Model-based Risk Assessment to Improve Enterprise Security

Jan Øyvind Aagedal[*], Folker den Braber[*], Theo Dimitrakos[§],
Bjørn Axel Gran[#], Dimitris Raptis[‡], Ketil Stølen[*]
[*]*SINTEF Telecom and Informatics, P.O.Box 124, Blindern, N-0314 Oslo, Norway*
[§] *CLRC Rutherford Appleton Laboratory, Oxfordshire, OX11 0QX, UK*
[#]*Institute for Energy Technology, P.O. Box 173, N-1751 Halden, Norway*
[‡]*INTRACOM, 19.5 Km Markopoulou Av., GR-19002, Peania Athens, Greece*
{Jan.Aagedal | Folker.den.Braber | Ketil.Stoelen}@sintef.no
T.Dimitrakos@rl.ac.uk, bjorn.axel.gran@hrp.no, drap@intracom.gr

Abstract

The main objective of the CORAS project is to provide methods and tools for precise, unambiguous, and efficient risk assessment of security critical systems. To this end, we advocate a model-based approach to risk assessment, and this paper attempts to define the required models for this.

Whereas traditional risk assessment is performed without any formal description of the target of evaluation or results of the risk assessment, CORAS aims to provide a well defined set of models well suited to (1) describe the target of assessment at the right level of abstraction, (2) as a medium for communication between different groups of stakeholders involved in a risk assessment, and (3) to document risk assessment results and the assumptions on which these results depend.

We propose here models for each step in a risk assessment process and report results of use.

1. Introduction

CORAS [1] is a research and development project under the European Information Society Technologies Programme. CORAS started in January 2001 and runs until July 2003. The consortium consists of three commercial companies: Intracom (Greece), Solinet (Germany) and Telenor (Norway); seven research institutes: CTI and FORTH (Greece), IFE, NCT, NR, and Sintef (Norway) and RAL (UK); as well as one university: QMUL (UK). Telenor and Sintef are administrative and scientific co-ordinators, respectively.

CORAS aims to produce an improved methodology for precise, unambiguous, and efficient risk analysis of security critical systems. The focus of the CORAS project is on the tight integration of viewpoint-oriented modelling in the risk assessment process. An important aspect of the CORAS project is the practical use of UML [2] in the context of security and risk assessment.

CORAS addresses security-critical systems in general, but puts particular emphasis on IT security. IT security includes all aspects related to defining, achieving, and maintaining confidentiality, integrity, availability, non-repudiation, accountability, authenticity, and reliability of IT systems [3]. An IT system in the sense of CORAS is not just technology, but also the humans interacting with the technology and all relevant aspects of the surrounding organisation and society.

The remainder of this paper is structured as follows. Section 2 provides background information on risk assessment and modelling. Section 3 presents the model-based risk assessment process and introduces the models that should be created as part of the model-based risk assessment process. Section 4 illustrates how the model-based risk assessment can be used by an e-commerce case. Finally, section 5 points to related work while section 6 summarises our results and identifies future work.

2. Background

In this section, we briefly present relevant background information on risk assessment and modelling.

2.1. Risk assessment

Risk assessment incorporates risk analysis and risk management, i.e., it combines systematic processes for risk identification and determination of their consequences, and how to deal with these risks. Many risk assessment methodologies exist, focussing on different types of risks or different areas of concern. The CORAS methodology builds on: HAZard and OPerability study (HazOp); Fault Tree Analysis (FTA); Failure Mode and Effect Criticality Analysis (FMECA); Markov analysis (Markov); CCTA Risk Analysis and Management Methodology (CRAMM).

The methods are to a great extent complementary. They address all types of risks associated with the target system. They also cover all phases in the system

Proceedings of the Sixth International ENTERPRISE DISTRIBUTED OBJECT COMPUTING Conference (EDOC'02)
0-7695-1656-4/02 $17.00 © 2002 IEEE

development and maintenance process. In general, qualitative methodologies for analysing risk are effective in identifying threats and failures in trust within the system, but they lack the ability to account for the dependencies between events. Tree-based techniques, however, take into consideration the dependencies between events. Risk assessment is generally accompanied by volumes of documents where attempting to find relationships and links is difficult.

2.1.1. Process

The Australian/New Zealand standard AS/NZS [4] is a widely recognised standard within the field of risk assessment. Figure 1 shows an overview of the risk assessment process in this standard. In CORAS, we use this process to position models within risk assessment.

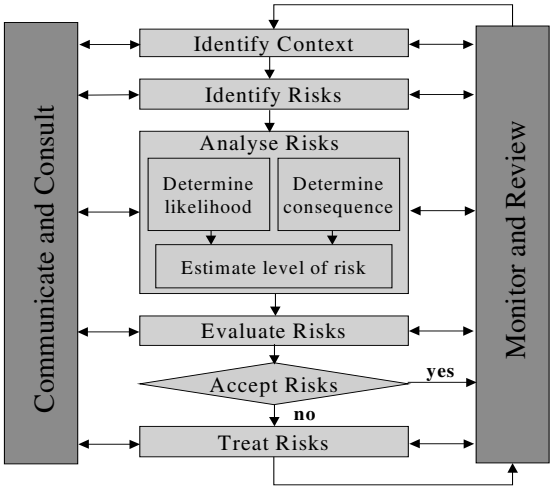

Figure 1. Risk assessment overview [4]

2.2. Modelling

Reference Model for Open Distributed Processing (RM-ODP) [5] is a standard reference model for distributed systems, based on object-orientation. RM-ODP divides the system documentation into five viewpoints. It also provides modelling, specification and structuring terminology, a conformance module addressing implementation and consistency requirements, as well as a distribution module defining transparencies and functions required to realise these transparencies.

UML is the de facto standard for documenting software architectures. However, UML is a large language and its use in different phases of system evolution is not standardised. In this paper we show how UML can be used to document both the target of risk assessment, and the results of such an assessment.

3. Model-based risk assessment

In this section we present the CORAS approach to risk assessment.

3.1. Motivation

CORAS focuses on the integration of viewpoint-oriented modelling in the risk assessment process. The integration of this state-of-the-art modelling technology in the risk assessment process, in the following referred to as model-based risk assessment, is motivated by several factors. Model-based risk assessment employs modelling technology for three main purposes:

1. Providing descriptions of the target of assessment at the right level of abstraction.
2. As a medium for communication and interaction between different groups of stakeholders involved in a risk analysis.
3. To document results and the assumptions on which these results depend.

Model-based risk assessment is motivated by several factors:

- Risk assessment requires correct descriptions of the target system, its context and all security features. The modelling technology improves the precision of such descriptions. Improved precision is expected to improve the quality of risk assessment results.
- The graphical style of UML furthers communication and interaction between stakeholders involved in a risk assessment. This is expected to improve the quality of results, and also speed up the risk analysis process since the danger of wasting time and resources on misconceptions is reduced.
- The modelling technology facilitates a more precise documentation of risk assessment results and the assumptions on which their validity depend. This is expected to reduce maintenance costs by increasing the possibilities for reuse.
- The modelling technology provides a solid basis for the integration of assessment methods that should improve the effectiveness of the assessment process.
- The modelling technology is supported by a rich set of tools from which the risk analysis may benefit. This may improve quality (as in the case of the two first bullets) and reduce costs (as in the case of the second bullet). It also furthers productivity and maintenance.
- The modelling technology provides a basis for tighter integration of risk management in the system development process. This may considerably reduce development costs and ensure that the specified security level is achieved.

3.2. CORAS framework

The main CORAS deliverable will be the CORAS framework. As indicated by Figure 2, the CORAS framework focuses on model-based risk assessment. The framework has four main pillars, a system documentation framework based on RM-ODP, a risk management process based on AS/NZS 4360, a system development process based on Unified Process [6], and a platform for tool-integration based on XML. The first two pillars are presented in this paper as they currently stand.

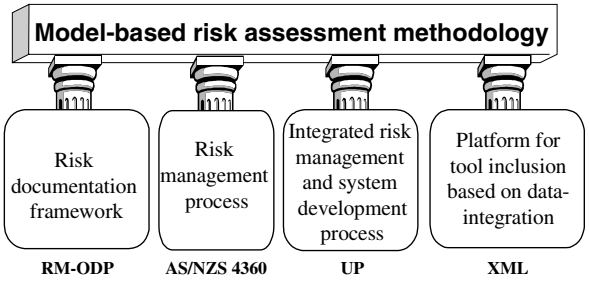

Figure 2. The CORAS framework

The third pillar, Unified Process, focuses mainly on system development rather than supporting the analysis of existing systems. The combination of the Unified Process and the risk management process of AS/NZS 4360 is an ongoing activity and will not be discussed here. The fourth pillar, the CORAS platform for tool integration, is currently being built around an internal data representation formalised in XML/XMI (characterised by XML schema). Cheap XML tools will provide the basic functionality.

3.3. Overview

The CORAS system documentation framework is a specialisation of RM-ODP. As such, the CORAS documentation framework can be understood as a reference framework for model-based risk assessment. RM-ODP contains many features that are not directly relevant for risk assessment. All RM-ODP features are, however, relevant for distributed systems. Since most systems of today are distributed or at least components of distributed systems, it seems reasonable to require that what is already in RM-ODP should also be an element of the CORAS system documentation framework. On the other hand, the CORAS system documentation framework should refine only those parts of RM-ODP that are directly relevant for risk assessment of security critical systems. The CORAS system documentation framework refines RM-ODP by introducing an additional structuring to the viewpoints.

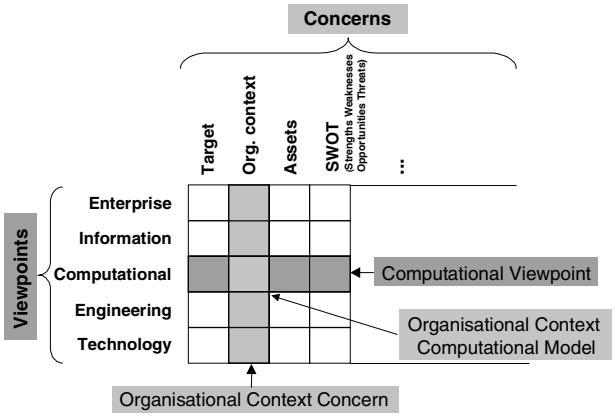

Figure 3. Viewpoints and concerns

As illustrated by Figure 3, the RM-ODP viewpoint structure is divided into concerns targeting security in general and model-based risk assessment in particular. These concerns may be understood as more specialised cross-viewpoint perspectives linking together related information within the five viewpoints. The concerns are further decomposed into models. A model provides the content of a concern with respect to a particular viewpoint. For each model there are guidelines for its development, including concrete recommendations with respect to which modelling languages to use. Figure 4 relates the 22 identified concerns to the five sequential sub-processes of the CORAS risk assessment process.

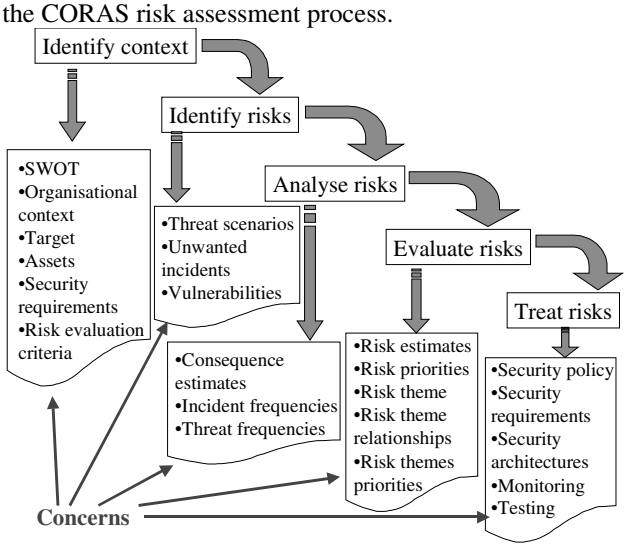

Figure 4. Concerns in process

3.4. Integration of risk assessment methods

In CORAS, we use a carefully selected integration of risk assessment methods in order to assess confidentiality, integrity, availability and accountability throughout the system development and maintenance process. Figure 5

provides an indicative organisation of the integrated risk assessment methods with respect to the different tasks of the CORAS risk management process, emphasising their complementary aspects. Providing details of the integration templates for these methods goes beyond the scope of this paper (see [7] for results in this direction).

	HazOp	FTA	FMECA	Markov	CRAMM
Identify context	In case of brief system description				Valuation of assets
Identify risks	Address all security aspects	Top-down starting from unwanted outcomes	Bottom-up for critical sub-parts		Focus on data groups
Analyse risks	As input for FTA/ FMECA/ Markov	Address top events, basic events, and likelihood	Address failure modes and consequences	Address system states, and likelihood	
Eval. risks	As input	Compare with criteria	Compare with criteria	Compare with criteria	
Treat risks	Identify treatment options	Address priorities	Address barriers and countermeasures	Support maintenance	Identify countermeasures

Figure 5: Relevance of risk assessment methods

3.5. Concerns in the risk assessment process

In this section we identify the concerns and their models that we propose to be created in the CORAS model-based risk assessment process, and we propose notations for each of the models.

3.5.1. Identify context

This activity consists of identification of area of concern, identification and evaluation of assets, and identification of security requirements.

Identify area of concern

The objective of this activity is to construct scenarios outlining concerns regarding the threats to important assets. These areas of concern are likely to lack sufficient detail with respect to the components of threat, and they must be further examined in the following activities.

The first concern to consider is the *SWOT concern*. This concern relates the organisation and its environment, identifying the organisation's strengths, weaknesses, opportunities and threats (SWOT). The context includes the financial, operational, competitive, political, social, client, cultural and legal aspects of the organisation's functions. We propose to use a class diagram where different characteristics of the organisation are classified into the SWOT categories.

The next concern is the *organisational context concern*. Before a risk assessment is commenced, it is necessary to understand the organisation and its capabilities, as well as its goals and objectives and the strategies that are in place to achieve them. This may potentially become a complex model and different notations may be used to document different aspects of the context of the target of evaluation. For instance, activity diagrams may document work processes that involve the target of evaluation and business concepts may be documented in class diagrams.

The final concern to consider in this activity is the *target concern*. This concern describes goals, objectives, strategies, scope and parameters of the activity, or part of the organisation to which the risk assessment process is being applied. Again, this may potentially become a complex model that involves different notations. For instance, use case diagrams with accompanying use case descriptions specify requirements, sequence diagrams illustrate potential scenarios in the target of evaluation, deployment diagrams show system configuration, etc. For a software system, the target concern is typically similar to an architecture specification, but limited to those parts relevant for security concerns.

Identify and evaluate assets

After identifying the scenarios to be examined, the next step is to identify and evaluate important system assets of relevance to these scenarios.

During this activity, the *asset concern* is considered. This concern focuses on the identified assets, their dependencies as well as the results from the valuation. The assets and their valuation can be listed in a table, but we recommend using class diagrams in order to show dependencies between assets.

Identify security requirements

The objective of this activity is to identify security requirements for preserving the identified assets. These security requirements can be classified as confidentiality, integrity, availability and accountability requirements.

To this end, we consider the *security requirements concern*. This concern contains requirements with respect to the four classes of security requirements. With respect to notation, we enhance the requirement specification in the target concern by classifying behaviour according to the different classes of security requirements (e.g., marking information objects as "confidential"), and we extend the requirements model by introducing security requirements. Conventionally, these are expressed in a structured textual form following for example the IEEE Standards [8, 9]. We also recommend using misuse case models [10]. A misuse case is a special kind of use case,

describing behaviour that the system owner does not want to occur. In addition, two kinds of relation between use/misuse cases are introduced in [10], the "prevent" and the "detect" relations. These are used to specify behavioural constraints between use/misuse cases; for instance that blocking repeated registrations may prevent flooding the system. By using misuse cases, security requirements are specified as unwanted behaviour (e.g., violations of policies). In addition to misuse case diagrams, we recommend to use templates such as the one proposed by Sindre and Opdahl in [11] to document misuse cases.

In addition to the security requirements model, we consider the *risk evaluation criteria concern* that describes the criteria against which risk is to be evaluated. In this concern, we specify the conditions that must be met for the system to behave within the required risk level. For instance, it may be required that the confidentiality level is at least "high", where the meaning of the term "high" is defined elsewhere. A simple OCL statement can be used to state this: "confidentiality >= high". This requires that "confidentiality" is well defined and that an ordering exist in which "high" is an element.

3.5.2. Identify risks

This activity consists of identification of threats to assets and identification of vulnerabilities of the assets.

Identify threats to assets

This activity targets to identify the potential threats towards each asset identified. This identification requires a more detailed understanding of the target of evaluation. To this end, we focus on the *threat scenario concern* that contains potential threat scenarios. We document this using sequence or misuse case diagrams.

We also focus on the *unwanted incident concern* that contains potential deviations in the target of evaluation. Similarly with the threat scenario concern, we use sequence and misuse case diagrams to document this. The difference between the threat scenario concern and the unwanted incident concern is that the former documents threats that may lead to misbehaviour whereas the latter documents situations where some threat have led to misbehaviour.

Identify vulnerabilities of assets

Here we try to identify the weaknesses of the assets that might be exploited by the threats previously identified. To this end we focus on the *vulnerability concern* that contains potential weaknesses of the assets. Again we document this concern using sequence and misuse case diagrams. However, this concern focus on weaknesses of the assets that may be exploited and only the scenarios that have no identified prevention mechanisms are depicted, i.e., "successful" misuse cases.

3.5.3. Analyse risks

To analyse risks, we perform a consequence/impact evaluation and evaluate likelihood of occurrence of the risks.

Consequence/impact evaluation

In a previous activity, the consequences of threats were identified in the form of concrete unwanted incidents. In this activity the objective is to determine the level of importance of these consequences, i.e., their impact. To describe this, we focus on the *consequence concern* that contains consequence estimates for the identified unwanted outcomes and a description of their consequences. We use a table of unwanted incidents and their consequences to specify this. Alternatively, we may relate unwanted incidents to their consequences in a class diagram.

Evaluate likelihood of occurrence

In this activity the objective is to determine the likelihood of unwanted incident occurrence. This likelihood depends on factors such as the value of the asset under attack, the asset vulnerabilities and the ease of their exploitation.

In this activity we concentrate on the *unwanted incident frequency concern*. To document this, we produce a frequency model that specifies frequency estimates for the unwanted incidents. This is specified as frequency estimates in an additional column in the consequence table. Included in this concern is the description of the potential causes of the unwanted incident, which often is described by the frequency model. We also focus on the *threat frequency concern* to produce a threat frequency model in which we specify frequency estimates for the identified threats. This is also specified in a table, this time of threats and their corresponding frequency estimates.

3.5.4. Risk evaluation

Risk evaluation means to determine level of risk, prioritise the risks, categorise the risks, determine the interrelationships between risk themes, and prioritising the resulting risks themes.

Determine level of risk

In this activity, the impact of a threat and the likelihood of occurrence are combined in order to estimate the level of risk. We call this the *risk estimates concern* and we produce a risk estimates model that specifies risk estimates for the identified unwanted incidents. For each identified unwanted incident, the risk can be derived based on the likelihood of occurrence. The risk can be specified by expanding the table of unwanted incidents with a column for their risk, classified according to predefined risk levels.

Prioritise risks

The objective of this activity is to evaluate the unwanted incidents and rank them by their estimated level of risk. To this end, we concentrate on the *risk priorities concern* that includes risk priorities based on the estimated risks. The risks can be prioritised according to their risk estimate and other factors such as whether prevention is believed to be achievable, identified by the risk evaluation criteria. Again, a prioritisation can be specified as a column in a risk table, either by grouping the risks into priority levels or by prioritising the risks into a totally ordered list.

Categorise the risks

In this activity, the risks are classified into themes, based on common characteristics. It is more effective and efficient to address risk themes than it is to address each risk individually. The *risk theme concern* targets this by grouping the risks into risk themes. Risks may be grouped according to the means that may be used to prevent them from occurring. For instance, encryption prevents many unwanted incidents such as eavesdropping, tampering, etc. This grouping can be illustrated in class diagrams by classifying similar risks into a risk theme.

Determine interrelationships between themes

The cause-and-effect relationships among the identified risks are identified in this activity. This activity helps to increase the understanding of a set of risks and to determine interrelationships and dependencies to consider when developing protection strategies later. We identify the relationships between risk themes in the *risk-theme relationships concern*. For instance, a risk theme may be in conflict with another risk theme. If encryption is used, this may be in conflict with availability unless appropriate measures are taken (e.g., distribution of suitable decryption schemes). Other relationships between risk themes than "conflicts_with" can be "prevents", "supports" etc. These relationships are illustrated in class diagrams.

Prioritise the resulting themes and risks

Finally, we rank the risk themes. The *risk-theme priority concern* focuses on risk-theme priorities based on the estimated risks. This ranking is done in tables with priorities similar to what is used to specify priorities for individual risks.

3.5.5. Risk treatment

The final activity of the risk assessment process is risk treatment. This activity consists of identification of treatment options and assessment of alternative approaches.

Identify treatment options

This activity includes the development of candidate approaches for mitigating the high-priority risks and themes. A number of candidate approaches exist, and we document them as follows.

A candidate treatment is to specify a security policy, and for this we focus on the *security policy concern*. This concern addresses changes to policies to handle identified security problems. In CORAS, we investigate the use of Ponder [12], a policy specification language, to document security policies framework accompanied by a suitable separate policy deployment scheme. Another possible treatment is a strengthening of the security requirements. We address this in the *security requirements concern* that focuses on strengthened security requirements to handle identified security problems.

The next possible treatment is a change to the security architecture. We focus on this in the *security architecture concern* that incorporates changes to the security architecture to handle identified security problems. This may typically involve changes to the target model.

A possible action that may lead to treatment is to improve testing. To identify this, we have a *testing concern* that focuses on requirements to testing to further investigate potential security problems. For this we use sequence diagrams, TTCN (Tree and Tabular Combined Notation) and we investigate the notations resulting from OMG's standardisation of a testing profile for UML.

Similarly, we can use monitoring to identify candidate treatment and we specify this in a *monitoring concern* that describes requirement to monitoring to help handling potential security problems. Monitoring is system function in its own right and it can be specified similar to system behaviour specification as specified in the target model.

Assess alternative approaches

Finally, after candidate mitigation approaches have been agreed upon, we search for potential solutions in a *treatment priority concern* and we document these in a list of solutions with priorities, typically in a tabular format.

3.6. ODP viewpoints

The five ODP viewpoints are orthogonal to the concerns we have identified for the risk assessment process. We use viewpoints because it may be relevant to view each concern from different viewpoints. For instance, for unwanted incidents, the viewpoint approach may reveal different kinds of unwanted incident. Unwanted incidents are related to the reduction of the value of some system asset, and assets may be visible only from some viewpoints, hence unwanted incidents are visible from only some viewpoints.

As an example, reduction of customer trust is an unwanted incident for an e-commerce platform ("customer trust" is the asset). This unwanted incident depends on a

number of other unwanted incidents that are visible only from some viewpoints. This comes from the fact that the asset "customer trust" depends on assets that are only visible from other viewpoints. Disclosure of confidential information (information viewpoint), erroneous charging (computational viewpoint), disclosure of encryption key (engineering viewpoint), and theft of main server (technology viewpoint) are all unwanted incidents that reduce customer trust.

Furthermore, each unwanted incident may have different causes pertinent to the five viewpoints. For instance, reduction of customer trust in an e-commerce platform may be caused by internal fraud (enterprise viewpoint), inconsistent information (information viewpoint), erroneous charging calculation (computational viewpoint), eavesdropping due to unsatisfactory encryption (engineering viewpoint) or by unavailability due to hardware failures (technology viewpoint).

4. Case study

The CORAS framework and process are being validated in extensive user trials in the areas of e-commerce and telemedicine. In this section we present the modelling approach followed in the first of the user trials (concerning the authentication mechanism of an e-commerce platform) and provide some examples of the risk analyses employed in this context.

In the e-commerce platform, users need be authenticated in order to access the personalised interface or preferences, like shopping lists. Technically, this is not a trivial issue as various alternative approaches have different trade-offs and several implementation pitfalls [13]. In the first trial, the user authentication mechanism used by the e-commerce platform was analysed.

4.1. Identify context

The specification of a system's behaviour can be expressed using UML diagrams like state (or activity) and sequence diagrams (focussing on the target concern from section 3.5.1). In particular, the overall behaviour of a web application like the e-commerce platform can be described as a UML state machine where each state corresponds to a specific HTML page. Events correspond to users clicking on links to other HTML pages in the application. The submission of HTML forms corresponds to events that carry parameters (the form fields). Using the same conventions for events (activation of HTML links), specific interactions of users with the application are expressed as UML sequence diagrams.

Using the modelling approach presented above, the state machine in Figure 6 provides a high level description of the e-commerce platform behaviour with respect to the user authentication and identification.

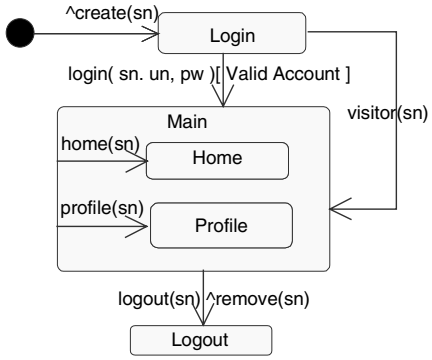

Figure 6. User Authentication behaviour

All HTML pages of the e-commerce platform (like most web applications) are created using a specific HTML template. The only pages were the template is not applied are the "Login" and "Logout" pages that appear before and after the login and logout. This template contains links to major functions of the application. The superstate "Main" includes the states that can be reached directly from the template. These states will normally include more states corresponding to pages reached from internal links.

The actual behaviour of the e-commerce platform has many more states but, for the sake of brevity, only those pertinent to login and registration are shown here.

When a user accesses the Login page, the server creates a unique *session ID* to identify the specific client. The session ID is used to associate each user's client with the user's data stored on the server. This session ID is sent to the user's client in all subsequent HTML pages; all HTML links contain the session ID as a parameter. In the state diagram above the session ID is denoted as the parameter "(sn)". The login carries the username and password as parameters. Users can also access the platform as visitors without authentication but they are not able to use functionality like shopping lists.

In addition to system behaviour and configuration, identification of assets is crucial in order to be able to perform risk assessment. Figure 7 shows business reputation (of the e-commerce platform provider) as an important asset. Business reputation is built up from the financial figures of the business, the trust that its clients have in the business and the reputation of the client. We look at a client as a client of the business having its own customers and therefore its own reputation.

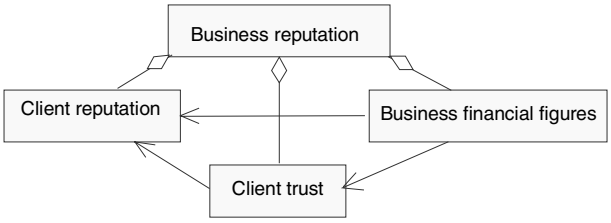

Figure 7. Some assets

Finally, we specify security requirements related to the system specification. An example is: "The User's data shall not become available to other Users by means of the system." The term "User's data" refers to elements in an information model of the platform.

4.2. Identify risks

In order to identify risks, risk analysis is performed. The risk analysis of the user authentication mechanism deployed by the e-commerce platform was based on models of its behaviour like those presented above. Initially CRAMM was applied in order to provide an identification of assets, which in turn provide a basis and justification for the security requirements the mechanism need to meet. Then HazOp, FMEA and FTA were performed, and some examples of this are presented below.

The objective of HazOp is to identify possible unwanted incidents, as well as their causes and consequences. Starting with the system's behaviour as expressed by the state diagram in Figure 6, all events were independently analysed. As an example, an excerpt of HazOp applied on a user's request to access the Login page ("^create(sn)" event) is presented in Figure 8.

In the HazOp table in Figure 8, the first column, Entity, correspond to events of the system behaviour followed by a brief informal description. Security attributes correspond to possible breach of security requirements of confidentiality, integrity availability and accountability. The Deviations column presents deviations from normal or expected behaviour, like undesirable (accidental or malicious) interactions with the system. The next columns present possible causes of the deviations, and their consequences. The Actions column presents steps that can be taken to avoid or mitigate the risk of the deviation to occur. Remarks are presented in the last column.

No.	Entity	Description	Security attribute	Deviation	Causes	Consequences	Actions	Remarks
1	^create (sn)	A user requests to access the Login Page. Server creates a new session number (SN)						
1.1			Disclosure					
1.1.1				User request captured	Openess of Internet	Not exploitable	N/A	No confidential information transmitted
1.1.2				Server response captured	Openness of Internet	SN revealed to capturer	No encryption justified	Deliberate session hijacking is possible
1.2			Manipulation					
1.2.1				A browser or proxy responds with a cached page	Browser or proxy (mis)configuration	User gets a page with invalid SN	N/A	The Login page will returned in the following client request
1.2.2						User gets a SN used by another user	Use large numbers for SN	Inadvertent session hijacking
1.3			Denial / Delay					
1.3.1				User request is blocked by proxy server	Proxy configuration	Server is not accessed	N/A	The server is not accessed
1.3.2				Server response is too slow				Generic deviation
1.4			Unaccountability					
1.4.1				Artificially large number of requests are generated	Deliberate server attack	(1) Creation of too many SNs (2) Server performance degradation	Block access based on client's IP address	Sensitive issue for SN-based user identification

Figure 8. HazOp table for Login page

FMEA was used to identify possible failure modes of individual components. For software systems, like the e-commerce platform, these failures can be wrong results and exceptions or error values returned by function calls to software components. Due the large size of modern software systems, the FMEA table soon becomes very large and time consuming to produce. The CORAS trials therefore focused only on parts of the Web, Application and Database servers of the e-commerce platform.

The objective of Fault Tree Analysis is to document in a structured way the possible routes that can lead to the violations of security requirements identified by HazOp or failures identified by FMEA. As an example, an excerpt of a Fault Tree demonstrating some possible routes that lead to breach of confidentiality by accessing a user's personal data in the e-commerce platform is presented in Figure 9.

The nodes of a fault tree are called *event blocks*, and the root node called *top-event*. The *OR-gates* join alternative means that can lead to their parent nodes. The round circles indicate that the parent events are *basic events* that are not analysed further. In this example, the tree is a branch of larger tree that covers a range of violations of security requirements. There are also more situations resulting in state S0, like circumventing the web server or internal fraud, but these are not presented here.

There is a close relation between the deviations identified by HazOp analysis, the possible failures identified by FMEA and the fault tree constructed in that these deviations appear as nodes ("event blocks") in the fault tree. For example, item 1.1.2 of HazOp table in Figure 8 identified that a capture of a server response

leads to the disclosure of Session ID. This is reflected in FTA tree in Figure 9 where state S4 can be achieved by means of reaching state S7. A tighter integration between the complementary risk analysis methods used is the next step in the development of the CORAS framework.

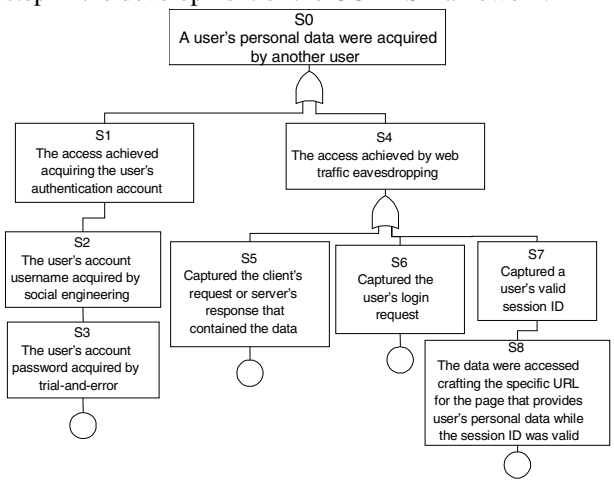

Figure 9. Fault tree example of capturing confidential data

As an alternative approach to document threats and vulnerabilities of systems, we also use misuse case diagrams. Figure 10 shows a misuse case diagram of the situation where a malicious user (Crook) floods the system (the dark oval is a misuse case and the dark actor is a mis-actor, i.e., someone who initiates misuse cases). The misuse case includes the regular use case of registering customer, but the crook misuses this by repeating it a number of times beyond the level the system can handle.

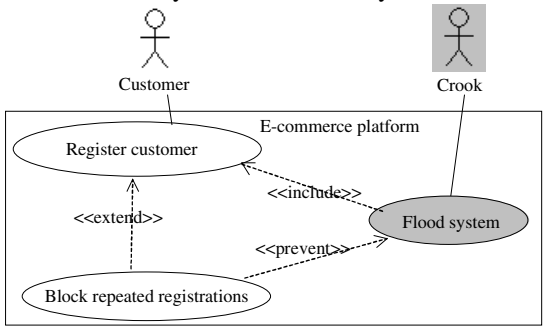

Figure 10. Misuse case diagram

4.3. Analyse risks, risk evaluation and risk treatment

When the risks are identified, analysis of impact and likelihood are performed. The results of this analysis typically annotate existing models or one creates tables listing the risks and assigning likelihood and consequences to them. Similarly, the risk evaluation activity produces levels of risk and risk priorities that further annotate existing specifications. For the categorisation of risks, risk themes may be specified in class diagrams. Figure 11 shows how risks are grouped by the means for which they can be avoided.

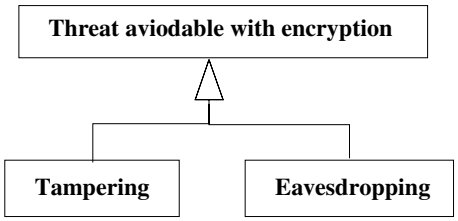

Figure 11. Risk theme

Finally, risk treatment improves system behaviour in order to reduce the chosen risks. Actually, the misuse case diagram in Figure 10 includes specification of a treatment by the "prevents" stereotyped association between the use case "Block repeated registrations" and the misuse case "Flood system". This is one way of specifying treatments.

4.4. Summary

The first CORAS trial session in the e-commerce domain served focused on assessing the CORAS approach in relation to the "risk identification" sub-process, and in order to gain familiarity with use of CRAMM, HazOp, FTA and FMECA for this purpose. The results from the first e-commerce trial are experiences with the use of the specific risk analysis methods, experiences from the overall process, and input to revisions of the way the trials are performed. In relation to the use of integrated risk assessment methods**,** we identified the following results:

- aspects of CRAMM were useful for the purpose of identifying important system assets.
- HazOp worked well with guideword-attributes [14] that reflected the security issues addressed.
- FTA was useful for structured/systematic risk analysis, but was time-consuming and unless contained within clearly defined assessment modules, it might present scalability problems.
- FMECA worked well, but required significant effort to organise and prepare its application.

Furthermore, the different methods provided different results, and the application of more than one method to support risk identification was considered beneficial.

The trial session also demonstrated, through the interactions between the models on the drawing board and the risk analysis methods, that model-based approach to risk assessment has the following main advantages against more traditional ways of conducting risk assessment:

- It supports describing the target of analysis at the right level of abstraction, contributing to the effectiveness and efficiency of the risk assessment process.
- It provides a superior medium for communication and interaction between different groups of stakeholders involved in a risk assessment, contributing to the effectiveness of the risk assessment and to imparting risk assessment feedback into system design.
- The combination of concerns and viewpoints provides a structured way of assessing all relevant aspects of a system from early on in the design process. More traditional approaches to risk assessment could have biased towards a particular view of the system (e.g., data, computation or communication) therefore increasing the possibility that risks originating in complementary views are not identified early enough or they are completely missed.
- Models of the target of evaluation are useful means to systematically address all interactions with the system and each component of the system, with reduced danger of omitting functionality of system components that may posses security risks.

Five more trial sessions are to be conducted (two of which on the same e-commerce platform) focusing on different parts of model based risk assessment and addressing different concerns in relation to different models of the target systems. A summary of the results of all trials of model-based risk assessment will be included in a CORAS deliverable to be released in the summer of 2003.

5. Related work

There exist a number of specialised risk assessment methodologies tailored towards specific domains. For healthcare information systems, the following are some influential methodologies: SEISMED – guidelines on IT security risk analysis for health care IT and security personnel; ISHTAR – implementing secure healthcare telematics applications in Europe; ODESSA - a generic methodology for health care data security; RAMME - a risk analysis model for a medical environment; CPRI Toolkit – health information risk assessment and management; TRA template – threat and risk assessment for health care organisations; Cognitive Fuzzy Modelling for Enhanced Risk Assessment in a Health Care Institution. In the following we review some general risk assessment methodologies that are similar to the CORAS approach.

Since 1990, work has been going on to align and develop existing national and international schemes in one, mutually accepted framework for testing IT security functionality. The Common Criteria [15] (CC) represents the outcome of this work. The Common Criteria project harmonises the European "Information Technology Security Evaluation Criteria (ITSEC)" [16], the Canadian "Canadian Trusted Computer Product Evaluation Criteria (CTCPEC)" and the American "Trusted Computer System Evaluation Criteria (TCSEC) and the Federal Criteria (FC)". Increasingly it is replacing national and regional criteria with a world-wide set accepted by the International Standards Organisation (ISO15408) [17].

The CC is generic and does not provide a methodology for risk analysis. CORAS, on the other hand, is devoted to methodology for risk analysis. Both the CC and CORAS emphasises semiformal and formal specification. However, contrary to the CC, CORAS addresses and develops concrete specification technology addressing risk analysis. The CC and CORAS are orthogonal approaches. The CC provides a common set of requirements for the security functions of IT products and systems, and provides common requirements for assurance measures applied to IT functions of IT products and systems during a security evaluation. CORAS provides specific methodology for one particular kind of assurance measure, namely risk analysis.

Surety Analysis (SA), developed in Sandia National Laboratories [18], is a methodology based on the creation of an explicit model that covers several aspects of the system's behaviour. The modelling framework in SA is proprietary whereas CORAS uses the standardised RM-ODP as a common basis. SA supports modelling by means of basic techniques such as interaction, state and data-flow diagrams. CORAS aims to use the descriptive power of UML/OCL (Object Constraint Language) and to investigate its enhancement with aspects of other modelling paradigms specific to security modelling. In SA, risk and reliability analysis are based on methods such as Fault and Event Trees. CORAS intends to provide support for a wider variety of risk analysis methods and will offer guidelines regarding the hand-to-hand use of modelling and risk analysis throughout the system's life cycle. SA is a generic framework that could be applied in different areas. CORAS uses generic modelling and risk analysis techniques, which are equally applicable across all areas of dependability. OPRA, the tool that has been implemented to support SA, has a "tight" integration of existing commercial software packages with tools developed in Sandia Labs especially for this project. The CORAS platform will facilitate a "loose" integration platform based on widely deployed interchange standards allows different users to adapt the CORAS platform to their needs.

RSDS is a tool-supported methodology developed by King's College London [19] and B-Core UK, ltd. The methodology has been applied in the specification and risk

analysis of reactive systems in automated manufacturing and chemical process control. Both RSDS and CORAS aim to integrate object-oriented modelling and risk analysis. However, CORAS focuses on security risk analysis whereas current work on RSDS focuses on safety and reliability analysis. RSDS focuses on a "tight", highly automated, keyword-driven integration of a small number of risk analysis and reasoning techniques into a single tool, whereas CORAS aims for a "loose" integration framework and development process to be used throughout the development life-cycle. RSDS focuses on critical software/hardware components of a system whereas CORAS aims to cover also "softer" aspects of enterprises such as information processing and policy specifications. RSDS uses a subset of UML, which is then translated to the Abstract Machine Notation of B and SMV modules, and further development requires interaction with these formal methods tools. CORAS are committed to use the RM-ODP standard, which may require a larger part of UML to be considered, and the constraints imposed by high automation in risk analysis and interaction in formal verification could be avoided. CORAS is more appropriate than RSDS for enterprise-wide use in complex security critical systems by teams of risk analysts and designers with little or no exposure to formal methods. RSDS is more appropriate for analysis and formal verification of critical components by developers with a strong background in formal methods and little exposure to risk analysis.

The Control Objectives for Information and related Technology (COBIT) [20] addresses the management of IT. The main objective of the COBIT project is the development of clear policies and good practices for security and control in IT for world-wide endorsement by commercial, governmental and professional organisations. Similar to CC, COBIT and CORAS are orthogonal approaches. COBIT focuses on control objectives defined in a process-oriented manner following the principle of business re-engineering. The IT process of assessing risks satisfies the business requirement of supporting management decisions through achieving IT objectives and responding to threats by reducing complexity, increasing objectivity and identifying important decision factors. It is enabled by the organisation engaging itself in IT risk-identification and impact analysis. CORAS provides a tight integration of viewpoint-oriented modelling in the whole risk management process, including the sub-processes of risk identification and risk analysis.

CCTA Risk Analysis and Management Methodology (CRAMM) was developed by the British Government's Central Computer and Telecommunications Agency (CCTA) [21] with the aim of providing a structured and consistent approach to computer security management for all systems. The UK National Health Service considers CRAMM to be the standard for the risk analysis of information systems within healthcare establishments. CRAMM is an important source of inspiration for CORAS, and aspects of CRAMM have been incorporated in CORAS. Contrary to CRAMM, CORAS provides a risk analysis process in which modelling is tightly integrated. CORAS employs modelling not only to document the target system, but also to describe its context and possible threats. Moreover, CORAS employs modelling to document the results from risk analysis and the assumptions on which these results depend. CORAS also employs graphical UML-based modelling as a medium for communication and interaction between different groups of stakeholders involved in a risk analysis. Contrary to CRAMM, CORAS complies with state-of-the-art international standards for risk management, documentation, modelling and development of systems. CORAS provides a platform for tool-integration based on XML technology supporting openness as well as interoperability. CRAMM is also supported by software, but this software is proprietary. CCTA has extended CRAMM into an overall system development process by developing an interface between CRAMM and SSADM (Structured Systems Analysis and Design Method). This corresponds to the CORAS integrated risk management and system development process based on AS/NZS 4360 and UP. However, SSADM is based on structured analysis that was the modelling technology of the 80'ies, while UP has been developed for state-of-the-art object-oriented modelling methodology in the OMG standard UML.

6. Conclusions and future work

In this paper we have presented an integration of modelling and risk analysis. This is beneficial for risk management for several reasons: It improves the risk analysis itself since the understanding of the target of evaluation is enhanced by precise specifications of how it is structured and how it behaves. Traditionally, risk analysis is performed on the basis of informal descriptions of the target of evaluation and such an approach is more open for misunderstandings. We argue that UML diagrams such as the state diagram in Figure 6 gives a superior specification of system behaviour compared to free text or some other informal formats. Secondly, a model-based risk assessment facilitates communication, both internally between the actors involved during risk analysis and externally to the stakeholders. We claim that for instance a misuse case diagram is easier to understand than a HazOp table for some stakeholders such as the management. Thirdly, the precision level is improved by

introducing semi-formal notations such as UML. This is not only true for specification of the target of evaluation, but also for the risk analysis results and on the assumptions on which their validity depend. Fourthly, by referring to a common model, different risk assessment methodologies such as HazOp and Fault trees are better integrated. For instance, the asset model can be referred to in both approaches so that they uncover different aspects of the same asset.

CORAS aims to achieve a tight integration between risk analysis techniques and modelling notation. In a first trial using the user-authentication mechanism of a Web e-Commerce platform, the system's behaviour was expressed as a UML state machine. Then HazOp was applied on each event of this state machine addressing security threats involved in each interaction with the system. The system's structure was expressed as a UML component diagram. Then FMEA was applied on each component of the system in order to identify potential failures of these components. This systematic way of assessing a system provides an assurance that all interactions with and all components of the system will be addressed. As a consequence, despite the relative simplicity of the mechanism analysed in the trial, considerable risks were identified that have to be addressed further.

CORAS is an ongoing project and we plan to improve support for model-based risk assessment by implementing the CORAS platform. This will provide risk assessors with an integration platform to use when performing the model-based risk assessment. We also plan to perform more trials to get feedback on the CORAS framework and identify areas of improvement.

Acknowledgements

The results reported in this paper follow from work carried out by the partners in the CORAS project.

References

[1] CORAS, "CORAS: A platform for risk analysis of security critical systems", 2000.

[2] OMG, "OMG Unified Modeling Language Specification, v1.4," Object Management Group, formal/01-09-67, September, 2001.

[3] ISO/IEC, "Guidelines for the management of IT Security - Part 1: Concepts and models for IT Security," ISO/IEC, TR 13335-1, 2001.

[4] Standards Australia, "AS/NZS 4360: Risk Management," Standards Australia, Standard, AS/NZS 4360, 1999.

[5] ISO/IEC JTC1/SC21, "Basic reference model of open distributed processing, part 1: Overview", ITU-T X.901 - ISO/IEC 10746-1, August, 1995.

[6] P. Kruchten, *Rational Unified Process*: Addison-Wesley, 1998.

[7] R. Fredriksen, M. Kristiansen, B. A. Gran, K. Stølen, T. A. Opperud, and T. Dimitrakos, "The CORAS Framework for a model-based risk management process," presented at 21st International Conference on Computer Safety, Reliability and Security, SAFECOMP 2002.

[8] IEEE, "IEEE Recommended Practice for Software Requirements Specifications," Std 830-1998.

[9] IEEE, "IEEE Guide for Developing System Requirements Specifications," Std 1233-1998.

[10] G. Sindre and A. L. Opdahl, "Eliciting Security Requirements by Misuse Cases," presented at 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-PACIFIC 2000), Sydney, Australia, 2000.

[11] G. Sindre and A. L. Opdahl, "Templates for Misuse Description," presented at Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, Interlaken, Switzerland, 2001.

[12] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "Ponder: A Language for Specifying Security and Management Policies for Distributed Systems - Version 2.3," Department of Computing, Imperial College, London, UK, Imperial College Research Report DoC 2000/1, October 20, 2000.

[13] K. Fu, E. Sit, K. Smith, and N. Feamster, "Dos and Don'ts of Client Authentication on the Web," MIT Laboratory for Computer Science, Tech Report, 818, May, 2001.

[14] R. Winther, O.-A. Johnsen, and B. A. Gran, "Security Assessments of Safety Critical Systems Using HAZOPs," presented at 20th International Conference on Computer Safety, Reliability and Security, SAFECOMP 2001, Budapest, Hungary, 2001.

[15] Common Criteria Organisation, "Common Criteria for Information Technology Security Evaluation", http://www.commoncriteria.org, accessed: 2002.

[16] Communications-Electronics Security Group, "Information Security Evaluation Criteria", http://www.cesg.gov.uk/assurance/iacs/itsec/index.htm

[17] ISO/IEC, "Information Technology -- Security techniques -- Evaluation Criteria for IT Security," ISO/IEC, 15408-1, 1999.

[18] Sandia National Laboratories, "Surety Analysis", http://www.sandia.gov, accessed: 2002.

[19] Reactive System Design Support, "RSDA", http://www.kcl.ac.uk.

[20] Control Objectives for Information and related Technology, "COBIT", http://www.isaca.org/ct_denld.htm.

[21] B. Barber and J. Davey, "The Use of the CCTA Risk Analysis and Management Methodology CRAMM in Health Information Systems," MEDINFO 92.