Model-based risk assessment – the CORAS approach

Ketil Stølen[1], Folker den Braber[1], Theo Dimitrakos[2], Rune Fredriksen[3], Bjørn Axel Gran[3], Siv-Hilde Houmb[4], Mass Soldal Lund[1], Yannis C. Stamatiou[5] and Jan Øyvind Aagedal[1]

[1]Sintef Telecom & Informatics, Norway,{kst,fbr,joea}@sintef.no:[2]CLRC Rutherford Appleton Laboratory, UK, t.dimitrakos@rl.ac.uk:[3]Institute for Energy Technology, Norway, {runefr,bjornag}@hrp.no: [4]Telenor R&D, Norway, siv-hilde.houmb@telenor.com: [5]Computer Technology Institute, Greece, stamatiu@cti.gr

Abstract: The EU-funded CORAS project (IST-2000-25031) is developing a framework for model-based risk assessment of security-critical systems. This framework is characterised by: (1) A careful integration of techniques and features from partly complementary risk assessment methods. (2) Patterns and methodology for UML oriented modelling targeting the different risk assessment methods. (3) A risk management process based on AS/NZS 4360. (4) A risk documentation framework based on RM-ODP. (5) An integrated risk management and system development process based on UP. (6) A platform for tool-inclusion based on XML.

Key words: risk assessment, modelling, IT security, maintenance

1. Introduction

CORAS [10] aims for improved methodology and computerised support for precise, unambiguous, and efficient risk assessment of security-critical systems. The CORAS project focuses on the tight integration of viewpoint-oriented semiformal modelling in the risk assessment process, in the following referred to as model-based risk assessment. Model-based risk assessment differs from traditional risk assessment in the sense that it

- combines complementary risk assessment methods for assessing different models of the target of evaluation;
- gives detailed recommendations for the use of modelling methodology in conjunction with risk assessment;
- provides modelling methodology to support the documentation of risk assessment results.

An important aspect of the CORAS project is the practical use of the Unified Modelling Language (UML) [32] and the Unified Process (UP) [21] in the context of security and risk assessment.

CORAS addresses security-critical systems in general, but places 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 [17]. An IT system for CORAS is not just technology, but also the humans interacting with the technology and all relevant aspects of the surrounding organisation and society.

The CORAS consortium consists of three commercial companies: Intracom (Greece), Solinet (Germany) and Telenor (Norway); seven research institutes: CTI (Greece), FORTH (Greece), IFE (Norway), NCT (Norway), NR (Norway), RAL (UK) and Sintef (Norway); as well as one university college: QMUL (UK). Telenor and Sintef are responsible for the administrative and scientific coordination, respectively. CORAS started in January 2001 and runs until July 2003. Since CORAS is an ongoing project, the approach presented in this chapter may not be fully implemented in the final version of the CORAS framework.

2. The CORAS framework

As illustrated in *Figure 1*, the main focus of the CORAS framework is model-based risk assessment; moreover, the framework is founded on four pillars: (1) A risk documentation framework based on RM-ODP [16]. (2) A risk management process based on AS/NZS 4360 [1]. (3) An integrated risk management and development process based on UP [21]. (4) A platform for tool-inclusion based on XML [5].
In the following subsections we describe the rationale behind the CORAS framework, its main focus as well as the four pillars.

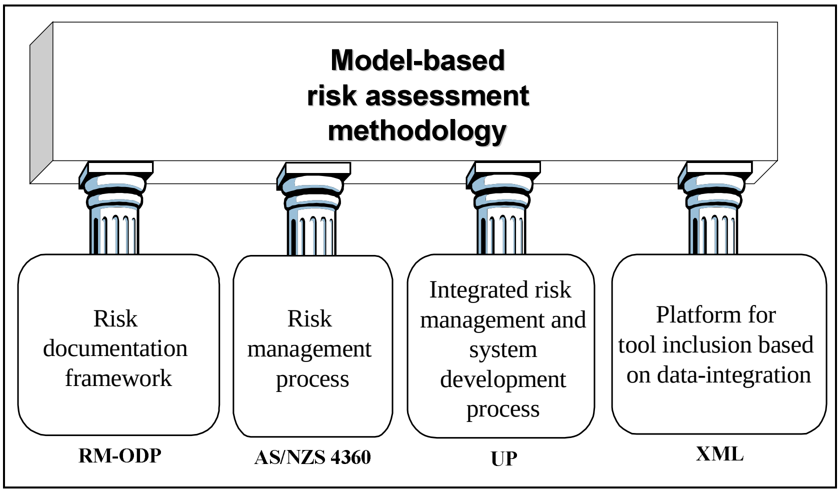

Figure 1. **The CORAS framework**

3. The rationale

As illustrated in *Figure 2*, model-based risk assessment employs modelling methodology for three main purposes: (1) To describe the target of evaluation at the right level of abstraction. (2) As a medium for communication and interaction between different groups of stakeholders involved in a risk assessment. (3) To document risk assessment results and the assumptions on which these results depend.

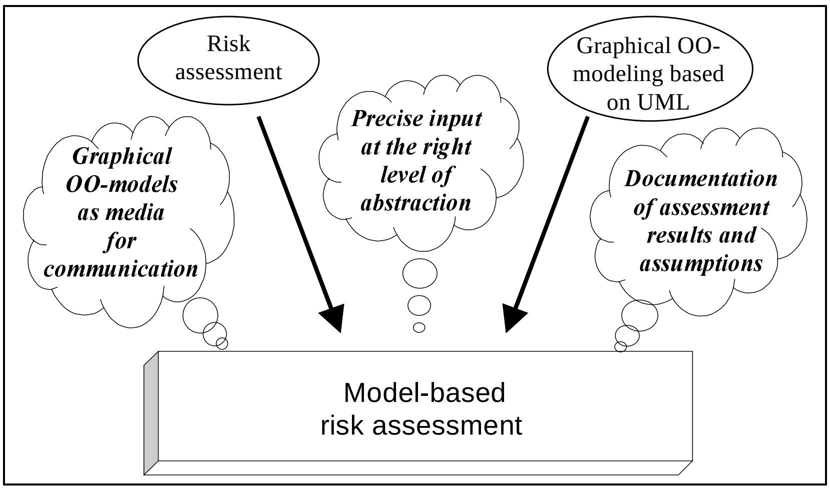

Figure 2. **Model-based risk assessment**

The choice of model-based risk assessment is motivated by several hypotheses:

- Risk assessment benefits from correct descriptions of the target of evaluation, its context and security issues. The modelling methodology furthers the precision of such descriptions, and this is likely to improve the quality of risk assessment results.
- The graphical style of UML facilitates communication and interaction between stakeholders involved in a risk assessment. This may improve the quality of risk assessment results, and reduce the danger of throwing away time and resources on misconceptions.
- The modelling methodology facilitates a more precise documentation of risk assessment results and the assumptions on which their validity depends. This is likely to reduce maintenance costs by increasing the possibilities for reusing and updating assessment results when the target of evaluation is maintained.
- The modelling methodology provides a solid basis for the integration of assessment methods. This may improve the effectiveness of the assessment process.
- The modelling methodology is supported by a rich set of tools from which the risk assessment benefits. This may improve the quality of assessment results and reduce costs. It may also further productivity and maintenance.
- The modelling methodology 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.

4. The risk documentation framework

The CORAS risk documentation framework is a specialisation of the Reference Model for Open Distributed Processing (RM-ODP) [16]. RM-ODP is an international standard reference model for distributed systems architecture, based on object-oriented techniques. 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 risk documentation framework is a specialisation of RM-ODP and 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 IT systems of today are distributed or at least components of distributed systems, the CORAS risk documentation framework contains RM-ODP in full. On the other hand, the CORAS risk documentation framework refines only those parts of RM-ODP that are directly relevant for risk assessment of security critical systems. The CORAS risk documentation framework refines RM-ODP in the following manner.

- The RM-ODP terminology is extended with two new classes of terminology, namely, concepts for risk assessment and security. *Figure 3* illustrates the relationship between some of the most important notions in the risk assessment terminology.
- The RM-ODP viewpoint structure is divided into concerns targeting security in general and model-based risk assessment in particular. As illustrated in *Figure 4*, concerns may be understood as 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.

– The RM-ODP conformance module is extended with additional support for conformance checking targeting concerns.

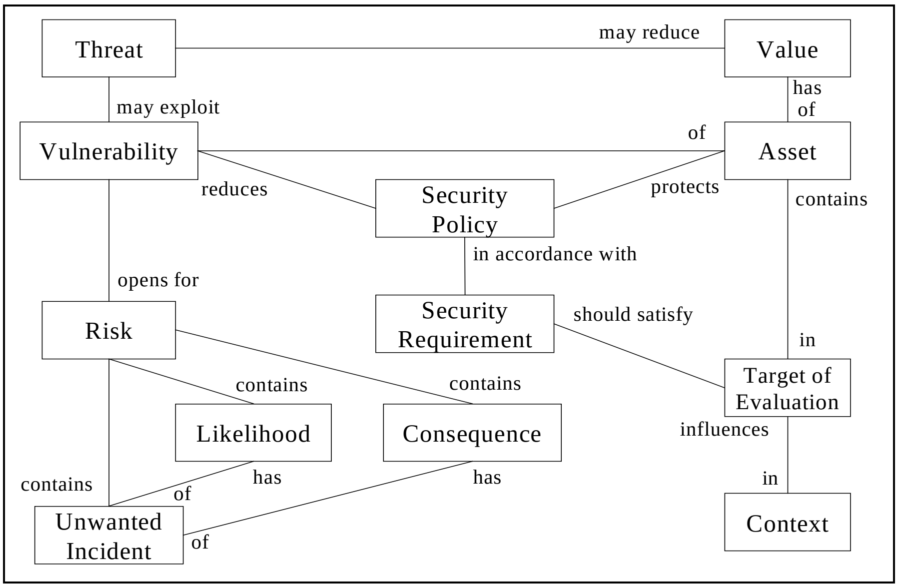

***Figure 3.* The CORAS terminology**

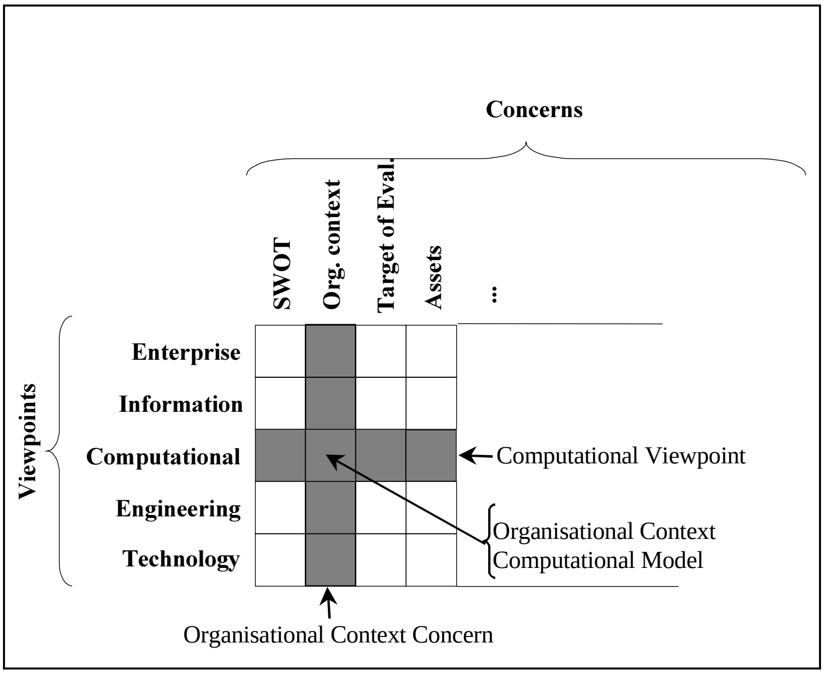

***Figure 4.* Relationship between viewpoints, concerns and models**

The CORAS risk documentation framework also provides libraries of reusable elements. These may be understood as specification fragments or patterns and templates for formalising risk assessment results capturing generic aspects suitable for reuse.

Finally, there are also plans to extend RM-ODP with a specialised risk assessment module containing a risk assessment process, risk assessment methodologies, international standards on which CORAS builds as well as integration formats for computerised tools.

5. The risk management process

The CORAS risk management process is based on AS/NZS 4360:1999 Risk Management [1] and ISO/IEC 17799:2000 Code of Practice for Information Security Management [19]. Moreover, it is complemented by ISO/IEC 13335: 2001 Guidelines for the management of IT-Security [17] and IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems [15]. As illustrated in *Figure 5*, AS/NZS 4360 provides a sequencing of the core part of the risk management process into sub-processes for context identification, risks identification, risks analysis, risks evaluation, and risks treatment.

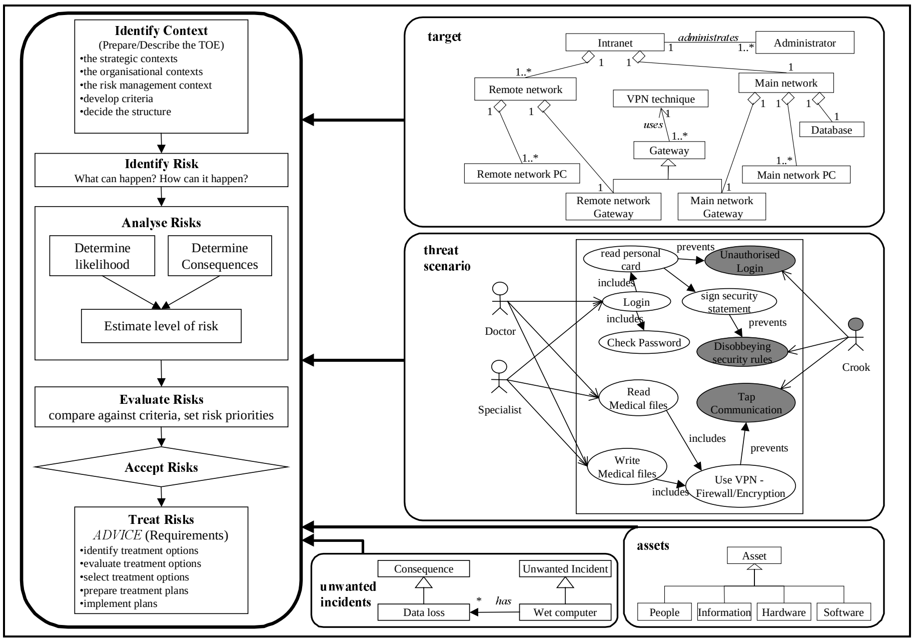

***Figure 5.* The role of UML in the CORAS risk management process**

For each of these sub-processes, the CORAS methodology gives detailed advice with respect to which models should be constructed, and how they should be expressed. *Figure 6* assigns concerns to the five sub-processes. Note that, even if the sub-processes are sequenced, AS/NZS 4360 is iterative and allows feedback.

Models expressed in the Unified Modelling Language (UML) [32] are used to support and guide the risk management process. The four diagrams to the right in *Figure 5* illustrate:

– specification of the target of evaluation with the help of a UML class diagram (aspect of the *target of evaluation concern* listed in *Figure 6*);
– specification of a threat scenario with the help of a misuse case diagram [31] (example element of the *threat scenarios concern* listed in *Figure 6*);
– specification of the assets to be protected with the help of a UML class diagram (aspect of the *assets concern* listed in *Figure 6*);
– specification of an unwanted incident with the help of a UML class diagram (example element of the *unwanted incidents concern* in *Figure 6*).

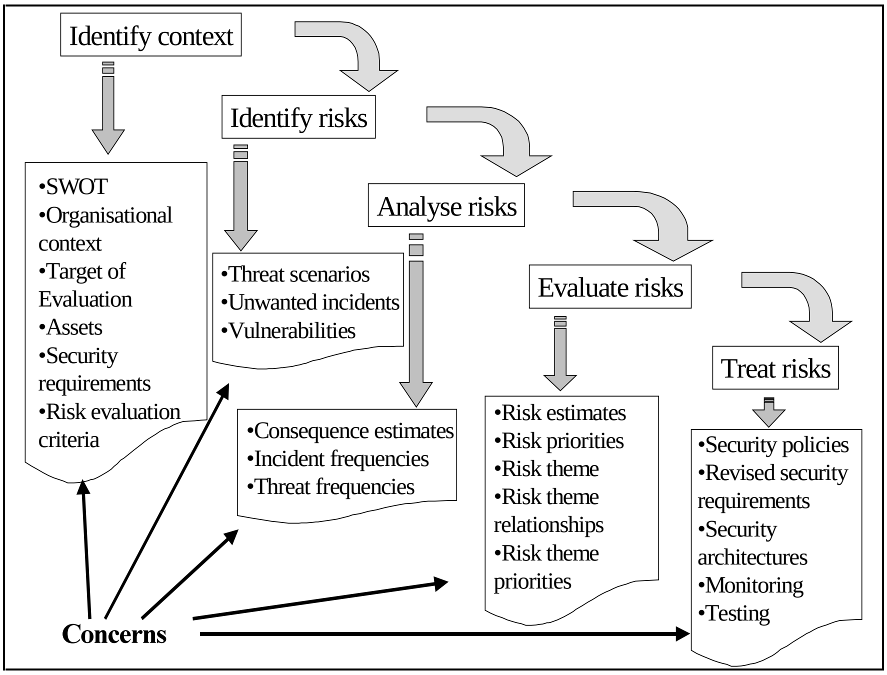

Figure 6. **The relationship between concerns and the risk management process**

6. The integrated risk management and system development process

The CORAS integrated risk management and system development process is based on an integration of the risk management process described above in the Unified Process (UP) [21]. In the following paragraphs we highlight the defining characteristics of this integrated process, as summarised in *Figure 7*.

In analogy to UP, the system development process is both stepwise incremental and iterative. In each phase of the system lifecycle, sufficiently refined models of the system 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 groups of stakeholders. Therefore, information in all viewpoints may be relevant to all phases of the lifecycle.

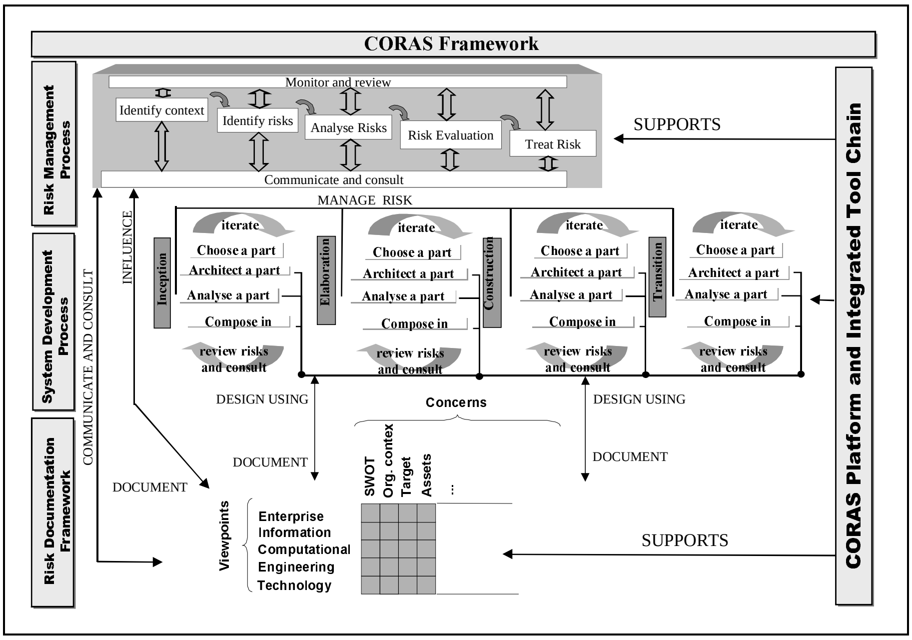

Figure 7. **The integrated risk management and system development process**

The risk management process follows the main iterations made in the system development process, as indicated in *Figure 8*. Each of the main iterations adds more detail to the target and the context of the assessment and previous results may need to be re-evaluated.

A set of agreed system requirements is one important outcome of the inception and elaboration phases. These requirements may be relevant to several viewpoints and can be described using a selection of different description methods, which are classified per concern. As one cannot expect that all security requirements are present from start, they have to be elicited. We anticipate that (appropriately adapted) model-based security risk assessment can also help with eliciting security requirements. However, risk assessment methods are traditionally designed to cope with unwanted incidents arising from design errors rather than specification problems related to missing requirements. For risk assessment to play a significant role in the elaboration phase, the CORAS risk assessment methods are being adapted to address requirement elicitation properly.

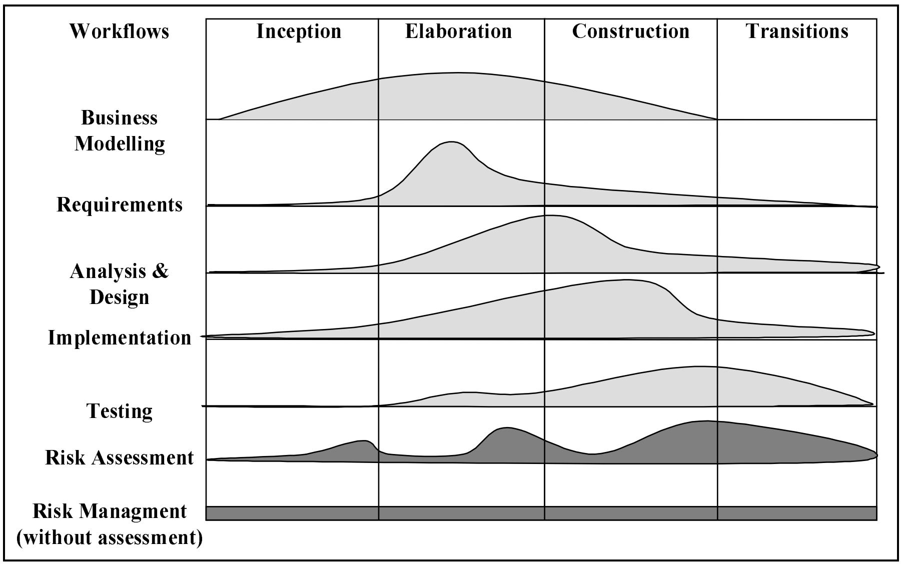

Figure 8. **Relating risk management to system development**

7. The platform for tool inclusion

A platform for tool inclusion based on data integration is under construction. Its internal data representation is formalised in the Extensible Markup Language (XML) [5]. Based on the Extensible Stylesheet Language Transformations (XSLT) [7], relevant aspects of this data representation may be mapped to the internal data representations of other tools (and the other way around). This allows the inclusion of sophisticated case-tools targeting system development as well as risk assessment tools and tools for vulnerability and threat management.

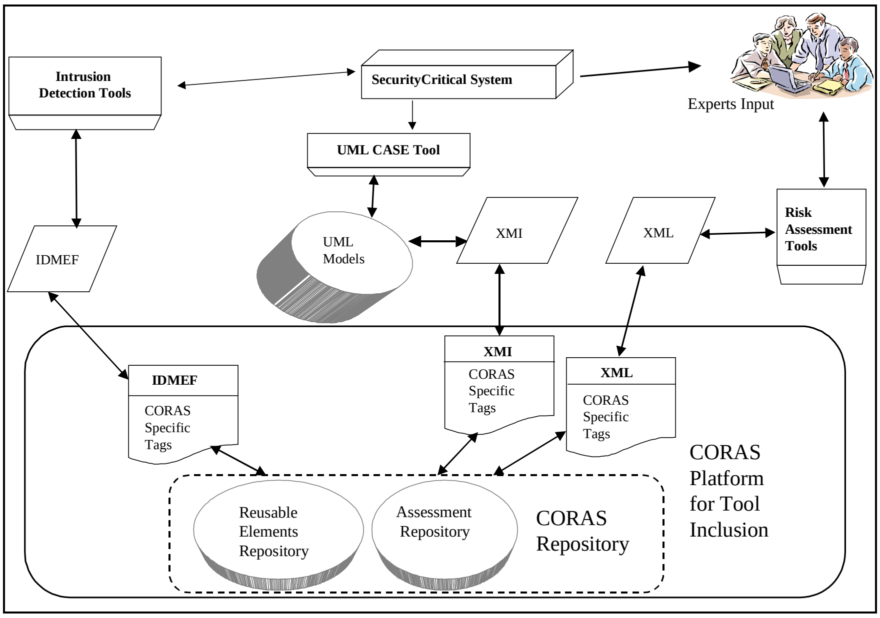

Figure 9. **The platform for tool inclusion**

As indicated in *Figure 9*, the CORAS platform is supposed to offer three interfaces for XML based data exchange:

- Interface based on the Intrusion Detection Exchange Format (IDMEF) [11]. IDMEF is an XML DTD targeting tools for intrusion detection and has been developed by the Intrusion Detection Working Group.
- Interface based on the XML Metadata Interchange (XMI) [29] standardised by the Object Management Group and targeting tools for UML modelling.
- Interface targeting risk assessment tools.

The CORAS platform will contain a repository divided into two sub-repositories: (1) The assessment repository storing the concrete results from already completed assessments and assessments in progress. (2) The reusable elements repository storing reusable models, patterns and templates from already completed risk assessments. Both sub-repositories mirror the decomposition into viewpoints and concerns illustrated in *Figure 4*.

8. The risk assessment methodology

The CORAS risk assessment methodology is a careful integration of techniques and formats inspired by HazOp Analysis [30], Fault Tree Analysis (FTA) [14], Failure Mode and Effect Criticality Analysis (FMECA) [4], Markov Analysis [25] as well as CRAMM [2].

The integrated risk assessment methods are to a large extent complementary. They address confidentiality, integrity, availability as well as accountability; in fact, as indicated by *Table 1*, all types of risks/threats/hazards associated with the target system can potentially be revealed and dealt with using these methodologies. They also cover all phases in the system development and maintenance process.

Table 1. The relevance of risk assessment methodologies

	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	
Evaluate risks	As input	Compare with criteria	Compare with criteria	Compare with criteria	
Treat risks	Identify treatment options	Address priorities	Address barriers and counter-measures	Support maintenance	Identify counter-measures

9. Conclusions

CORAS advocates model-based risk assessment. Model-based risk assessment is put forward as a means to improved efficiency of the risk assessment process as well as more reliable assessment results, since the understanding of the target of evaluation is enhanced by precise specifications of its structure and behaviour. Firstly, we argue that UML diagrams give a superior specification of system behaviour compared to free text or other informal formats. Secondly, a model-based risk assessment facilitates communication, both internally between the actors involved in the risk assessment and externally to the stakeholders. Thirdly, improved precision is not only of importance to understand the target of evaluation and the set of possible threats, but also for the documentation of the risk assessment results and the assumptions on which their validity depends. tructured documentation of risk assessment results and the assumptions on which they depend provides the basis for maintenance of assessment results as well as a component-based approach to risk assessment.

The CORAS project runs until July 2003. The development of the CORAS methodology and framework is guided by concrete experiences from two major trials, one within e-commerce and one within telemedicine. Both trials are divided into three trial sessions. The first trial session took place in January 2002, and the final trial sessions are planned for January 2003.

There are of course other approaches to model-based risk assessment. See for instance CRAMM [3], ATAM [8], SA [34] and RSDS [24]. The particular angle of the CORAS approach with its emphasis on security and risk assessment tightly integrated in a UML and RM-ODP is however new.

Contract-oriented specification has been suggested in many contexts and under different names. Within the RM-ODP community one speaks of contracts related to quality of service specification [12]. In the formal methods community there are numerous variations; the pre/post [13], the rely/guarantee [22] and the assumption/guarantee [28] styles are all instances of contract-oriented specification. Other more applied examples are the design-by-contract paradigm, introduced by Bertrand Meyer [26], and the UML based approach advocated by Mingins/Liu [27].

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 (CC) [18] represents the outcome of this work. The Common Criteria project harmonises the European “Information Technology Security Evaluation Criteria (ITSEC) [20]”, the “Canadian Trusted Computer Product Evaluation Criteria (CTCPEC)” and the American “Trusted Computer System Evaluation Criteria (TCSEC) and the Federal Criteria (FC)”. The CC is generic and does not provide methodology for risk assessment. CORAS, on the other hand, is devoted to methodology for risk assessment. Both the CC and CORAS places emphasis on semiformal and formal specification. However, contrary to the CC, CORAS addresses and develops concrete specification technology addressing risk assessment. The CC and CORAS are orthogonal approaches. The CC provides a common set of requirements for the security functions of IT products and systems, as well as a common set of requirements for assurance measures applied to the IT functions of IT products and systems during a security evaluation. CORAS provides specific methodology for one particular kind of assurance measure, namely risk assessment of security critical systems.

REFERENCES

[1] AS/NZS 4360:1999 Risk management.
[2] Atkinson, C., Bayer, J., Bunse, C., Kamsties, E., Laitenberger, O., Laqua, R., Muthig, D., Paech, B., Wüst, J., Zettel, J. Component-based product line engineering with UML. Addison-Wesley, 2002.
[3] Barber, B., Davey, J. The use of the CCTA risk analysis and management methodology CRAMM. Proc. MEDINFO92, North Holland, 1589–1593, 1992.
[4] Bouti, A., Ait Kadi, D. A state-of-the-art review of FMEA/FMECA. International Journal of reliability, quality and safety engineering 1:515-543, 1994.
[5] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. Extensible markup language (XML) 1.0 (Second edition). World Wide Web Consortium recommendation REC-xml, October 2000.
[6] Bræk, R., Gorman, J., Haugen, Ø., Melby, G., Møller-Pedersen, B., Sanders, R. Quality by construction exemplified by TIMe - the integrated methodology. Telektronikk 95(1):73-82, 1999.
[7] Clark, J. XSL transformations (XSLT) 1.0, World Wide Web Consortium recommendation REC-xslt, November 1999.
[8] Clements, P., Kazman, R., Klein, M. Evaluating software architectures: methods and case studies. Addison-Wesley, 2002.
[9] Clements, P., Northrop, L. Software product lines: practices and patterns. Addision-Wesley, 2001.
[10] CORAS: A platform for risk analysis of security critical systems. IST-2000-25031, 2000. (http://www.nr.no/coras/)
[11] Curry, D., Debar Merrill Lynch, H. Intrusion detection message exchange format (IDMEF). Working draft, December 28, 2001.
[12] Fevrier, A., Najm, E., Stefani, J. B. Contracts for ODP. Proc. ARTS97, LNCS, 1997.
[13] Hoare, C. A. R. An axiomatic basis for computer programming. Communications of the ACM, 12:576-583, 1969.
[14] IEC 1025: 1990 Fault tree analysis (FTA).
[15] IEC 61508: 2000 Functional safety of electrical/electronic/programmable safety related systems.
[16] ISO/IEC 10746: 1995 Basic reference model for open distributed processing.
[17] ISO/IEC TR 13335-1:2001: Information technology – Guidelines for the management of IT Security – Part 1: Concepts and models for IT Security.
[18] ISO/IEC 15408:1999 Information technology – Security techniques – Evaluation criteria for IT security.
[19] ISO/IEC 17799: 2000 Information technology – Code of practise for information security management.
[20] Information technology security evaluation criteria (ITSEC), version 1.2, Office for Official Publications of the European Communities, June 1991.
[21] Jacobson, I., Rumbaugh, J., Booch, G. The unified software development process. Addison-Wesley, 1999.
[22] Jones, C. B. Development methods for computer programs including a notion of interference. PhD-thesis, Oxford University, 1981.
[23] Kang, K. C., Cohen, S. G., Novak, W. E., Peterson, A. S. Feature-oriented domain analaysis (FODA) feasibility study. Technical report UMIAC-TR-21, SEI, 1990.
[24] Lano, K., Androutsopoulos, K., Clark, D. Structuring and design of reactive systems using RSDS and B. Proc. FASE 2000, LNCS 1783, 97-111, 2000.
[25] Littlewood, B. A reliability model for systems with Markov structure. Appl. Stat. 24:172-177, 1975.
[26] Meyer, B. Object-oriented software construction. Prentice Hall, 1997.
[27] Mingis, C., Liu, Y. From UML to design by contract. Journal of object-oriented programming, April issue: 6-9, 2001.
[28] Misra, J., Chandy, K. M. Proofs of networks of processes. IEEE transactions on software engineering, 7:417-426,1981.
[29] OMG-XML Metadata Interchange (XMI) Specification, v1.2, http://www.omg.org.
[30] Redmill, F., Chudleigh, M., Catmur, J. Hazop and software hazop. Wiley, 1999.
[31] Sindre, G., Opdahl, A. L. Eliciting security requirements by misuse cases. In Proc. TOOLS_PACIFIC 2000. IEEE Computer Society Press, 120-131, 2000.
[32] UML proposal to the Object management group, Version 1.4, 2000.
[33] Weiss, D. M. and Lai, C. T. R. Software product line engineering: a family based software engineering process. Addison-Wesley, 1999.
[34] Wyss, G. D., Craft, R. L., Funkhouser, D. R. The use of object-oriented analysis methods in surety analysis. SAND Report 99-1242. Sandia National Laboratories, 1999.