The CORAS Framework for a Model-Based Risk Management Process

Rune Fredriksen[1], Monica Kristiansen[1], Bjørn Axel Gran[1], Ketil Stølen[2], Tom Arthur Opperud[3], and Theo Dimitrakos[4]

[1] Institute for Energy Technology, Halden, Norway
{Rune.Fredriksen,Monica Kristiansen,Bjorn-Axel Gran}@hrp.no
http://www.ife.no
[2] Sintef Telecom and Informatics, Oslo, Norway
Ketil.Stolen@informatics.sintef.no
http://www.sintef.no
[3] Telenor Communications AS R&D, Fornebu, Norway
Tom-Arthur.Opperud@telenor.com
http://www.telenor.no
[4] CLRC Rutherford Appleton Laboratory (RAL), Oxfordshire, UK
T.Dimitrakos@rl.ac.uk
http://www.rl.ac.uk

Abstract. CORAS is a research and technological development project under the Information Society Technologies (IST) Programme (Commission of the European Communities, Directorate-General Information Society). One of the main objectives of CORAS is to develop a practical framework, exploiting methods for risk analysis, semiformal methods for object-oriented modelling, and computerised tools, for a precise, unambiguous, and efficient risk assessment of security critical systems. This paper presents the CORAS framework and the related conclusions from the CORAS project so far.

1 Introduction

CORAS [1] is a research and technological development project under the Information Society Technologies (IST) Programme (Commission of the European Communities, Directorate-General Information Society). CORAS started up in January 2001 and runs until July 2003. The CORAS main objectives are as follows:

- To develop a practical framework, exploiting methods for risk analysis, semiformal methods for object-oriented modelling, and computerised tools, for a precise, unambiguous, and efficient risk assessment of security critical systems.
- To apply the framework in two security critical application domains: telemedicine and e-commerce.
- To assess the applicability, usability, and efficiency of the framework.
- To promote the exploitation potential of the CORAS framework.

S. Anderson et al. (Eds.): SAFECOMP 2002, LNCS 2434, pp. 94–105, 2002.

2 The CORAS Framework

This section provides a high-level overview of the CORAS framework for a model-based risk management process. By "a model-based risk management process" we mean a tight integration of state-of-the-art UML-oriented modelling technology (UML = Unified Modeling Language) [2] in the risk management process. The CORAS model-based risk management process employs modelling technology for three main purposes:

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

A model-based risk management process is motivated by several factors:

- Risk assessment requires correct descriptions of the target system, its context and all security relevant 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 identification and assessment 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 the risk management process in the system development process. This may considerably reduce development costs and ensure that the specified security level is achieved.

The CORAS framework for a model-based risk management process has four main anchor-points, a system documentation framework based on the Reference Model for Open Distributed Processing (RM-ODP) [3], a risk management process based on the risk management standard AS/NZS 4360 [4], a system development process based on the Rational Unified Process (RUP) [5], and a platform

for tool-integration based on eXtensible Markup Language (XML) [6]. In the following we describe the four anchor-points and the model-based risk management process in further detail.

2.1 The CORAS Risk Management Process

The CORAS risk management process provides a sequencing of the risk management process into the following five sub-processes:

1. *Context Identification*: Identify the context of the analysis that will follow. The approach proposed here is to select usage scenarios of the system under examination.
2. *Risk Identification*: Identify the threats to assets and the vulnerabilities of these assets.
3. *Risk Analysis*: Assign values to the consequence and the likelihood of occurrence of each threat identified in sub-process 2.
4. *Risk Evaluation*: Identify the level of risk associated with the threats already identified and assessed in the previous sub-processes
5. Risk Treatment: Address the treatment of the identified risks.

The initial experimentation with UML diagrams can be summarised into the following:

1. UML use case diagrams support the identification of both the users of a system (actors) and the tasks (use cases) they must undertake with the system. UML scenario descriptions can be used to give more detailed input to the identification of different usage scenarios in the CORAS risk management process.
2. UML class/object diagrams identify the classes/objects needed to achieve the tasks, which the system must help to perform, and the relationships between the classes/objects. While class diagrams give the relationships between general classes, object diagrams present the instantiated classes. This distinction could be important when communicating with users of the system.

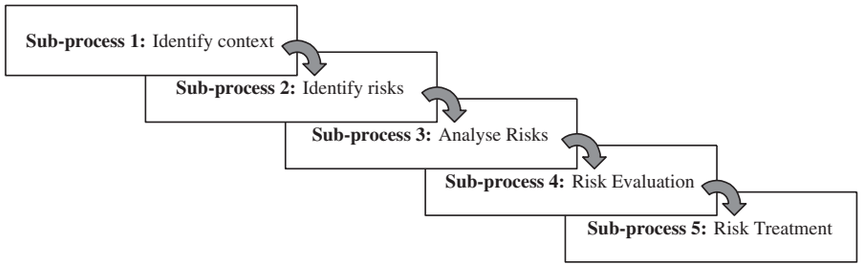

Fig. 1. Overview over the CORAS risk management process

3. UML sequence diagrams describe some aspects of system behaviour by e.g. showing which messages are passed between objects and in what order they must occur. This gives a dynamic picture of the system and essential information to the identification of important usage scenarios.
4. UML activity diagrams describe how activities are co-ordinated and record the dependencies between activities.
5. UML state chart diagrams or UML activity diagrams can be used to represent state transition diagrams. The UML state chart diagram may be used to identify the sequence of state transitions that leads to a security break.

2.2 The CORAS System Documentation Framework

The CORAS system documentation framework is based on RM-ODP. RM-ODP defines the standard reference model for distributed systems architecture, based on object-oriented techniques, accepted at the international level. RM-ODP is adopted by ISO (ISO/IEC 10746 series: 1995) as well as by the International Telecommunication Union (ITU) (ITU-T X.900 series: 1995).

As indicated by Figure 2, 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. The CORAS framework extends RM-ODP with:

1. Concepts and terminology for risk management and security.
2. Carefully defined viewpoint-perspectives targeting model-based risk management of security-critical systems.
3. Libraries of standard modelling components.
4. Additional support for conformance checking.
5. A risk management module containing risk assessment methods, risk management processes, and a specification of the international standards on which CORAS is based.

2.3 The CORAS Platform for Tool Integration

The CORAS platform is based on data integration implemented in terms of XML technology. Figure 3 outlines the overall structure. The platform is built up around an internal data representation formalised in XML/XMI (characterised by XML schema). Standard XML tools provide much of the basic functionality. This functionality allows experimentation with the CORAS platform and can be used by the CORAS crew during the trials. Based on the eXtensible Stylesheet Language (XSL), relevant aspects of the internal data representation may be mapped to the internal data representations of other tools (and the other way around). This allows the integration of sophisticated case-tools targeting system development as well as risk analysis tools and tools for vulnerability and treat management.

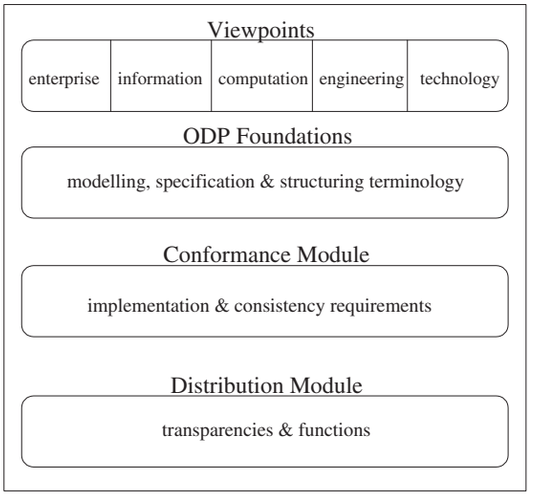

Fig. 2. The main components of RM-ODP

2.4 The CORAS Model-Based Risk Management Process

The CORAS methodology for a model-based risk management process builds on:

- HAZard and OPerability study (HAZOP) [7];
- Fault Tree Analysis (FTA) [8];
- Failure Mode and Effect Criticality Analysis (FMECA) [9];
- Markov analysis methods (Markov) [10];
- Goals Means Task Analysis (GMTA) [11];
- CCTA Risk Analysis and Management Methodology (CRAMM) [12].

These methods are to a large extent complementary. They address confidentiality, integrity, availability as well as accountability; in fact, all types of risks, threats, hazards associated with the target system can potentially be revealed and dealt with. They also cover all phases in the system development and maintenance process. In addition to the selected methods other methods may also be needed to implement the different sub-processes in the CORAS risk management process. So far two additional methods have been identified. These are Cause-Consequence Analysis (CCA) [13] and Event-Tree Analysis (ETA) [13].

The CORAS risk management process tightly integrates state-of-the-art technology for semiformal object-oriented modelling. Modelling is not only used to provide a precise description of the target system, but also to describe its context and possible threats. Furthermore, description techniques are employed to document the risk assessment results and the assumptions on which these results

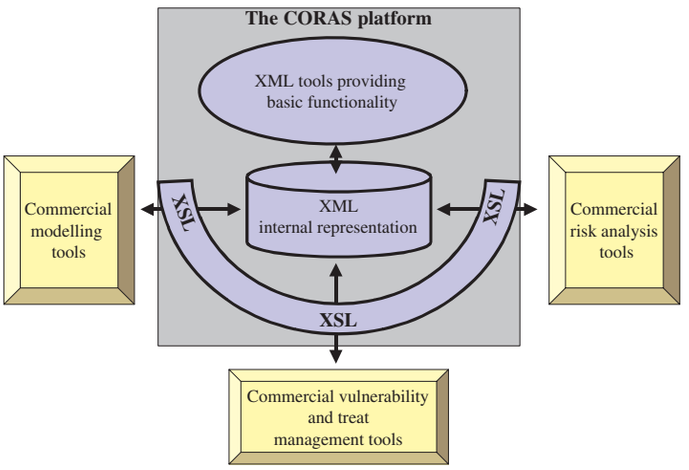

Fig. 3. The meta-model of the CORAS platform

depend. Finally, graphical UML-based modelling provides a medium for communication and interaction between different groups of stakeholders involved in risk identification and assessment.

The following table gives a brief summary and a preliminary guideline to which methods that should be applied for which sub-process in the CORAS model-based risk management process. This guideline will be updated further during the progress of the CORAS project.

Risk management requires a firm but nevertheless easily understandable basis for communication between different groups of stakeholders. Graphical object-oriented modelling techniques have proved well suited in this respect for requirements capture and analysis. We believe they are equally suited as a language for communication in the case of risk management. Entity relation diagrams, sequence charts, dataflow diagrams and state diagrams represent mature paradigms used daily in the IT industry throughout the world. They are supported by a wide set of sophisticated case-tool technologies, they are to a large extent complementary and, together they support all stages in a system development.

Policies related to risk management and security are important input to risk assessment of security critical systems. Moreover, results from a risk assessment will often indicate the need for additional policies. Ponder [14] is a very expressive declarative, object-oriented language for specifying security and management policies for distributed systems. Ponder may benefit from an integration with graphical modelling techniques. Although the four kinds of graphical modelling techniques and Ponder are very general paradigms they do not always provide

Table 1. How the RA methods apply to the CORAS risk management process

Sub-process	**Goal**	**Recommended methods**
Context Identification	Identify the context of the analysis (e.g. areas of concern, assets and security requirements).	CRAMM, HAZOP
Risk Identification	Identify threats.	HAZOP
Risk Analysis	Find consequence and likelihood of occurence.	FMECA, CCA, ETA
Risk Evaluation	Evaluate risk (e.g. risk level, prioritise, categorise, determine interrelationships and prioritise).	CRAMM
Risk Treatment	Identify treatment options and assess alternative approaches.	HAZOP

the required expressiveness. Predicate logic based approaches like OCL [15] in addition to contract-oriented modelling are therefore also needed.

2.5 The CORAS System Development and Maintenance Process

The CORAS system development and maintenance process is based on an integration of the AS/NZS 4360 standard for risk management and an adaptation of the RUP for system development. RUP is adapted to support RM-ODP inspired viewpoint oriented modelling. Emphasis is placed on describing the evolution of the correlation between risk management and viewpoint oriented modelling throughout the system's development and maintenance lifecycle.

In analogy to RUP, the CORAS process is both stepwise incremental and iterative. In each phase of the system lifecycle, sufficiently refined versions of the system (or its model) are constructed through subsequent iterations. Then the system lifecycle moves from one phase into another.

In analogy to the RM-ODP viewpoints, the viewpoints of the CORAS framework are not layered; they are different abstractions of the same system focusing on different areas of concern. Therefore, information in all viewpoints may be relevant to all phases of the lifecycle.

3 Standard Modeling Components

Much is common from one risk assessment to the next. CORAS aims to exploit this by providing libraries of reusable specification fragments targeting the risk management process and risk assessment. These reusable specification fragments are in the following referred to as standard modelling components. They will typically be UML diagrams annotated with constraints expressed in OCL

(Object Constraint Language), or in for this purpose other suitable specification languages.

The process of developing standard modelling components will continue in the CORAS project. In this phase the focus has been to:

1. Build libraries of standard modelling components for the various security models developed in CORAS.
2. Provide guidelines for the structuring and maintenance of standard modelling components.

The following preliminary results and conclusions have been reached:

1. Standard modelling components may serve multiple purposes in a process for model-based risk management. They can represent general patterns for security architectures, or security policies. They can also represent the generic parts of different classes of threat scenarios, as well as schemes for recording risk assessment results and the assumptions on which they depend.
2. In order to make effective use of such a library, there is need for a computerised repository supporting standard database features like storage, update, rule-based use, access, search, maintenance, configuration management, version control, etc.
3. XMI offers a standardised textual XML-based representation of UML specifications. Since UML is the main modelling methodology of the CORAS framework and XML has been chosen as the main CORAS technology for tool integration, the repository should support XMI based exchange of models.
4. The UML meta-model is defined in Meta Object Facility (MOF) [16]. In relation to a CORAS repository, MOF may serve as a means to define a recommended subset of UML for expressing standard modelling components, required UML extensions to support a model-based risk management process, as well as the grammar of component packets. The repository should therefore be MOF based.
5. To support effective and smooth development of a consistent library, a single CORAS repository that all partners in the consortium may access via the Internet would be useful.
6. The OMG standards MOF and XMI ensure open access to the library and flexible exchange of standard modelling components between tools. There are already commercial tools for building repositories supporting MOF and XMI on the market; others are under development. The consortium will formalise the library of standard modelling components in terms of MOF and XMI using a suitable for this purpose UML CASE-tool.

4 The CORAS Trials

The trials in CORAS are performed within two different areas: e-commerce and telemedicine. The purpose of the trials is to experiment with all aspects of the

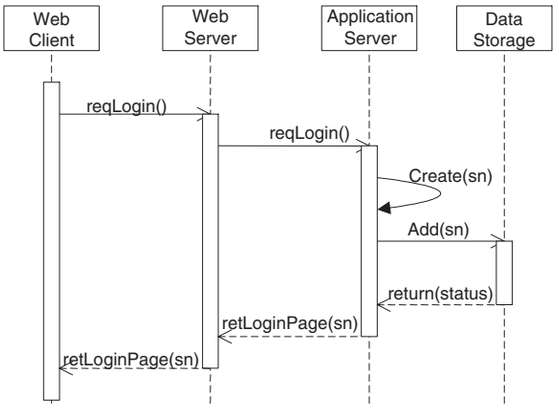

Fig. 4. An example of the UML sequence diagram used in the trial

framework during its development, provide feedback for improvements and offer an overall assessment. The first e-commerce trial was based on the authentication mechanism. Among other models an indicative UML sequence diagram for starting the FMECA method, see Figure 4, was used. It is important to stress that the sequence diagram presented here is only one example of typical possible behaviours of the system. Scenarios like unsuccessful login, visitor accessing the platform, registration of new user, etc, could also be modelled. A more detailed description of the CORAS trials will be provided in the reports from the CORAS project.

This trial was focused on the sub process 2 – identify risks, and to gain familiarity with use of CRAMM, HAZOP, FTA and FMECA for this purpose. The results from the first e-commerce trial are divided into four partly overlapping classes:

1. Experiences with the use of the specific risk analysis methods.
2. Experiences from the overall process.
3. Input to changes to the way the trials are performed.
4. Input to minor changes of the CORAS risk management process.

4.1 Experiences with the Use of the Specific Risk Analysis Methods

The individual methods used during the first e-commerce risk analysis trial session provided the following main results:

- CRAMM was useful for identification of important system assets.
- HAZOP worked well with security-related guidewords/attributes [17] that reflected the security issues addressed.

- FTA was useful for structured/systematic risk analysis, but was time-consuming and might present scalability problems.
- FMEA worked well, but has to be well organised before it is applied and it may even be prepared beforehand by the system developers.

The trial also demonstrated, through the interactions between the models on the board and the risk analysis methods that model-based risk assessment provides an effective medium for communication and interaction between different groups of stakeholders involved in a risk assessment.

4.2 Experiences from the Overall Process

The CORAS risk management process was initially difficult to follow without guidance from experienced risk analysts. Especially the interfacing between models and the objective for using each method was not initially clear. During the process it became obvious that sufficient input of documentation, including models, was critical to obtain valuable results. The process did, however, provide identification of threats and some important issues were discovered despite time limitations. The different methods provided complementary results, and the application of more than one method was very successful.

4.3 Input to Changes to the Way the Trials Are Performed

One of the objectives of the first e-commerce trial was to provide input on how the following trials should be performed. Four major issues were addressed:

1. The trials should be more realistic, regarding the people that participate, the duration and the functionality that is analysed.
2. The CORAS risk management process should be followed more formally.
3. Documentation, including models, should be provided in sufficient time before the trial so that clarifications can be provided in time.
4. Tool support for the different risk analysis methods would make the application of the methods more productive.

4.4 Input to Minor Changes of the CORAS Risk Management Process

The major results from this trial for the subsequent updates of the CORAS risk management process are:

1. Guidelines for the application of the CORAS risk management process need to be provided;
2. The terminology in use need to be defined in more detail; and
3. Templates for the different risk analysis methods need to be available.

5 Conclusions

This paper presents the preliminary CORAS model-based risk management process. The main objective of the CORAS project is to develop a framework to support risk assessment of security critical systems, such as telemedicine or e-commerce systems. A hypothesis where risk analysis methods traditionally used in a safety context were applied in a security context, has been evaluated - and will be evaluated further during the forthcoming trials.

This paper also presents the experiences from the first trial in the project. The different methods provided complementary results, and the use of more than one method seemed to be an effective approach. The first trial experiences also demonstrated the advantages of the interactions between the models on the board and the risk analysis methods. In addition the trial provided the identification of threats and some important issues for further follow up. The trials to be performed during the spring 2002 will provide feedback to updated versions of the recommendations developed in the CORAS project.

References

[1] CORAS: "A Platform for Risk Analysis of Security Critical Systems", IST-2000-25031,(2000).(`http://www.nr.no/coras/`) 94
[2] OMG: UML proposal to the Object Management Group(OMG), Version 1.4, 2000. 95
[3] ISO/IEC 10746: Basic Reference Model of Open Distributed Processing, 1999. 95
[4] AS/NZS 4360: Risk Management. Australian/New Zealand Standard 1999. 95
[5] Krutchen, P.: The Rational Unified Process, An Introduction, Addison-Wesley (1999) 95
[6] W3C: Extensible Markup Language (XML) 1.0 October 2000 96
[7] Redmill F., Chudleigh M., Catmur J.: Hazop and Software Hazop, Wiley, 1999. 98
[8] Andrews J. D., Moss, T. R.: Reliability and Risk Assessment, 1st Ed. Longman Group UK, 1993. 98
[9] Bouti A., Kadi A. D.: A state-of-the-art review of FMEA/FMECA, International Journal of Reliability, Quality and Safety Engineering, vol. 1, no. 4, pp (515-543), 1994. 98
[10] Littlewood B.: A Reliability Model for Systems with Markov Structure, Appl. Stat., 24 (2), pp (172-177), 1975. 98
[11] Hollnagel E.: Human Reliability Analysis: Context and Control, Academic press, London, UK, 1993. 98
[12] Barber B., Davey J.: Use of the CRAMM in Health Information Systems, MEDINFO 92, ed Lun K. C., Degoulet P., Piemme T. E. and Rienhoff O., North Holland Publishing Co, Amsterdam, pp (1589 – 1593), 1992. 98
[13] Henley E. J., and Kumamoto, H.: Probabilistic Risk Assessment and Management for Engineers and Scientists. 2nd Ed. IEEE Press, 1996. 98
[14] Damianou N., Dulay N., Lupu E., and Sloman M.: Ponder: A Language for Specifying Security and Management Policies for Distributed Systems. The Language Specification - Version 2.2. Research Report DoC 2000/1, Department of Computing, Imperial College, London, April, 2000. 99

[15] Warmer Jos B., and Kleppe Anneke G.: The Object Constraint Language - precise modeling with UML. Addison-Wesley, 1999. 100

[16] OMG: Meta Object Facility. Object Management Group(OMG), `http://www.omg.org` 101

[17] Winther, Rune et al.: Security Assessments of Safety Critical Systems Using HAZOPs, U.Voges (Ed.): SAFECOMP 2001, LNCS 2187, pp. (14-24), 2001, Springer-Verlag Berlin Heidelberg 2001. 102