brændeland and Stølen [2] for a presentation of the full evaluation. We evaluate the integrated approach with regard to the following criteria. In an integrated component-oriented process it should be possible to:

1. Identify parties holding assets at the component level.
2. Identify, value and represent assets at the component level.
3. Represent threats towards assets and analyse component behaviour at the component level, in relation to various threats.
4. Capture the notion of a risk at the component level in terms of impact and probability of an event. Furthermore it should be possible to:
5. Compose individual components documented according to 1–4, into composite components, and deduce the security risk level of the composition from the security risk documentation of its constituents.

This chapter addresses the intersection of security risk analysis and component-oriented development. In the next section the component specific notions are introduced. The evaluation of the integrated process is the structured into and presented over Section 6.4 through Section 6.7. These section scorrespond to the early stages in a component-based development process. Finally, in Section 6.8, we will conclude.

6.2 The Component Model

Figure 6.1 shows our conceptual model of a component. An interface is a contract with a client, describing a set of behaviours provided by a component object. It defines a list of operations, their signatures and semantics.

A component is a contract with the realiser. It describes provided interfaces and component dependencies in terms of required interfaces. By required interface we mean the calls the component needs to make, in order to implement the operations described in the provided interfaces. We distinguish between basic components and composite components. A basic component provides only one interface. We obtain components with more than one provided interface by the composition of basic components. Composite components can also be composed to obtain new composite components.

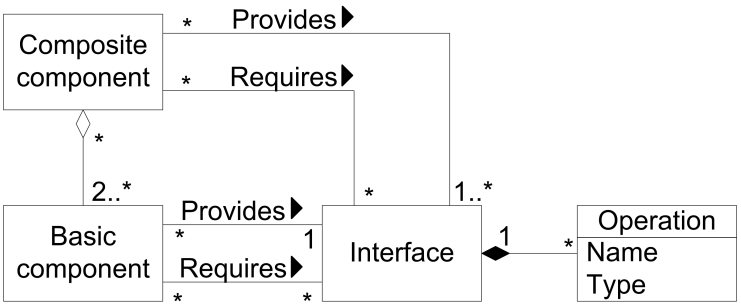

Figure 6.1: Conceptual model of a component

6.3 An Integrated Approach

In the following we investigate how the CORAS process can be integrated into a component-based system development process as described by Cheesman and Daniels [5]. They describe a process for specification and development of components using UML diagrams, illustrated by the left hand side of Figure 6.2. The grey boxes represent workflows as defined in the Rational Unified Process [19]. Each workflow produces an artefact that is used as input in the next workflow.

The component-oriented process starts by describing the overall system level requirements, such as functional requirements and quality of service requirements. During the requirements workflow the system is viewed as a black box, any internal behaviour of sub-components are hidden. According to [5], the requirements workflow should deliver a business concept model and a set of use cases to the specification workflow. During the specification workflow the system is decomposed into basic components that are refined further, independently of each other. It entails identifying provided and required interfaces, describing component dependencies and specifying how the basic components can be fitted together into a composite component that refines the original requirements. The output from the specification workflow is used in the provisioning workflow to determine what components to build or buy, in the assembly workflow to guide the correct integration of components, and in the test workflow as input to test scripts.

We aim, in particular, to integrate security analysis into the early stages of system development and focus, as indicated by Figure 6.2, on the requirements and specification workflow. While the requirements definition captures the qual-

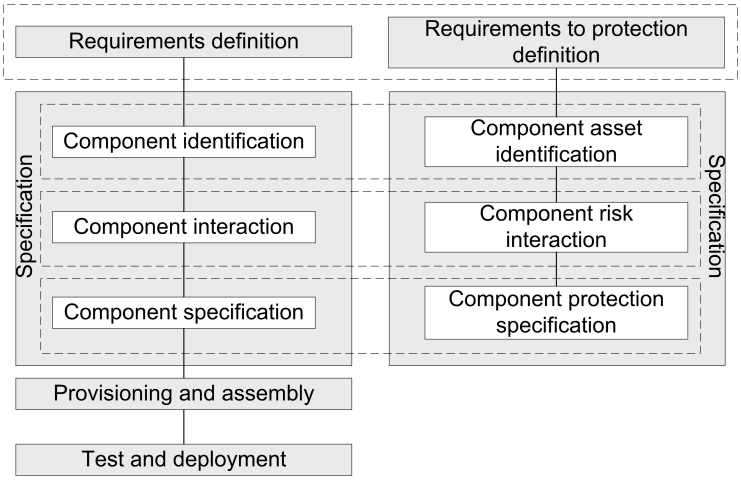

Figure 6.2: Integrated process

ity of service and functional requirements, the requirements to protection specify the acceptable level of security risk, i.e., what may be tolerated with respect to security risks, as further explained in Section 6.4. In component-based security analysis we need to describe parties and assets at the component level and in Section 6.5 we extend the component identification to achieve this. In Section 6.6 we augment the specification of the component interactions with specifications of the component risk interactions. Even if the basic component specifications is verified to refine the system requirements this of course does not mean that the requirements to protection are fulfilled. As explained further in Section 6.7 in addition to specifying the ordinary component logic we must also characterise its way of protection.

We use a combination of UML 2.0 notation [26] and CORAS diagrams in the component specifications.

6.4 Requirements

In accordance with Figure 6.2 we first give the requirements definition as described by Cheesman and Daniels [5]. Then we present the requirements to protection definition.

6.4.1 Requirements Definition

The purpose of requirements definition is to describe what services the system should provide, allocate responsibilities for the various services and to decide upon the system boundary [5]. We use a UML use case diagram shown in Figure 6.3 to clarify the software boundary, identify the actors interacting with the system, and list those interactions. A use case diagram is usually used in combination with a textual description capturing the details of each use case. We can also specify details of the use cases in sequence diagrams. The diagram to the right in Figure 6.4 captures the details of the Startup use case. When a

user starts the im client, the client searches for other peers. After starting the client, the user can register or login. Registration is specified as optional since this happens only the first time a user launches an im component. If a login is successful the im service is registered.

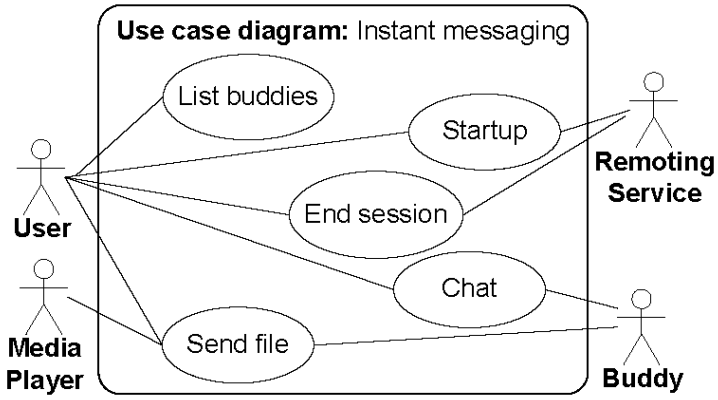

Figure 6.3: Use case diagram

To the left in Figure 6.4 is an interaction overview diagram showing the flow of interactions of the im component. After startup the user can either check for other buddies, send messages or send files.

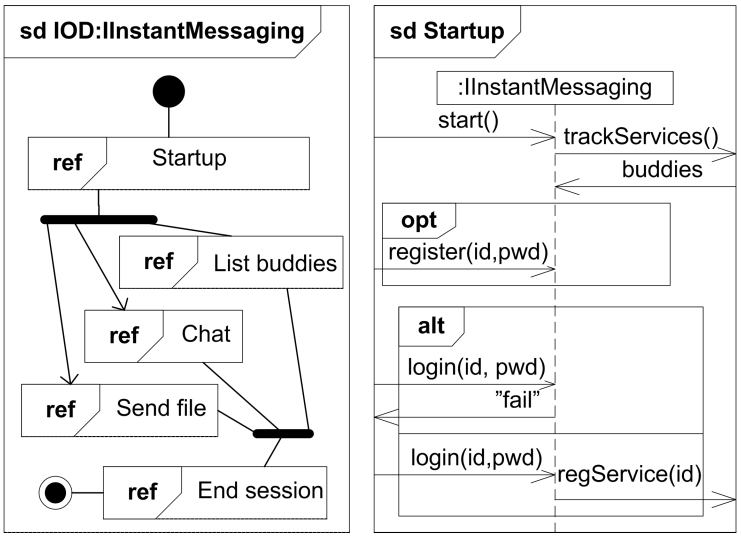

Figure 6.4: Interactions of the im component

6.4.2 Requirements to Protection Definition

The requirements to protection definition specify what a given party may tolerate with respect to risk towards an asset. Within the CORAS method, these requirements are referred to as risk evaluation criteria. Prior to defining the risk evaluation criteria it is necessary to identify the relevant parties, identify the assets held by these parties, and finally valuate the identified assets.

The use case diagram in Figure 6.3 serves as input to the identification of the relevant parties and their assets. In a use case the actors interact with the system as a whole, not some specific part of it. The actors identified in a use-case correspond to component clients or suppliers, represented through provided and required interfaces.

Figure 6.5 show examples of identified assets. We identify UserId, MsgContent and Efficiency as assets of the actor User and MediaFile as an asset of the MediaPlayer. The UserId may be of value for the User if she employs the same id for many different services and access to the UserId could provide access to these. We have for simplicity chosen to annotate the asset diagram with the respective asset values.

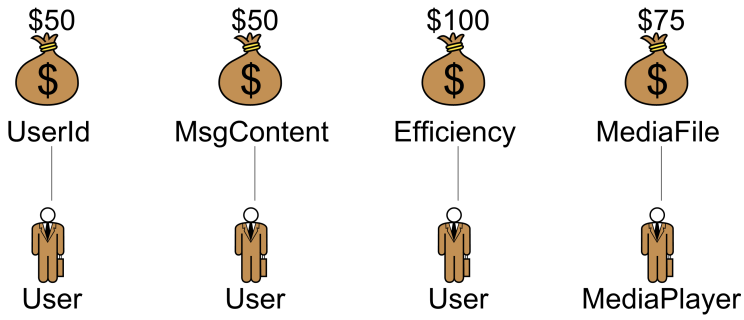

Figure 6.5: Asset identification

We can now for each party define risk evaluation criteria for each of the party's assets. A risk evaluation criterion may for example be that "The probability of a risk reducing the MediaFile asset with 30 dollars should not be higher than 0.25."

6.5 Components and their Assets

In accordance with Figure 6.2, we first conduct component identification, thereafter we identify the assets of each component.

6.5.1 Component Identification

Component identification is the first stage of the specification workflow. It entails decomposing the system into basic components and identifying component interfaces. A use case indicates the types of operations that the system should offer through interfaces. We group the responsibility for the Startup, Chat, List

buddies and End session and assign it to one interface that we call IChat, as illustrated in Figure 6.6[1]. The responsibility for the Send file use case is assigned to a separate interface IFileTransfer. We identify an interface IChannel that a Buddy can use for sending messages or files to a client.

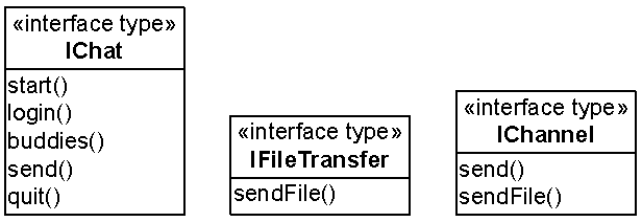

Figure 6.6: Interface types

We must also identify business interfaces, if any, i.e., the interfaces handling information managed by the system. We assume that we have already defined a business concept model and refined this into a business type model providing the necessary input for this task. The only type of information that the im component is itself responsible for managing, is the UserId. Hence, we identify one business interface IUserMgr. Other types of information will be handled by external components.

We also need to establish whether there are existing interfaces and systems that are part of the environment into which the im service will be deployed. As illustrated in Figure 6.7 the im component requires an interface for remote services, such as registration and tracking, and interfaces for displaying messages and playing music files. Buddies registered through the IRemote interface will be other Channels providing interfaces that the Chat and FileTransfer components employs for sending messages and files, respectively. We indicate component dependencies using the ball and socket notation of UML component diagrams. The component dependencies are detailed further as part of the component interaction specification.

6.5.2 Component Asset Identification

During the requirements workflow we identified system level parties and assets. We now decompose and identify parties and assets at the level of basic components. The actor MediaPlayer corresponds to the provider of the IMediaPlayer interface and the actor RemotingService corresponds to the provider of the IRemote interface. The actor User corresponds to the providers and clients of the external interfaces: IFileTransfer, IChat and IDisplay. The assets UserId and MsgContent can be affected by operations of the IChat interface and we assign these assets to the client of that interface. We assign the Efficiency asset to the provider of the IDisplay interface which the Channel component requires to convey received messages.

[1]Cheesman and Daniels [5] introduce the stereotypes ≪interface type≫ and ≪comp spec≫, rather than applying the predefined UML modelling elements ≪interface≫ and ≪component≫, which are used for modelling the implementation level.

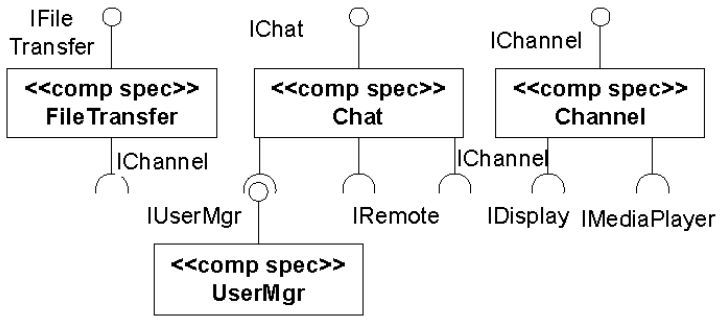

Figure 6.7: Basic components

Above we identified the component UserMgr which provides an interface that the Chat component employs to handle user information. Hence the UserMgr and Chat components are parties of each other, i.e. the UserMgr is a party with respect to the Chat component and vice versa. We assign an asset UserId to the requirer of the UserMgr interface, i.e. the Chat component. Since the UserId assets are held by different components and parties, they should be considered as two different assets. This is because they may be valuated differently by different parties.

6.6 Interactions

As indicated by Figure 6.2, there are two kinds of interactions to be specified. We first describe the wanted interactions and then the risk interactions.

6.6.1 Component Interaction

We now have to decide how the various sub-components work together in order to deliver the required functionality. This entails specifying which operations the Chat component calls from the IUserMgr interface in order to perform the login operation and so on. For example, as illustrated in Figure 6.8 when sendFile is called, the IChannel component calls a method to play the music file. As explained below, this is not without risk.

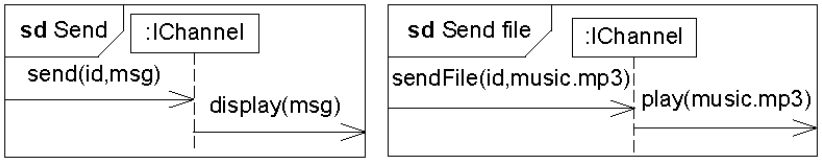

Figure 6.8: Interactions of the channel component

Interface behaviour can be seen as an abstraction over basic component interactions. A provided interface represent the view of the user, who does

not need to know how an operation is implemented. Formally we obtain the provided interface of a basic component by filtering away the interactions on the required interface [3].

6.6.2 Component Risk Interaction

In order to describe the risk interactions, we first identify risks and estimate their risk levels. This is done following step 4 and 5 of the CORAS method. We use the system specification diagrams as input to the structured brainstorming session. The result is documented in CORAS threat diagrams as the one in Figure 6.9. The vulnerability "Accept not required" refers to the fact that the im component automatically plays music without consulting the user. This may be exploited to send a specially crafted music file designed to exploit possible buffer overflow vulnerabilities in a media player. This diagram shows that the occurrence of "Send crafted file" initiates the threat scenario "Play crafted file". The information security incident "Receive malicious code" affects to the asset "MediaFile".

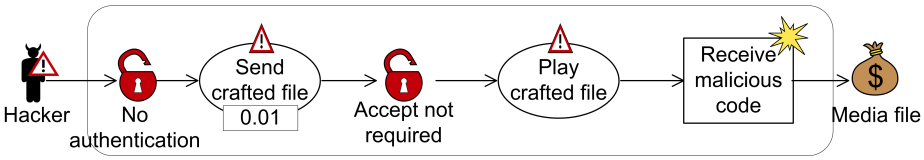

Figure 6.9: Threat related to file transfer

To determine the risk level of the identified risks we estimate their consequence and probability values. Threat diagrams are used as input to brainstorming sessions aiming to calculate impacts and probabilities of risks. In order to estimate the probability of a hacker successfully sending a crafted media file we must know both the probability of an attack and the probability of the success of an attack (degree of vulnerability), given an attempt. The probability of a single threat scenario can for example be estimated from historical data. While the number of mobile device threats continues to increase, the number of reported threats is still relatively small [28]. We therefore estimate the probability of an attack to be fairly low (between 0 and 0.01).

In order to estimate the success of an attack we must consider vulnerabilities and the effectiveness of control mechanisms if any such exists. As the IChannel component is not designed to check validity of file names and automatically opens incoming music files, the probability that the IChannel component attempts to play a crafted file is 1. We multiply the probability of an attack with the probability of its success to obtain the probability of the unwanted incident. Hence the probability that the IMediaPlayer receives a crafted file is the same as the probability of the attack.

The risk interaction can be specified more detailed using UML sequence diagrams. In a component setting we represent threats as lifelines. A threat may initiate a threat scenario by calling an operation of one of the components interfaces. Figure 6.10 shows the interactions of a hacker sending a crafted mp3 file and the interactions of the IChat receiving it.

We represent an asset as a lifeline that has a value. In an interaction we represent the reduction of asset value by a special message called reduce, which takes as argument the amount by which the asset value should be reduced, see Figure 6.10. As the sequence diagram illustrates an example run, we assume the initial asset value have been set elsewhere and parameterise the specification with the asset value k of type $. The value of an asset at a given point in time is computed from its initial value and all occurrences of reduce, with the asset as receiver, up to that point. The value of an asset can not go below zero.

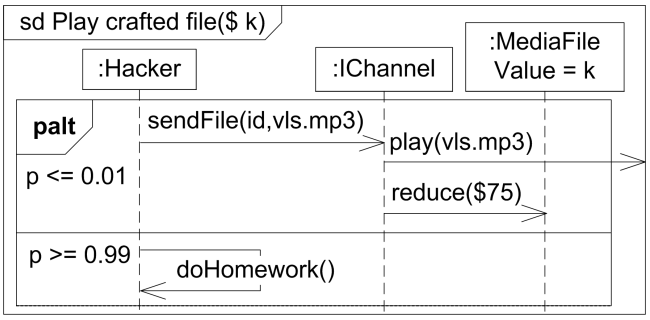

Figure 6.10: A hacker sends a crafted mp3 file

A risk is measured in terms of its probability and its impact. The manifestation of a risk in an interaction is the transmission of a reduce event to an asset. In order to decide the probability of a risk one must know the combined probability of the interactions leading to the transmission of the reduce message. We use the palt operator of probabilistic STAIRS [25]] to specify probabilistic choice, as illustrated in Figure 6.10.

palt is a combined fragment with multiple operands. Each operand is annotated with a probability value specifying the probability for the interactions of this operand to execute when the combined fragment is executed. Only one of the operands is executed in one execution of the combined fragment, and the totality of the probabilities of the operands always add up to 1.

6.7 Specification

During the final stage of the specification workflow, we finalise the component specifications and verify that the requirements, including the requirements to protection, are met by the component specifications. In accordance with Figure 6.2, we first describe what this stage entails with regard to component specification and then with regard to the protection specification.

6.7.1 Component Specification

Component specification is the final stage of the specification workflow. During this stage we describe how the basic components can be fitted together into a composite component that refines the original requirements. In a composite component we hide local interactions, so the interactions between the Chat and UserMgr sub-components become hidden.

6.7.2 Component Protection Specification

We must also combine the component security risk specifications using the composition operators of probabilistic STAIRS to check whether their composition fulfils the requirements to protection, i.e. the risk acceptance criteria given for each asset. We refer to [3] for the full details of composite security risk specifications. If the identified risks are found to violate the risk acceptance criteria, we need to identify protection mechanisms and alter the specification of the components accordingly, to ensure that the requirements to protection are met.

In order to reduce the probability of the risk we presented for the Channel component, we can remove the behaviour where an incoming file is automatically opened and supplement with alternative behaviour requiring that the user should be consulted before opening an incoming file. The new specification constitutes the component protection specification.

6.8 Summary and conclusion

We have presented a case-based evaluation of the feasibility of applying model-based security analysis in order to integrate security analysis in component-oriented development. We claim that the integrated approach presented here meets the requirements of Section 6.1 in the following sense:

1. By limiting the notion of a party to that of a component client or supplier, we manage to identify and represent parties at the component level

2. We identify assets for each party and specify them as lifelines with a value. We represent the reduction of asset value as the reception of a special kind of reduce message.

3. We represent threats as lifelines that may interact with a component and initiate a threat scenario.

4. We represent a risk as a probabilistic interaction leading to the reduction of an asset value.

5. The component security analysis results are documented using sequence diagrams augmented with palt, a combined fragment for specifying probability. We obtain the risk level of a composite component by composing the security risk specifications of its subcomponents, using composition.

Chapter 7

The CORAS Tool

The CORAS Tool for model-based security risk analysis supports documentation and reuse of risk analysis results through integration of different risk analysis and software development techniques and tools. Built-in consistency checking facilitates the maintenance of the results as the target of analysis and risk analysis results evolve.

7.1 Introduction

The CORAS framework for UML-based security risk analysis, in the following referred to as security analysis, consists of among other things a methodology, a language, and a tool. The CORAS methodology integrates aspects from partly complementary risk analysis techniques, like HazOp [24], FMEA [1], and FTA [12], with state-of-the-art system modeling methodology based on UML 2.0 [23]. A graphical UML-based language has been developed to support documentation and communication of security analysis results [20].

The integration of different risk analysis approaches facilitates the analysis of different aspects of a system or organisation, e.g. security, legal issues and business processes, resulting in a combined picture of the relevant risks. For example, the notion of trust is tightly interwoven with notions like security and usability [18, 8]. Furthermore, it is difficult to separate trust from the expectation of a legal framework that offers protection in the cases where the trust relationship fails [17]. An analysis of trust should therefore encompass a number of issues including technological, legal, sociological and psychological aspects.

This chapter presents the CORAS Tool which has been developed to provide computerized support for performing security analysis in accordance with the CORAS methodology using the graphical CORAS language.

7.2 The CORAS Tool

Security analysis is a highly elaborate and prolific process which involves many different types of documentation, such as UML models, tables with analysis data, and natural language descriptions of the target of evaluation. Access to this information and its change history needs to be provided. The information

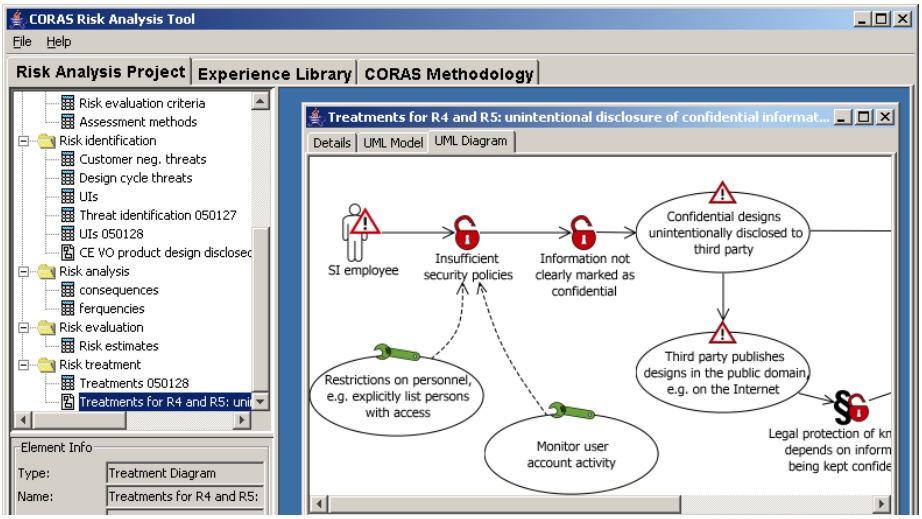

Figure 7.1: Screenshot of the CORAS Tool

also needs to be shared between the various modelling and analysis tools used by the analyst. In addition, it is important to maintain consistency between all the different pieces of information. Computerised support for documentation, maintenance and reuse of analysis results is thus of high importance.

The CORAS Tool has been developed based on the principles and requirements outlined above and is publicly available as open source [6]. The tool follows a client-server model and is developed entirely in Java. The security analyst uses the CORAS client application to create new analysis projects, document and edit security analysis results, generate analysis reports, and manage and reuse experiences from previous analyses. Help is provided to the user in the form of integrated electronic versions of the CORAS methodology as well as user guides. A screenshot of the CORAS client application is shown in Figure 7.1.

Analysis data is stored in a centralised database on the server, enabling multiple users to collaborate on the same analysis project. The server is based on standard Enterprise Java Beans (EJB) technology and runs on top of the open source JBoss application server. Figure 7.2 shows an overview of the structure of the CORAS tool.

There are two databases in the CORAS Tool, called repositories. The assessment repository stores results from the actual security analyses, while the experience repository contains reusable results from previous analyses in the form of UML-models, checklists, procedures and more. By facilitating reuse, the tool helps the user avoid starting from scratch for each new analysis. The repositories are versioned, so that all changes are maintained and the user can go back and look at the state of a security analysis at a previous point in time. The repositories are implemented on top of the open-source XML database eXist [21].

A wide variety of UML modelling tools and security analysis tools exist and are in use by security analysts and system engineers today. It is therefore important for our tool to provide flexible support for integration with external

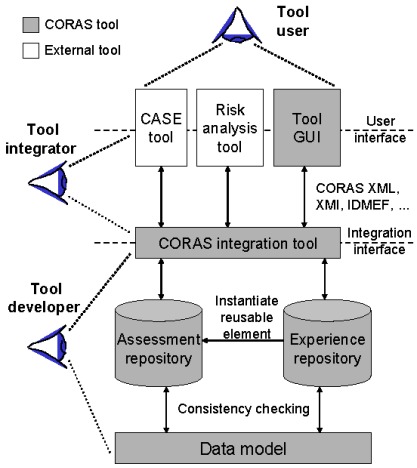

Figure 7.2: Overview of the CORAS Tool

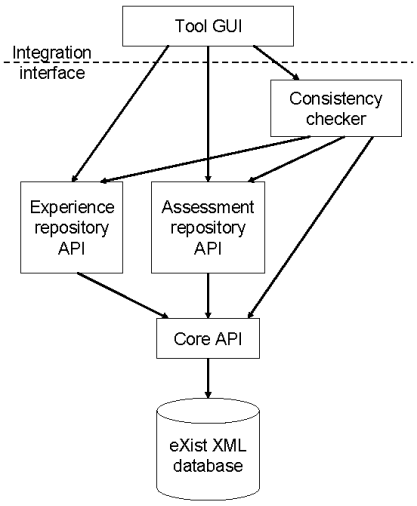

Figure 7.2: Integration interface

tools. Though the security analysis process is model-driven, the tool does not deal exclusively with UML models, but must also be able to include tables and other constructs such as fault tree diagrams, intrusion detection logs and natural language descriptions. To satisfy these requirements, the tool provides an integration layer with a defined API which can be used by other tools to integrate with the CORAS tool. Standardised XML formats are utilised for data integration, such as XMI for the interchange of UML models.

Three different viewpoints or roles are defined in Figure 7.2 "Overview of the CORAS Tool". The tool user represents the end-user, e.g. a security analyst who uses the CORAS tool together with various modelling and security analysis tools The tool integrator, on the other hand, is responsible for integrating the various external tools with the CORAS tool, using the integration interface. Finally, the tool developer will implement the functionality of the CORAS tool itself.

Many entities of a security analysis, e.g. assets, parties and threats, appear multiple times throughout the analysis results. It is therefore important to ensure that the various analysis results are mutually consistent with each other. We have defined an internal risk analysis data model based on the concepts and relationships defined in the CORAS language [20]. This model extracts information from the analysis data and uses this as a basis for the consistency checking. Consistency rules are defined using the XML Stylesheet Transformation Language (XSLT). Figure 7.2 "Integration interface" shows how the main components of the tool interact.

Appendix A

The CORAS Guideline

Throughout this book, we have presented examples of how CORAS is used for risk analysis. In addition to using CORAS diagrams, these examples also have a common *method* centred around the seven steps first introduced in Chapter 2. In this appendix we have summarised the essential points of the CORAS method, presenting a table for each step with tasks, people that should participate and modelling guidelines if appropriate.

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: – Decision makers (required) – Technical expertise (optional) – Users (optional)
Modelling 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 modelling techniques such as UML or data flow-diagram.	

Table A.2: Step 1: Introductory meeting

Tasks:	**People that should participate:**
• 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 guideline:
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 how 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)

Table A.4: Step 2: High-level analysis

Tasks:	**People that should participate:**
• 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 each asset within the scope of the analysis.	• 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.

Table A.6: Step 3: Approval

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[1]) – Technical expertise (required) – Users (required)

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. ([11] contains a useful classification). This makes it easier to generalise over the risks, e.g. "these risks are caused by deliberate intruders" or "these risks are caused by human errors".
3. Assets are listed to the right, outside the region.
4. Threats are placed to the left in the region; threats that can be classified as external (hackers, intruders etc.) are placed outside the region.
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 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".

Table A.8: Step 4: Risk identification

Tasks:	**People that should participate:**
• 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: – Decision makers (required) – Technical expertise regarding the target (required) – 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.

Table A.10: Step 5: Risk estimation

Tasks:	**People that should participate:**
• 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[2]) – Users (required/optional[2])

Modelling guideline:
Risk diagrams:

1. Use the threat diagram and replace all unwanted incidents with risk symbols, showing a short risk 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.
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 related to network, all risks for specific assets etc.).

Table A.12: Step 6: Risk evaluation

Tasks:	**People that should participate:**
• 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 guideline:
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).

Table A.14: Step 7: Risk treatment

Bibliography

[1] Abdelkader Bouti and Daoud Ait Kadi. A state-of-the-art review of FMEA/FMECA. International Journal of Reliability, Quality and Safety Engineering, 1(4):515–543, 1994.

[2] Gyrd Brændeland and Ketil Stølen. Using model-based security analysis in component-oriented system development. a case-based evaluation. Technical Report 342, University of Oslo, Department of Informatics, 2006.

[3] Gyrd Brændeland and Ketil Stølen. A semantic paradigm for component-based specification integrating a notion of security risk. Technical Report 341, University of Oslo, Department of Informatics, 2006.

[4] Luis M. Camarinha–Matos, Hamideh Afsarmanesh, Hermann Löh, Flavius Sturm, and Martin Ollus. A strategic roadmap for advanced virtual organizations. Springer, 2004.

[5] John Cheesman and John Daniels. UML Components. A simple process for specifying component-based software. Component software series. Addison-Wesley, 2001.

[6] CORAS. The CORAS project, 2006. `http://coras.sourceforge.net/`.

[7] Theo Dimitrakos, David Golby, and Paul Kearney. Towards a Trust and Contract Management Framework for dynamic Virtual Organizations. In Proceedings of eAdoptions and the Knowledge Economy: eChallenges 2004. IOS press, 2004.

[8] Florian N. Egger. Towards a model of trust for e-commerce system design. In Proc. of the CHI 2000 Workshop: Designing Interactive Systems for 1-to-1 E-commerce, 2000.

[9] Tobias Mahler et al. Report on Legal Issues. Technical Report D15 WP9, TrustCoM IST Project 01945, 2005.

[10] Thomas Heymann. Outsourcing in Deutschland-eine Bestandsaufnahme zu Vertragsgestaltung. Die Grundtypen des Outsourcing und ihre Konsequenzen für die Vertragsgestaltung. Computer und Recht, 21(10):706–710, 2005.

[11] Ida Hogganvik and Ketil Stølen. Risk Analysis Terminology for IT-systems: Does it match intuition? In Proceedings of the Fourth International Symposium on Empirical Software Engineering (ISESE'05), pages 13–23. IEEE Computer Society, 2005.

[12] IEC. Fault Tree Analysis (FTA), 1990. IEC 1025.

[13] ISO/IEC. Information technology – Code of practice for information security management. 17799.

[14] ISO/IEC. Information Technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management, 2004. TR 13335–1.

[15] ISO/IEC 10746. Basic reference model for open distributed processing, 1995.

[16] ISO/IEC13335. Information technology - Guidelines for management of IT Security, 1996-2000. http://www.iso.ch.

[17] Sara Jones, Marc Wilikens, Philip Morris, and Marcelo Masera. Trust requirements in e-business. Commun. ACM, 43(12):81–87, 2000.

[18] Audun Jøsang, Roslan Ismail, and Colin Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems (to appear), 2006.

[19] Philippe Kruchten, editor. The rational unified process. An introduction. Addison-Wesley, 2004.

[20] Mass Soldal Lund, Ida Hogganvik, Fredrik Seehusen, and Ketil Stølen. UML profile for security assessment. Technical Report STF40 A03066, SINTEF Telecom and informatics, 2003.

[21] Wolfgang Meier. eXist: An Open Source Native XML Database. In A. Chaudri, M. Jeckle, E. Rahm, and R. Unland, editors, Web, Web-Services, and Database Systems, volume 2593 of LNCS, pages 169–183. Springer, 2002.

[22] Claus D. Müller-Hengstenberg. Der Vertrag als Mittel des Risikomanagements. Computer und Recht, 21(5):385–392, 2005.

[23] Object Management Group. UML 2.0 Superstructure Final Adopted specification, October 2004.

[24] Felix Redmill, Morris Chudleigh, and James Catmir. System safety: HazOp and software HazOp. Wiley, 1999.

[25] Atle Refsdal, Knut Eilif Husa, and Ketil Stølen. Specification and refinement of soft real-time requirements using sequence diagrams. In FORMATS, volume 3829 of Lecture Notes in Computer Science, pages 32–48. Springer, 2005.

[26] James Rumbaugh, Ivar Jacobsen, and Grady Booch. The unified modeling language reference manual. Addison-Wesley, 2005.

[27] Standards Australia, Standards New Zealand. Australian/New Zealand Standard. Risk Management, 2004. AS/NZS 4360:2004.

[28] Symantec. Symantec internet security threat report. Trends for July 05–December 05, March 2006.

[29] TrustCoM. The TrustCoM IST Project. http://www.eu-trustcom.com/.

[30] Thomas Watson and Peter Kriens. OSGi component programming. Tutorial held at Eclipsecon 2006, 2006.