NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET

FAKULTET FOR INFORMASJONSTEKNOLOGI, MATEMATIKK OG ELEKTROTEKNIKK

HOVEDOPPGAVE

Kandidatens navn:	Ida Hogganvik
Fag:	SIF80/HO2 (2003 VÅR)
Oppgavens tittel (norsk):	AMBRA – Et arkitekturrammeverk for modellbasert risikoanalyse
Oppgavens tittel (engelsk):	AMBRA – Architectural framework for Model-based Risk Analysis

Oppgavens tekst:

The standard IEEE-1471 provides a conceptual framework for architecture description. The intended domain is software-intensive systems. IEEE-1471 is of relevance also outside its primary domain. For example, it is currently a foundation for OMG's work on model-driven architecture (MDA). An MDA is not a system architecture but rather an architecture for the system development process.
The CORAS project has developed a methodology for model-based security risk-analysis based on UML, UP and leading standards for risk management, risk analysis and security. In particular, the project has developed a UML profile for risk analysis as well as an RM-ODP inspired architecture for documentation of risk assessments defining 22 specialized cross-viewpoint perspectives, each documenting specific activities in the security risk management process.
The task is to develop a specialization of IEEE-1471, integrating terms from model-based risk analysis that is compliant with MDA.

Oppgaven gitt:	15.1.2003
Besvarelsen leveres innen:	11.6.2003
Besvarelsen levert:	11.6.2003
Utført ved:	NTNU
Veileder:	Ketil Stølen, SINTEF Tele og data

Trondheim,

Faglærer: Letizia Jaccheri

Abstract

This thesis builds on the generally accepted view that risk analysis during development and operation of systems facilitates security, safety, robustness and cost effectiveness.. We often need specialized risk analysis methodologies to fit a particular domain or system. We like to exploit the fact that much of the risk analysis terminology is common even if it is used on different domains, arranging for interoperability between domains since a methodology intended for one domain may have the ability to use parts of methodologies or tools from other domains.

The overall goal for this thesis is to develop an architectural framework for model-based risk analysis, serving as a foundation for development of model-based risk analysis methodologies. The framework will contain a terminology inspired by both risk analysis and architectural description based on models. This framework will be called AMBRA - Architectural framework for Model-Based Risk Analysis (AMBRA).

We will base us upon an existing risk analysis methodology called CORAS. We also believe a well described target system is important in risk analysis, helping the participants to understand the system, especially when dealing with the abstract nature of IT-systems. We will base ourselves on IEEE-1471 'Recommended Practice for architectural descriptions of software-intensive systems' for this purpose. This standard emphasizes representing the system according to its stakeholders' interests and by this ensuring relevance and right level of details in the models. Moreover its notion of system and overall terminology is very general and carries over to most domains.

One of the initial conditions for writing for this thesis was the desire to include the concepts from a standard for architectural descriptions in the context of CORAS and the most interesting standard to include was IEEE-1471. IEEE-1471 is more 'abstract' or 'high-level' than CORAS, which is a concrete methodology. It was therefore necessary to "extract" the essence of CORAS before connecting it to IEEE-1471. We compared IEEE-1471 with CORAS and decided on a common terminology for AMBRA, including a model of the risk management process. We decided to use a 'judgment study' and 'formal theory' to evaluate AMBRA against our success criteria.

We have developed an architectural framework for model-based risk analysis, including a general terminology combining risk analysis terms from CORAS with terms from IEEE-1471. The framework is not completed; some tasks are left for future work. We have also created a conceptual model of the risk management process with sub processes and activities including terms from IEEE-1471, expressed in UML and arranged for our work to be an extension, with focus on risk analysis, to a MDA framework when finished.

Preface

This is a diploma thesis written for the Department of Computer and Information Science, Faculty of Information Technology, Mathematics and Electrical Engineering, the Norwegian University of Science and Technology – NTNU.
The thesis has been written in cooperation with SINTEF Telecom and Informatics in Oslo during the spring semester 2003.

Professor: Letizia Jaccheri, Department of Computer and Information Science NTNU.
Supervisor: Ketil Stølen, Senior Scientist, SINTEF Telecom and Informatics, Oslo.

Thanks to Letizia Jaccheri, Ketil Stølen and the participants in my judgment study for valuable guidance and feedback.

Ida Hogganvik

Table of Content

List of Figures

1 Introduction

You have probably heard the term *risk analysis*, but what does it involve? The overall point of a risk analysis is to investigate *what can go wrong with your system* and *how can you prevent it?* Unless you know every little part of your system you need a structured way like a risk analysis to assess it. A useful vehicle in this risk analysis would be models describing your system and helping you to answer your questions. The models should describe important features about your system in a way you understand and with proper level of details.

This thesis aims to integrate risk analysis in the setting of system development and system architecture. This thesis defines an architectural framework for model-based risk analysis. Thereby it provides developers of model-based risk analysis methodologies a framework to work within. This ensures interoperability between methodologies for risk analysis. It also facilitates a tight integration of risk analysis in system development.

In the following we motivate why we consider proper architectural descriptions in combination with risk analysis as important. We start with a general explanation of risk analysis and architecture, followed by a discussion about risk analysis of IT-systems and architecture of IT-systems.

1.1 Risk analysis in general

Every time we are engaged in the development process or operating phase of something we are interested in achieving the best result as possible. Unfortunately there is always a possibility for unwanted incidents to occur, that may become a problem for our plans. To prevent failure of our project we need to assess and monitor all factors which potentially may cause harm. For this purpose we conduct risk analysis.

Let us start with a small and naïve example, known to everyone who has been on a beach If we want to build a sandcastle we face several threats which may harm our construction. Someone might destroy it by tramping on it, a wave might wash it away or weather like rain and wind might cause it to crumble away:

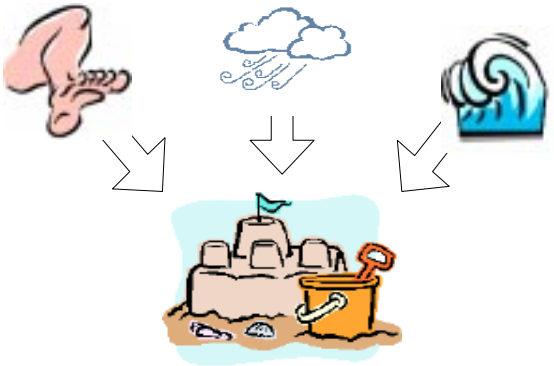

Figure 1-1 Risks for a sand castle

The threats could probably destroy our construction, but by analyzing them we can take precautions to secure the sand castle. This survey of the threats and their effects could be a start of a risk analysis, where the goal is to find treatments for the risks, i.e. ways of reducing the probability for them to happen.

A way of treating the risk of someone tramping on the castle is building it on a less crowded place. The risk for it being washed away by a wave may be treated by building it far away from the waterline. The risk for it being destroyed by weather is difficult to treat. We might build it in some shelter from wind and rain, but it is not possible to make a sandcastle on a beach strong enough to resist rain and wind.

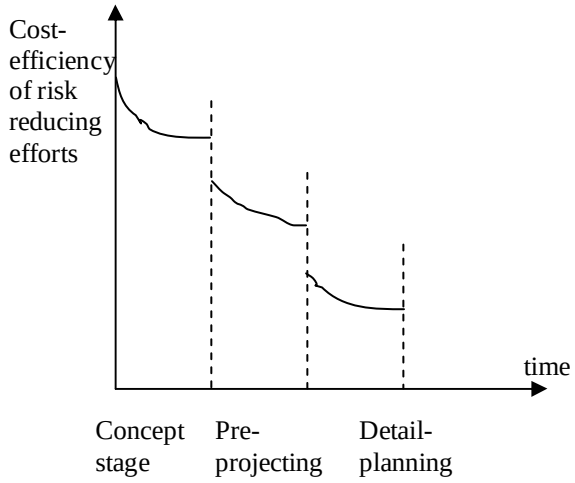

Figure 1-2 Cost efficiency in efforts to reduce risks

Another situation where it is important to get an overview of possible risks is in a real construction situation. We might look upon a house as several components which must fit in with each other in an acceptable way. The constructor needs full overview of the requirements of each component, and must not overlook anything or make wrong estimates as these errors may be costly to fix later in the construction phase.
The constructor follows methods and uses materials he reckons give good results, but this is not a guarantee for a perfect building. In order to achieve success he needs thoroughly planning already in the early concept stage where the house is designed. Making changes to the drawing in this phase are cheap and less complex than in later phases as illustrated in Figure 1-2 from [Rausand 91]. It is rarely optimal with improvised ad hoc solutions in later phases as implies more and unnecessary work. If we are to return to our sandcastle example, Figure 1-3 and Figure 1-4 illustrates how risk analysis is used in different stages of the process, before building the sandcastle and after. The cost of not considering possible risks in early phases, i.e. before building the sandcastle, is that the work of building has to be done twice. This is one of the reasons for why it is important to continuously check whether the requirements conform to the construction schedule during all phases.

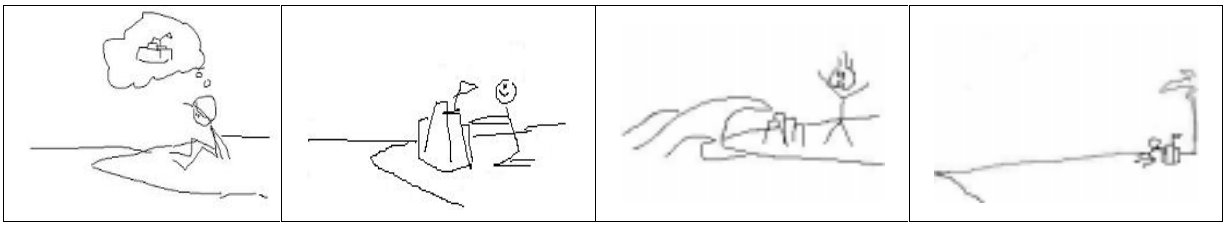

Figure 1-3 Risk analysis after building

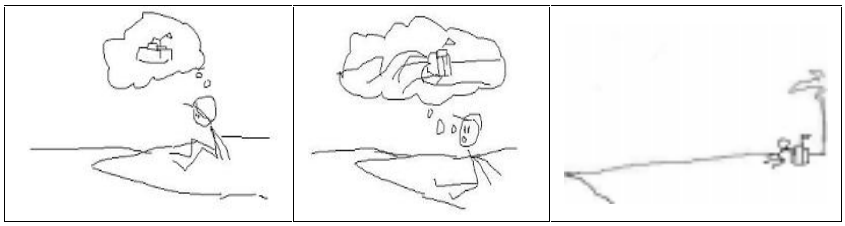

Figure 1-4 Risk analysis before building

To ensure that every concern is assessed before entering the next stage of constructing, the team may conduct a risk analysis. A risk analysis in the constructing context controls whether the physical construction of the building (the target system of the risk analysis) conforms to the approved cost and risk estimates, existing requirements and premises. Especially they evaluate possible modifications and their consequences with respect to other systems in the

building. If water pipes are placed in a way that they are exposed to frost there is danger for the water freezing, something which may be caught in a risk analysis of the plumbing system seen as an independent component of a house.

So in which situations except in the construction phase is it appropriate to do risk analysis of a system? In [Rausand 91] the situations where risk analysis normally is used are classified according to which *phase* the system is in:

The planning phase of the system:

In a construction context this would be the phase where you design the building according to the acquirer's requirements:

- Risk analysis is used as design tool (helps specify the requirements based on the risk analysis)
- Risk analysis is used as control mechanism (to control risk level according to acceptance level)

The operating phase of the system:

If we continue with examples from a construction context, this phase would be when the building is finished, and either a problem has occurred or the acquirer likes to make changes to the building:

- To analyse effects of changes made to the system
- To analyse causes to problems with the system

An example of risk analysis where a problem has occurred during the system's operating phase is described in Figure 1-5. In this case the system is a person, and the operating phase is the daily life of the person:

A patient visits the hospital with an injured wrist. The doctor does a kind of medical risk analysis of the system (i.e. person) which answers the following questions:

1. What is the injury?
An injured wrist on the left side.
2. How does the injury affect the person?
The left arm of the patient does not function as normal because of a broken wrist.
3. What incident caused the injury?
A fall on snowboard.
4. Why was the outcome a broken wrist?
The wrist did not stand the weight of the patient supporting himself in the fall.
5. How to repair the injury?
Put wrist in plaster according to a special method.
6. How to prevent injuries like this in the future?
It is not possible to make a wrist strong enough to resist a fall like this, but it is possible to strengthen it by using wrist protection.

Figure 1-5 Risk analysis example

In order to do a successful risk analysis the people involved need to have a clear understanding of how the target of the analysis works. In a risk analysis where the target is a part of a house, understanding the system could be provided through detailed drawings of the building developed by the architect.

1.2 Architecture in general

It is possible to claim that everything that consists of components has an architecture, i.e. a structure of how the components relate to each other. If we were to describe the visible 'top-level' architecture of a human being we could make a drawing of a person with head, arms and legs. A general definition of *architecture* is "*the fundamental structure of an entity and the interrelationships among its parts. The architecture of an entity captures its essential features and principal properties, it also captures its main components, how they are structured, how they interact, and the principles guiding their design and evolution*" [Runde,Stølen 03].

A physical thing like a building is the most obvious example of a something with an architecture. The way doors and windows are placed on the different floors, where the walls divide the floor into rooms etc is the architecture of a building. It is possible to describe an architecture of a building in different notations, but the goal is always to illustrate the structure in a way that other people could understand and find useful. Figure 1-6 presents examples of architectural descriptions of buildings.

Figure 1-6 Examples of architectural description

When the architect draws the building, he or she decides where walls, doors, water pipes and electrical lines will be placed in order to function properly. The drawing will be a common point of reference for the architect, the buyer, the contractor and possible subcontractors. In a risk analysis context, architectural descriptions could be an important vehicle for the participants in the challenge of understanding the target structure and configuration. To make accurate and understandable models of the system is crucial and can, as we discuss later, be a major challenge when dealing with more abstract systems like IT-systems.

1.3 Risk analysis of IT-systems

In a way we have gradually become more and more dependent on IT-systems. They are an important part of our infrastructure like the telephone system, e-mail, online bank services etc. In many cases we are totally dependent on IT-systems and will not manage without them. This leads to high demands to developers of IT-systems, as the systems have to be highly reliable and robust. An IT-system resembles a house in that it is composed of several components, but building IT-systems differs in many ways from constructing houses. It is difficult to see or touch the physical organization of an IT-system, therefore making it more difficult to investigate the effects of decisions made in development or maintenance. When developing an IT-system, changing or repairing it, it is essential to analyse what the effects of your decisions will be. This is why we need well defined methods for risk analysis of IT-systems. A risk analysis helps to discover the effects a change to a component may have on the dependent components. The goal of the risk analysis is to get an overview of threats, risks and vulnerabilities of the system helping you to design a robust and reliable system.

The international standardization organisation (ISO) defines standards for risk analysis related to IT-systems and provides this definition of risk analysis: *The process of identifying security risks, determining their magnitude, and identifying areas needing safeguards.* [ISO13335]

The general purpose of a risk analysis is to evaluate a system or process in order to discover its weaknesses and possible ways of improvement. In relation to IT-systems we often focus on security and limitations on use or capacity in order to make the system as robust, secure and user friendly as possible.

In order to do a successful risk analysis the involved persons need a good understanding of how the target system works and what parts it consists of. This is as already mentioned especially challenging for IT-systems because of their abstract nature. Creating detailed and understandable descriptions of an IT-system's architecture requires more effort and knowledge than with other systems.

1.4 Architecture of IT-systems

In the context of building a house, the 'architecture' is the structure and organization of the materials and components comprising the house. In an IT-system development context 'architecture' could be the way hardware components are used together or the abstract structure of program code.

When explaining what the architecture of an IT-system is, it is natural to start with an example of one of the most widespread used IT-systems, -the PC. A common 'PC-system' consists of parts like mouse, keyboard, computer, monitor and possibly loudspeakers. Both mouse and keyboard are used to provide information to the computer, and information from the computer is passed on to the user through monitor and speakers.

Figure 1-7 a simple IT-system

The simplest notation for describing architecture of an IT-system is the so-called 'boxes and arrows'-notation. The intention is to show the different parts (boxes) of the system, and how they interact (arrows). If we were to make a model of our simple IT-system like in Figure 1-7 using 'boxes and arrows'-notation it could look like Figure 1-8. In addition to displaying each of the elements in the system as boxes, the arrows denote information flow. Information flows from the user via mouse and keyboard into the computer, and the computer sends information back through monitor and speakers.

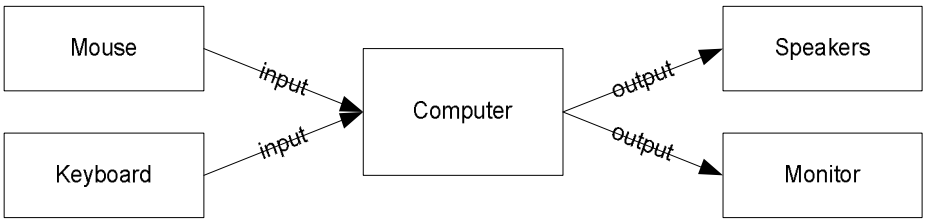

Figure 1-8 architectural description of a simple IT-system

There exist several other notations and conventions for describing IT-systems, which focus on different aspects like dataflow, the different states of a system, interaction between components or diagrams showing typical use cases. For more information about these

notations we refer to Unified Modelling Language (UML) [UML]. The most important is that the models are made in a way which other people can understand and therefore be used as a foundation for system evaluation.

In this thesis we focus on a particular standard for architectural description of IT-systems called ”IEEE-1471 Recommended practice for Architectural Description of Software-intensive Systems” [IEEE1471]. A more detailed presentation of this will be provided in Chapter 2.

1.5 The overall aim and rationale

There is a common understanding that risk analysis may contribute to discover weaknesses of a system in an early phase of the system development process or during operation. This could reduce the costs by avoiding expensive redesign if the risks were discovered in later phases. It is also considered important to conduct risk analysis before changing a system and when investigating causes of errors or accidents. Our focus will be on risk analysis in connection with IT-systems.

This thesis builds on the overall hypothesis that risk analysis of systems during development and operation is necessary in order to achieve secure, safe, reliable and cost-effective systems.

In order to emphasize the different systems characteristics developers often need to specialize risk analysis methodologies to fit a particular domain or system. A telephone system for emergency calls has for instance much higher demands to availability than a system recording all the books in a library. We like to exploit the fact that much of the risk analysis terminology is common even if it is used on different domains. By collecting this terminology in a general framework, it will function as a foundation for the development of risk analysis methodologies. We will by this arrange for interoperability between domains since a risk analysis methodology intended for one domain have the ability to use parts of methodologies or tools from other domains.

The goal is to develop an architectural framework with a terminology influenced of both risk analysis and the conventions for architectural descriptions focusing on the use of models. This framework will be called AMBRA - Architectural framework for Model-Based Risk Analysis (AMBRA). In addition the framework will arrange for future inclusion of a guideline and a library for reusable components (see Chapter 7).

To ensure the inclusion of all relevant concepts from risk analysis in our terminology we will base us upon an existing model-based risk analysis methodology called CORAS (see Section 2.4). By using a model-based risk analysis methodology as a foundation for our work we strengthens IEEE-1471’s focus on models.

As already stated we intend the terminology to include terms from a standard for architectural description. We believe a well described target system is important for the risk analysis results by helping the participants understand the system, especially when dealing with the abstract nature of IT-systems. The models of how a system is organized, i.e. the architecture of the system, may be described according to different types of conventions. We make use of IEEE-1471 because its notion of system and its terminology is general and carries over to most domains.

Models of same type of systems, used in different risk analysis, may have elements in common in the same manner as terminology used in different domains. Therefore it is possible to reuse models of same type system if they are stored in a structure optimized for intuitive and efficient retrieval. If one conducts multiple risk analysis of the same type of system, one may reuse previous models of the system and thus reduce costs and effort. Also known threats and weaknesses of one type of system may be reused in risk analysis on similar systems. This motivates for the importance of storing results from risk analysis in a structure which makes it easy to efficient retrieve relevant information for the risk analyst.

AMBRA will define the phases the risk management process should include and the concepts a risk management activity should document, but not impose the exact risk analysis techniques to use.

The overall goal for this thesis is to develop an architectural framework for model-based risk analysis, serving as a foundation for development of model-based risk analysis methodologies. The framework will contain a terminology inspired by both risk analysis and architectural description based on models. Our intention is not to develop the complete AMBRA, but to centre on the terminology part.

1.6 Our methodological approach

This Section is meant to provide the reader with an overview of our methodological approach. Details about each stage in the process are left to subsequent Sections.
One of the initial conditions for writing for this thesis was the desire to include the concepts from a standard for architectural descriptions in the context of CORAS. The most interesting standard to include was IEEE-1471, which is described with a useful and understandable conceptual specification. IEEE-1471 is recognised as a “framework for defining architecture” i.e. more abstract than CORAS which is a concrete methodology. Hence it was necessary to “extract” the essence of CORAS into a more abstract specification of model-based risk analysis, before connecting it with the conceptual specification in IEEE-1471. Obviously it would be difficult to abstract the whole methodology, therefore we decided to concentrate on the documentation framework in CORAS. This framework is described with a conceptual specification similar to the model in IEEE-1471 and appropriate for our use. Then we started comparing the terms and concepts in IEEE-1471 with CORAS, deciding which terms and interpretations to use in AMBRA. After settled for a complete terminology for AMBRA, we extended the conceptual specification of IEEE-1471 with the specification abstracted from CORAS forming a conceptual specification of AMBRA. This conceptual specification also includes a description of the risk management process according to CORAS.

Initially we formulated five success criteria supporting a superior success criterion for AMBRA, stating our requirements to AMBRA. Several different research strategies were considered in order to find the appropriate ones to evaluate these criteria. We decided to use two types of strategies, a judgment study where a set of people ‘judge’ AMBRA and formal theory based on logical and sound argumentation. In addition to this evaluation of AMBRA, there is a Chapter discussing different general aspects of AMBRA in the end of the thesis.

1.7 Thesis structure

This is a presentation of the structure of this thesis.

- Chapter 1 "**Introduction**"
  This Chapter introduces the motivation for the development of AMBRA, describing *risk analysis* and *architecture* to provide a basic comprehension of these concepts necessary for understanding the rest of the thesis

- Chapter 2 "**Background and Overall Context**"
  Provides background information about the different foundations for developing AMBRA. We believe the reader will find it useful to have some knowledge about the basic frameworks AMBRA will rely on.

- Chapter 3 "**Problem analysis**"
  Gives a detailed survey of the motivation for this thesis and describes how we plan to validate AMBRA when finished.

- Chapter 4 "**Relationships in terminology between IEEE-1471 and CORAS**"
  Discusses the differences and similarities between the two frameworks.

- Chapter 5 "**Formalizing the AMBRA Terminology**"
  Gives the full description of the AMBRA terminology specified in UML.

- Chapter 6 "**Evaluation of AMBRA against Success criteria**"
  Different aspects of AMBRA are evaluated against the initial criteria for success

- Chapter 7 "**Discussion**"
  An evaluation of the decisions made during development of AMBRA.

- Chapter 8 "**Conclusions**"
  Provides the reader with the main findings from the development of AMBRA, subjects for further work with AMBRA and other related work.

- **References**
  References to sources of information used in this thesis.

- **Appendix**
  - Questionnaire from the Judgment study. As a part of the validation motivated in Chapter 3 a judgment study was conducted to assess AMBRA
  - Introduction to Object Management Group (OMG)
  - The five RM-ODP viewpoints

Figure 1-11 gives a description of the thesis structure and the arrows indicates alternative sequences of reading the thesis. If the reader is not interested in the motivation for developing AMBRA, only in the framework itself, he or she may skip Chapter 3 and 4.

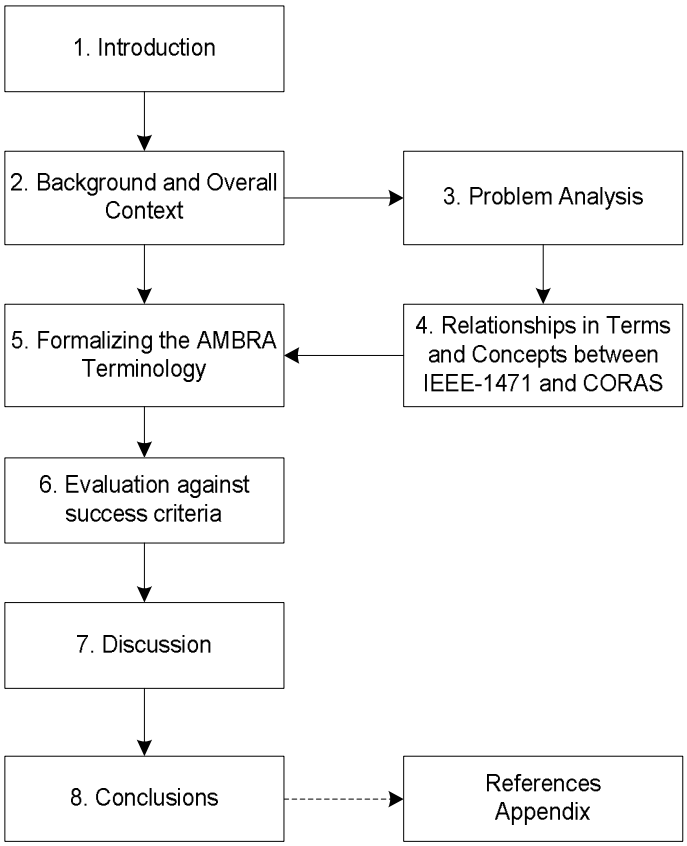

Figure 1-9 Reading guide to the thesis

2 Background and Overall Context

This Chapter gives an introduction to architectural frameworks in general and how AMBRA relates to the specific frameworks like CORAS, IEEE-1471 and MDA. This is followed by a presentation of the core concepts of each of the frameworks. In addition we provide a Section about UML and an introduction to the different levels of architecture, 'the architectural hierarchy'. Even if you are familiar with some of these topics, we still recommend you to read this Chapter as it provides useful information for the rest of the thesis.

2.1 An introduction to architectural frameworks

In our previous work we have only discussed architecture in relationship with physical things. However, also conceptual entities may have architecture like e.g. an enterprise, the structure of a system development process or the internal organization of a database. In the introduction 'architecture' was defined as "*the fundamental structure of an entity and the interrelationships among its parts. The architecture of an entity captures its essential features and principal properties, it also captures its main components, how they are structured, how they interact, and the principles guiding their design and evolution.*" As should be clear, this definition does not limit itself to only physical objects.

An example of a non-physical-object with an architecture is the process of software development. For illustration purposes we use the 'Waterfall method' [Sommerville 95] for system development, well-known by many in the IT- industry.
This process may be divided into several stages or components like 'analysis', 'design', 'implementation', 'testing' and 'maintenance'. The order in which we execute these components is not arbitrary, the initial requirements must exist before design, and testing requires an implemented system.
A possible architectural description of the stages in this process is illustrated in Figure 2-1; boxes are components and arrows denote information flow:

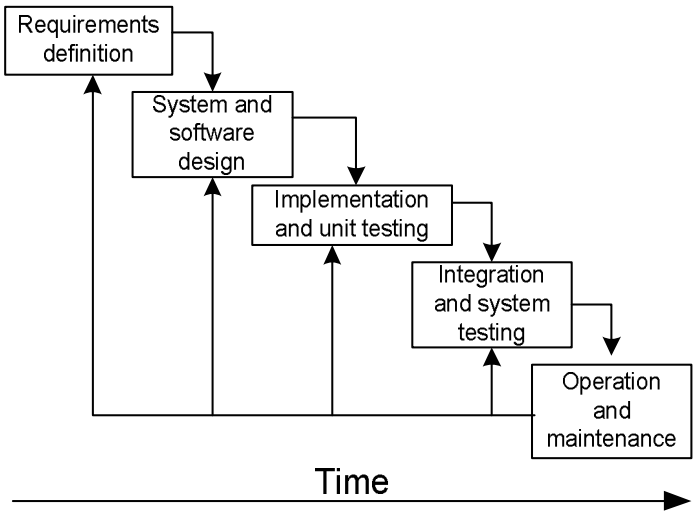

Figure 2-1 Example of architecture of a system development process

This architectural description could be an example of a simplified framework for how to build a software system. This framework describes the coarse features of the development process, i.e. the different components and the dependencies between them.

What is a framework really?
Merriam-Webster's Collegiate Dictionary [meriamwebster.com] defines framework as:
1: a) a basic conceptual structure (as of ideas) <*the framework of the constitution*>
b) a skeletal, openwork, or structural frame
2 : Frame of reference, which is defined as:
1: a set of ideas, conditions, or assumptions that determine how something will be approached, perceived, or understood <*a Marxian frame of reference*>
Merriam-Webster's Collegiate Thesaurus [meriamwebster.com] claims that framework is synonymous with structure; a structure is defined as 'something made up of more or less interdependent elements and having a definite organizational pattern <*the complex bureaucratic structure of modern government*>.

An architectural framework is therefore a description of the structure of architectures, providing a point of reference for the architecture builders.

2.2 AMBRA and Related Frameworks

AMBRA will be related to the following architectural frameworks: the *Recommended Practice for Architectural Description of Software-intensive Systems* (IEEE-1471) [IEEE1471].This is a framework providing guidelines for how to describe the structure of an IT-system and AMBRA should include a specialization of this standard. IEEE-1471 is further discussed in Section 2.3.

AMBRA is indirectly related to the *Reference Model for Open Distributed Processing* (RM-ODP) [ISO10746] (see Figure 2-2). RM-ODP is a framework with guidelines for how to create system development methods for developing distributed systems. IEEE-1471 is recognized as a documentation vehicle for RM-ODP, used as an inspiration for how to document the system. RM-ODP is briefly discussed in Section 2.3.1.1, for extensive information about practical use of RM-ODP we refer to [Putman01].

AMBRA is inspired by the *Model-driven Architecture* (MDA) [omg.org/mda]. In [Runde,Stølen 03] the authors have made an effort to explain MDA: *"a framework for defining system architecture development methodologies by giving directions for making architectures, serving as a tool-box for system developers and system architects, and facilitating integration and interoperability between systems"*. MDA is based on the use of models of the system. This intends to facilitate communication between people involved in the process and is by many recognized as the new direction for system development. IEEE-1471 has been a foundation for the work with MDA. Our work will focus on the use of models inspired by MDA which is described in Section 2.5.

AMBRA will include an abstraction of CORAS, which is a more concrete framework than the previous three. CORAS is a complete methodology for risk analysis, i.e. a detailed framework for how to conduct risk analysis. This thesis will include concepts related to risk analysis based on parts of CORAS. Particularly interesting is the structure CORAS use for documenting results from a risk analysis which is based on RM-ODP. The reader will be introduced to CORAS is in Section 2.4.

Figure 2-2 illustrates the relationships between the different frameworks [1]:

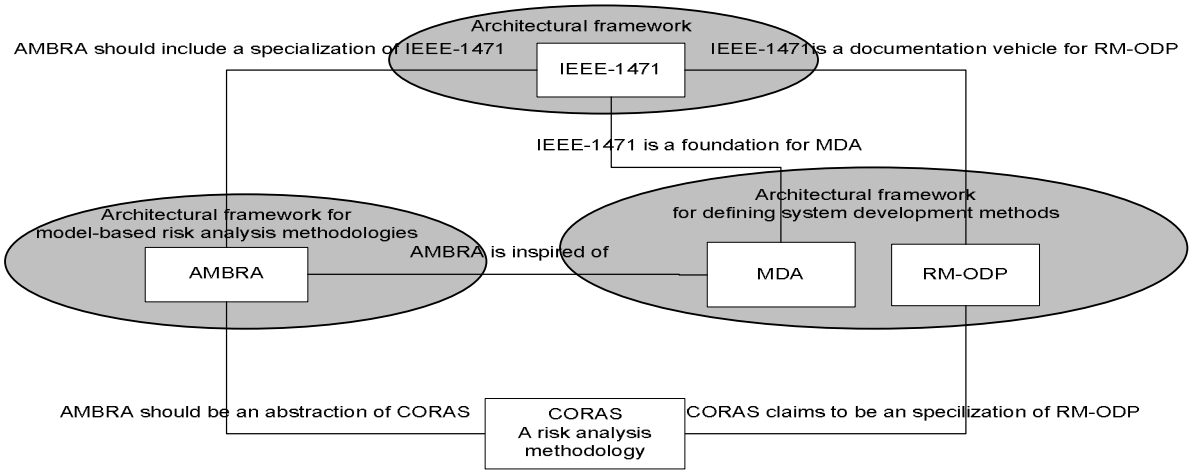

Figure 2-2 Frameworks and how they relate

So, in the end,-what is AMBRA meant to be?

The result of this work will be an architectural framework for model-based risk analysis. With foundation in a well defined risk analysis methodology (CORAS) and an internationally accepted standard for architectural description (IEEE-1471), the goal is to create an architectural framework for the development of model-based risk analysis methodologies. We believe this will facilitate the development by defining relevant concepts and their dependencies which a method should include, i.e. an architecture for this type of methodology. The framework will not define how the analysis should be performed, i.e. which specific risk analysis techniques to use.

2.3 IEEE-1471 - 'Recommended Practice for Architectural Description of Software-intensive Systems'

IEEE-1471 "Recommended Practice for Architectural Description of Software-intensive Systems" [IEEE1471] is defined by the Institute of Electrical and Electronics Engineers Inc (IEEE) [ieee.org]. A so-called '*Recommended practice'*, is one of the types of IEEE-standard. The adoption and interpretation of a recommended practice is the responsibility of the using organization, but the authors hope it will evolve to a standard with wide use. The standard is published by the IEEE Software Engineering Standards Committee (SESC) [SESC] whose vision is to develop a family of products and services based on software engineering standards.

The standard focuses on how to describe the architecture of software-intensive systems, by providing a basic framework for the content of such a description. Software intensive systems are systems where software contributes essential to the design, construction, deployment and

[1] The boxes illustrate frameworks and lines denote relationship between them. The type of relationship is given by the associated line name, and the shaded areas denote the types of frameworks. The frameworks are situated according to their level of abstraction where IEEE-1471 as the most abstract and general one is placed on the top.

evolution of the system. However, the overall terminology is general and carries over to systems in general.

According to IEEE-1471 architectural descriptions could be used for:

- Expression of the system and its evolution
- Communication among the system stakeholders
- Evaluation and comparison of architectures in a consistent manner
- Planning, managing, and executing the activities of system development
- Expression of the persistent characteristics and supporting principles of a system to guide acceptable change
- Verification of a system implementation's compliance with an architectural description
- Recording contributions to the body of knowledge of software-intensive systems architecture

This recommended practice claims then to facilitate the expression and communication of architectures and thus be a foundation for quality and cost gains through standardization of elements and practices for architectural description.

IEEE-1471 defines architecture as "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution".

The recommended practice emphasizes that systems have an architecture that can be recorded by an architectural description, and every system has one or more stakeholders with some concerns about the system.

An IEEE-1471 compliant architectural description must (in brief):

- Define the stakeholders for the system (whose architecture is being described)
- Define those stakeholders concerns
- Select a set of viewpoints for the description
- Describe the system in a series of models
    - The models are organized into views
    - Each view much be conformant to exactly one selected viewpoint
- Some minor ancillary requirements

The architectural description recommended in the IEEE-1471 is organized by *views*, each view conforms to one of the *viewpoints*. Each view addresses one or more of the concerns of the systems stakeholders. A view is a representation of the entire system from a single perspective and may consist of one or more architectural models (e.g. figure/diagram).

Each architectural model is developed using methods established by its associated architectural viewpoint. The recommended practice is meant to reflect the generally accepted trends in practices for architectural descriptions, and provide a technical framework for further evolution in this area. Figure 2-3 from IEEE-1471 provides a summary of the key concepts and their inter-relationships like view, stakeholder and architectural description[2].

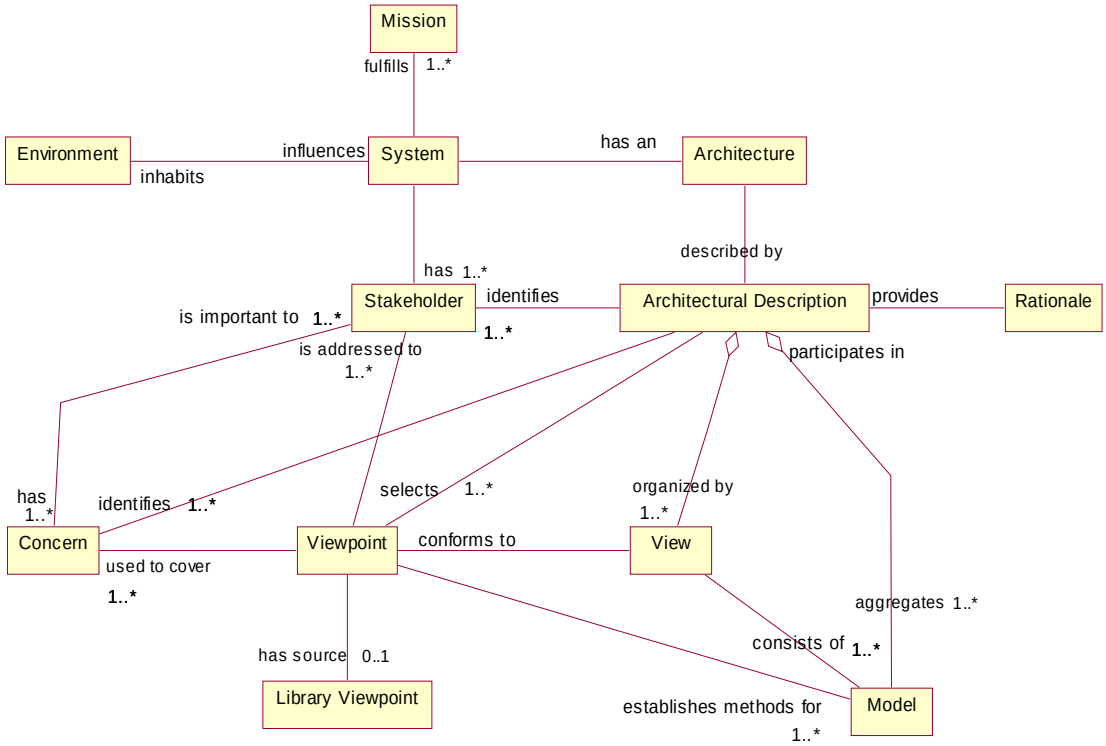

Figure 2-3 the conceptual model of an architectural description

One of the intentions with IEEE-1471 is to help to ensure that architectures take into consideration the context of the system of interest. It also helps to ensure that architectural descriptions will be structured to meet the needs of the multiple stakeholders of the system.

The intended users of this standard are divided into two classes:
The first class of intended users of IEEE-1471 are stakeholders in system development and evolution, typically:

- Those that use, own, and acquire the system (users, operators, and acquirers or clients).
- Those that develop describe and document architectures (architects).

[2] According to IEEE-1471 Figure 2-3 presents 'the concepts in the context of an architecture for a particular system and an associated architectural description. This is not to assume or require that a system has only one architecture or that there is only one architectural description depicting that architecture.
In the Figure, boxes represent classes of things. Lines connecting boxes represent associations between things. An association has two roles (one in each direction). A role can optionally be named with a label. The role from A to B is closest to B, and vice versa. For example, the roles between SYSTEM and ENVIRONMENT can be read: A SYSTEM inhabits an ENVIRONMENT, and an ENVIRONMENT influences a SYSTEM. In the Figure, roles are one-to-one unless otherwise noted. A role can have a multiplicity, e.g., a role marked with "1..*" is used to denote many, as in a one-to-many or many-to-many association. A diamond (at the end of an association line) denotes a part-of relationship. For example, VIEWS are a part of an ARCHITECTURAL DESCRIPTION.' This notation is from the Unified Modelling Language Specification [UML].

- Those that develop deliver and maintain the system (architects, designers, programmers, maintainers, testers, domain engineers, quality assurance staff, configuration management staff, suppliers and project managers or developers).
- Those who oversee and evaluate systems and their development (chief information officer, auditors, and independent assessors).

The secondary class of users are:

- Those involved in the enterprise-wide infrastructure activities that span multiple system developments, including methodologists, process and process-improvement engineers, researchers, producers of standards, tool-builders and trainers.

The standard does not recommend a specific architectural description language (ADL). The reason is that there is no consensus among the broad spectrum of potential user-communities for IEEE-1471 on any ADL. In fact there is no consensus on what the requirements for an ADL should be. There is, however, a consensus that architects should choose their notations carefully, document these choices, and be encouraged to use well-founded ADLs. The philosophy of IEEE-1471 is to use the right tool for the job: use notations appropriate to addressing the architectural concerns that specifically arise for the system at hand.

2.3.1 Competitors to IEEE-1471

There exist other standards for architectural descriptions which may be viewed as competitors to IEEE-1471. In the following we provide a brief description of some well-known: RM-ODP, TOGAF and C4ISR-AF. In common for they all is the use of concepts from the terminology in IEEE-1471 extended with more details and often a concrete methodology.

2.3.1.1 'RM-ODP'

The Reference Model for Open Distributed Processing systems (RM-ODP) [ISO10746] defines concepts and structuring rules for how to describe the architecture of distributed systems in an object-oriented manner.
RM-ODP is internationally accepted by the International Standards Organization (ISO) and the International Telecommunication Union (ITU).

RM-ODP is much used in mission-critical systems industries such as e.g. telecommunications, health care and financial services industry. The standard is also used as foundation for the Object Management Group's work with Object Management Architecture (OMA), used for defining enterprise architectures and software architectures.
Like IEEE-1471, RM-ODP also uses viewpoints, but limits itself to five different types. According to RM-ODP a viewpoint is a part of the specification of a complete system, including information relevant to some particular area of concern.
The five types of viewpoints all complement each other and allow for a thorough and complete description of the system. The term "system" is understood in a general, system-theoretic sense. This is allows for the interpretation of this term to vary from a specific (hardware or software) component of the distributed information processing system of an organization, to the whole information processing system, even an enterprise or organization.

It is possible to claim that IEEE-1471 is a more abstract reference framework than RM-ODP. RM-ODP has five predefined viewpoints, while IEEE-1471 only specifies that an architectural description should include one or more views and viewpoints.
IEEE-1471 is a kind of documentation vehicle for RM-ODP by providing an organizing framework for the architectural descriptions. RM-ODP can be seen as additional normative

restrictions on IEEE-1471convetions for architectural descriptions. For further information about use of RM-ODP see [Putman01].

2.3.1.2 'TOGAF'

The open Group Architectural Framework (TOGAF) is an architectural framework that includes both architectural descriptions and the methodologies used to produce these.
According to the Open Group, TOGAF enables you to "design, evaluate, and build the right architecture for your organization"
TOGAF consists of two main parts:
The TOGAF Architecture Development Method (ADM), which explains how to derive an organization-specific IT architecture that addresses business requirements. This includes:

- A way of developing the architecture
- Architectural views
- Examples and linkages to practical case studies
- Tools for architecture development

The TOGAF Foundation Architecture: an architecture of generic services and functions that provides a foundation on which specific architectures and architectural building blocks can be built. This includes a database of open industry standards that can be used to define the particular services and other components of an organization-specific architecture

TOGAF encourages but does not mandate the use of IEEE-1471. Organizations that have incorporated, or plan to incorporate, IEEE-1471 into their IT architecture practice should find that although some of the terminology used is not completely consistent with it, none of the key concepts in TOGAF is incompatible with this standard,.

An example used in TOGAF is the use of the terms "Business Architecture", "Technical Architecture", etc. These terms reflect common usage, but are at variance with IEEE-1471 (in which "architecture" is a property of a thing, not a thing in its own right). The plan is a gradually convergence between TOGAF and relevant standards for architecture description as IEEE-1471 gains increased acceptance within the industry. [TOGAF]

2.3.1.3 'C4ISR-AF'

The Command, Control, Communication, Computer, Information, Surveillance, Reconnaissance – Architectural Framework (C4ISR-AF) is an architectural framework used in the defence sector. C4ISR-AF focuses primarily on interoperability, targeting both operational, system and technical aspects of command, control, communication and information systems (i.e. C4I systems). In C4ISR-AF, architectural descriptions are divided into three major perspectives (views) that logically combine to describe the architecture. These three views are the operational, systems and technical views.
Like with RM-ODP, C4ISR-AF also can be seen as additional normative restrictions on IEEE-1471 architectural descriptions. However, even if the architecture products in C4ISR-AF are targeted towards describing C4I systems with many C4I-specific concerns, many are generally applicable. For further information about C4ISR-AF we refer to [C4ISR-AF].

2.4 CORAS – 'A platform for risk analysis of security critical systems'

CORAS is a research and technological development project under Information Society Technologies (IST) Program. One of the main objectives of CORAS is "to develop a practical framework for a precise, unambiguous and efficient risk analysis, by exploiting the synthesis

of risk analysis methods with semiformal specification methods (in particular, methods for object oriented modelling) and computerized tools, in order to improve the risk analysis of security critical systems" [nr.no/coras].

CORAS is characterized by:

- Techniques from risk analysis methods
- Patterns and methodology for UML oriented modelling
- A risk management process based on [AS/NZS4360] and [ISO17799]
- A risk documentation framework based on RM-ODP [ISO10746]
- An integrated risk management and system development process
- A platform for tool-inclusion based on XML

We believe CORAS is a well defined and thoroughly tested risk analysis methodology, and will use it as a basis for the risk analysis concepts in AMBRA.

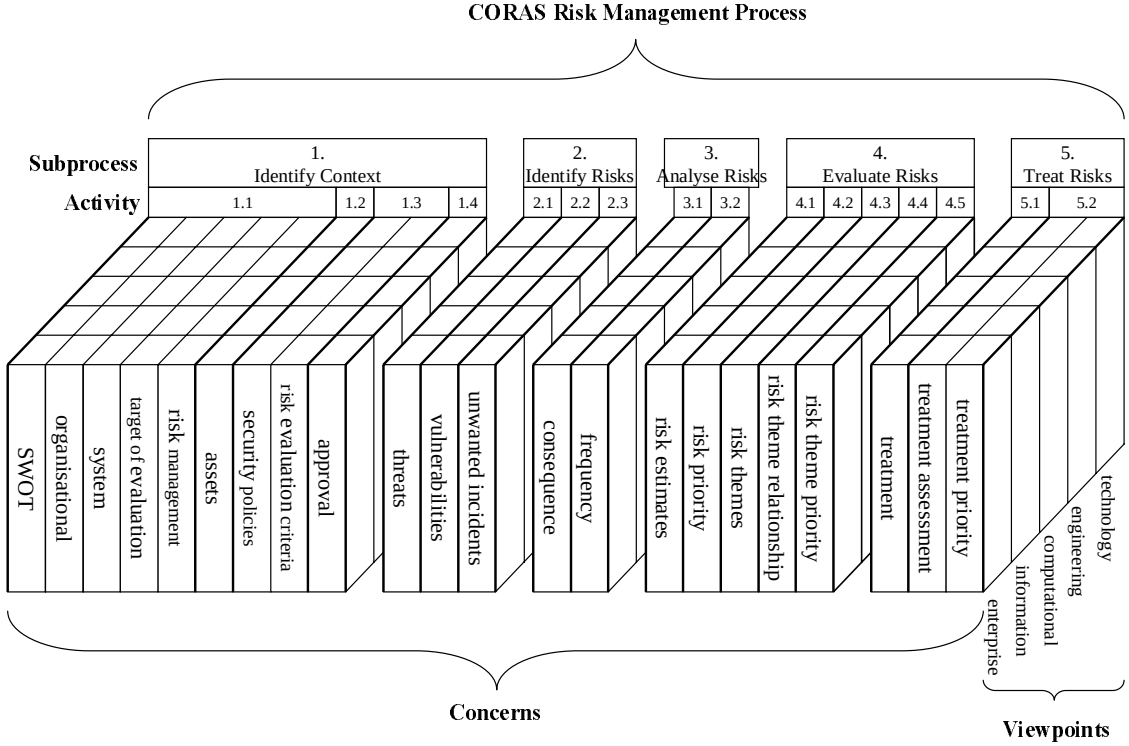

Figure 2-4 The CORAS Risk Documentation Framework

In order to abstract the essence of CORAS we find it sensible to look at the documentation framework as a basis for this. The framework illustrates important terms and relationships in a risk analysis methodology, but does not include details about which techniques that should be used. Figure 2-4 from [CORAS] illustrates the documentation framework in CORAS, which uses terms like "concern", "viewpoint" and "models" all recognized from IEEE-1471.

There exists several publications and reports concerning CORAS, in this thesis we will mainly use [Braber et.al. 03] and[Lund et.al. 03] as reference.

The *relationships* between central terms in CORAS risk documentation framework (Figure 2-4) is displayed in Figure 2-5 [Lund et.al. 03]. It shows the dependencies between analysis results, and illustrates what information is important to capture during a risk analysis.

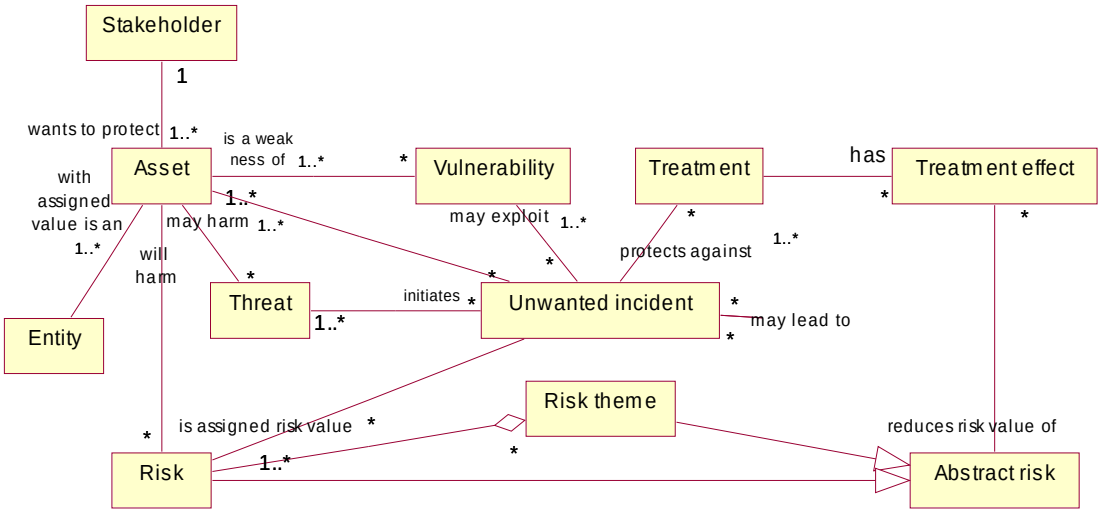

Figure 2-5 A conceptual model of CORAS' risk documentation framework

The definitions of the terms in Figure 2-5 are taken from [Lund et.al. 03] and are as follows:

- *Stakeholder:* A person or organization who has interest in the assessed system.
- *Entity:* An entity becomes an asset when assigned a value by a stakeholder.
- *Asset:* A part or feature of the system that has value for one of the stakeholders.
- *Threat:* A potential cause of an unwanted incident, which may result in harm to a system or organization and its assets.
- *Vulnerability:* A weakness with respect to an asset or group of assets which can be exploited by one or more threats.
- *Risk:* An unwanted incident that has been assigned a consequence value, a frequency value and a resulting risk value.
- *Abstract risk:* The common properties of risks and risk themes, such as risk value.
- *Risk theme:* A categorization of similar risks, assigned its own risk value.
- *Treatment effect:* A treatment's capability to reduce the risk value of a particular risk.
- *Treatment:* A way of reducing the risk value of a risk or risk theme.
- *Unwanted incident:* An undesired event that may reduce the value of an asset

To summarize:
Sections 2.2 and 2.3 have shortly introduced you to IEEE-1471 and CORAS, in Figure 2-6 we have illustrated how AMBRA relates to them.

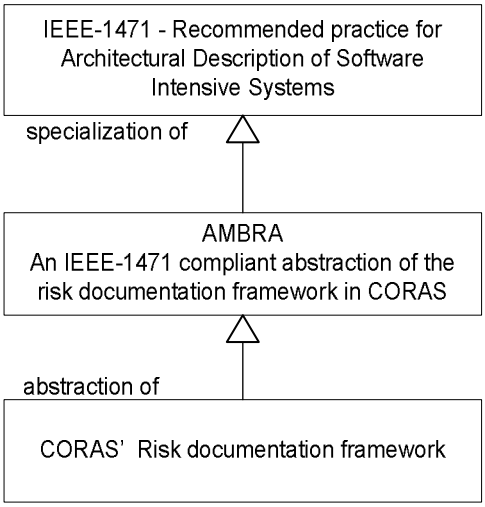

Figure 2-6 Relationships IEEE-1471, AMBRA and CORAS

2.5 MDA - 'Model-Driven Architecture'

The standardization organization OMG (Object Management Group) is currently working on developing a model-driven[3] framework for the software development process. The need for an architecture using object technology, meant for distributed application integration with focus on reusability of components, interoperability and portability, lead to development of the Model Driven Architecture™ (MDA™)
MDA is referred to as a new way of writing specifications and developing applications, with foundation in the IEEE-1471. It is claimed to improve quality and efficiency by facilitating precise communication and interaction between stakeholders and also reduce maintenance costs by providing possibility for reuse. The aim is to raise the level of abstraction so that models, rather than implementation code, become the main units of reuse.

MDA development is model-based and uses models to specify, visualize and document the system before the actual coding. This type of development focuses first on the functionality and behaviour of distributed applications or systems, independent of the technologies in which it will be implemented. MDA separates implementation details from business functions and OMG therefore claims there is no need for repeating the process of modelling an application or system's functionality and behaviour each time a new technology comes along [omg.org/mda].

Someone think the documentation on MDA from OMG does not explain exactly *what* MDA is, and the authors of [Runde,Stølen 03] make an effort to answer this question. They view MDA as "*a framework for defining system architecture development methodologies by giving directions for making architectures, serving as a tool-box for system developers and system architects, and facilitating integration and interoperability between systems.*" [Runde,Stølen 03] specify in their article the three parts of the framework in a terminology, a library of reusable elements and a methodology. The main focus of the article is the terminology part.

[3] In order to avoid confusion we will not make any distinction between the terms *model-driven* and *model-based*

Figure 2-7 illustrates the MDA framework, marking the parts we reckon as too specific and detailed for this thesis with dashed lines.

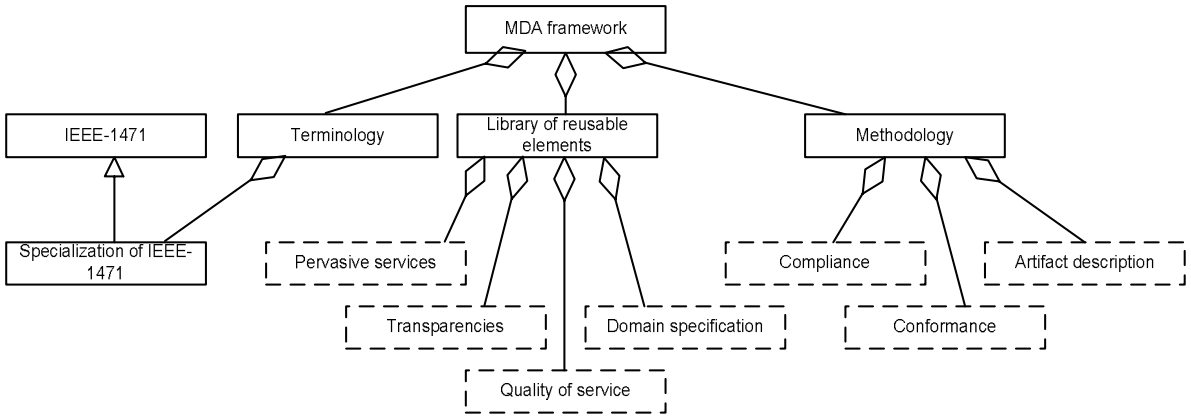

Figure 2-7 The MDA framework

The *library of reusable elements* in MDA should according to OMG include models for use across different applications, and the *methodology* should include a concrete methodology of MDA with aspects like compliance checking, conformance checking and instantiation of library elements. [Runde,Stølen 03] discuss these parts only superficial, because they find the specification made of OMG somewhat weak and unsatisfactory described.

Figure 2-8 taken from the article[4], introduces MDA in the hierarchy of architectural levels. Each level in the Figure is a possible instance of the level above, and an analogy to the process of building a house is used:

Architecture hierarchy	System development example	Building analogy
MDA framework (for defining architectures for system architecture development)	(MDA framework)	Building laws and regulations
Methodology (architecture) for developing system architecture	KobrA [Atkinson et.al. 02]	Rules and conventions for developing the architecture of homes
System architectural description	The Library Systems Product Line [Bayer et.al. 01]	Architectural description of new home
Documented system	An implementation of the Library System, together with documentation	New home with documentation

Figure 2-8 The hierarchy of architectural levels

MDA is meant to be a framework for defining methods for system development. In almost the same manner there exist superior building laws and regulations when a contractor plans a house. According to these superior laws and regulations the contractor develops a process for building a special type of house, e.g. a sky scraper. As a step in this process he creates an architectural description of the sky scraper. The next step in the process is building the sky scraper according to the description he made in the previous step.

[4] The Figure is recreated with basis in an equivalent Figure in [KRS03].

2.6 UML - 'Unified Modelling Language'

The unified modelling language (UML) is defined as a standard by OMG (Object Management Group). UML enjoys a widespread industry support because it is recognized as methodology-independent. Regardless of the methodology used to perform analysis or design, it is claimed that it is possible to use UML to express the results. This is a common language that can be used from product conception to delivery, from system to detailed design levels. UML is suitable for creating models of a system which is helpful in specifying, visualizing and documenting a system. UML is also one of the foundations in OMG's work with model-driven architecture (MDA).

UML defines twelve types of diagrams divided into three categories:

- Class Diagram, Object Diagram, Component Diagram, and Deployment Diagram represent static application structure.

- Use Case Diagrams, Sequence Diagrams, Activity Diagrams, Collaboration Diagrams, and Statechart Diagrams represent different aspects of the systems' dynamic behaviour.

- Model Management Diagrams include Packages, Subsystems, and Models represent ways one can organize and manage application modules

In this thesis we will mainly use 'the class diagram' which lets us model the concepts in AMBRA and the relationships between them. We expect the reader to be familiar with UML notations like cardinality, aggregation, specialization and roles. The version we will use is UML 1.4. For more information about UML we refer to [UML].

2.7 An introduction to the 'architectural hierarchy'

As already mentioned 'architecture' is not only used for describing the structure of a system or a building, but also the how the system was developed, i.e. the methodology. In the case with AMBRA we are describing an architectural framework for a methodology, not a system. In the same manner as they do in [Runde,Stølen 03] we have tried to illustrate the hierarchy of architectures in relation to our focus on risk analysis in Figure 2-9. As an example analogy we have used the process of medical treatment which in many cases resembles a risk analysis process. The Figure is divided into different levels where the lower level is a possible instance of the level above. At the top level you find basic medical conventions relevant for all types of medical treatment. Based on this scientists develop a method for examination in order to discover an injury or disease. This also leads to a description of how to apply the treatment to the patient. If the treatment was applied successfully the result is a healthy person.

In the same manner we have described an example from risk analysis. On the top level we have a framework for defining risk analysis methodologies which AMBRA is an example of. A possible instance of AMBRA is the CORAS risk analysis methodology. Applying this methodology may result in a recommendation of a security architecture like a firewall solution. The result of implementing this architecture will probably be a more secure system.

Architecture hierarchy	Risk analysis example	Medical treatment analogy
AMBRA - Architectural framework for model-based risk analysis methodologies	-	Medical knowledge 
Methodology for risk analysis	CORAS Risk Analysis methodology	Research for establishing a protection against head injuries 
Security architecture	Security architecture for server configuration based on the use of a firewall	Recommended protection against head injuries 
Documented system based on security architecture	An implemented firewall solution	Person wearing a helmet for protecting the head 

Figure 2-9 Architecture levels in relation with risk analysis

This thesis will address high level architecture in the same level as MDA and RM-ODP, but unlike MDA and RM-ODP which focus on system development methodologies, we will focus on risk analysis methodologies.

3 Problem Analysis

In Section 1.5 we motivated the development of AMBRA. The overall argument was the need for risk analysis in order to achieve secure, safe, robust and cost-effective systems. In risk analysis much is common between different domains. This motivates reuse of e.g. terminology, results and artefacts from successful risk analysis. Another aspect we emphasize is the use of models in risk analysis, in our case often represented as architectural descriptions of the target system. AMBRA will base itself upon a well defined model-based risk analysis methodology (CORAS), be specified in a well known modelling language (UML) and follow international accepted standards for architectural descriptions (IEEE-1471). We hope this will lead to a better understanding and integration of risk analysis in system development.

Our goal for AMBRA is for it to function as a reference framework for development of model-based risk analysis methodologies. AMBRA will address different groups of users, some will use it directly, but a large group will be indirect users, users of or affected by methodologies developed according to AMBRA. Direct users will typically be methodologists and indirect users will be people like risk analysts, target users and quality assurance people.
In the following we motivate and explain this goal in further detail. In particular we formulate one superior success criterion and five concrete success criteria characterizing what it means to reach this superior criterion. The order of which we present the success criteria is of no importance, the essence of them is summarized in the overall success criterion: *AMBRA should be a supporting architectural framework for model-based risk analysis.*
Afterwards we will evaluate AMBRA against each of the five success criteria. If we find they have been reached successfully they support our superior success criterion.

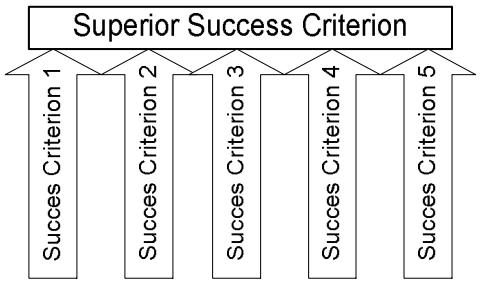

Figure 3-1 The structure of the success criteria for AMBRA

In research we often hear about *research hypotheses*, statements expressing a solution for or conclusion about the topic the researchers explore. Normally one uses two hypotheses, a null-hypothesis (H_0) and an alternative hypothesis (H_1). To explain this we use an example where researchers want to check if UML is suitable for a specific modelling purpose. The research team believes it will be perfect for this purpose and formulates the alternative hypothesis H_1: "UML is well suited for this modelling purpose". Since they can not be sure of this they need to formulate H_0: "UML did not make any improvement to the modelling of this area". Then the researchers conduct some experiments and based on this they conclude with one of the hypotheses.
Our superior success criterion is our H_1 and the five success criteria may be understood as sub-hypotheses that together are meant to "imply" the overall hypothesis. In the style of Popper [Popper], falsification of one of the success criteria will falsify the overall hypothesis.

3.1 The refined problem characterization

In the following we present the superior success criterion for AMBRA, and then we deduce a set of more specific requirements (success criteria), aiming to characterize what it means to fulfil our overall objective.

3.1.1 The superior success criterion

AMBRA will exploit the fact that there is much common in terminology between different risk analysis domains. A risk analysis methodology developed for one type of system will most likely use much of the same terminology as one developed for a different system.
By gathering this common terminology into a general framework like AMBRA, it will ensure that a methodology developed according to this framework will include well-known and thoroughly understandable terms. We certainly believe there is a need for specialized risk analysis methodologies for particular domains. But instead of starting from "scratch" each time one can apply AMBRA as a *general* framework, serving as a foundation for the terminology in a model-based risk analysis methodology. This will provide the developer with a basic terminology and structure, but with the possibility to add more specific terms in order to suit the specific domain.

AMBRA will adopt the same organization as [Runde,Stølen 03] with: terminology, methodology (called AMBRA guideline) and library of reusable elements (displayed in Figure 3-2) in order to comply with the same type of MDA framework. We believe this tripartition will make the framework easier to understand and use, as each part is concerned about different aspects. Also this partition makes it possible for us to finish only one of the parts, the terminology, leaving the other two, library and guideline, as subjects for further work.

The terminology part of AMBRA will, as specified, include aspects from the CORAS risk analysis methodology and concepts from architectural descriptions according to IEEE-1471. A conceptual model of the terminology and risk management process in AMBRA will be specified in UML.

The library of reusable elements will not be adressed in this thesis, only a description will be discussed in Chapter 7. We have chosen to rename the methodology part in MDA to 'AMBRA guideline' to avoid confusion of terms. This guideline will include guidance for how to use AMBRA, how to conduct conformance and compliance checking etc., but like the library part, this will be subject of further work.

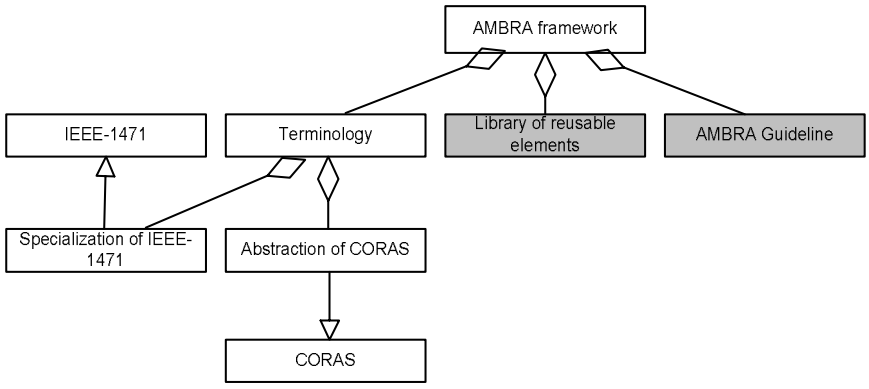

Figure 3-2 The AMBRA framework

Based on this we specify the superior success criterion in detail as:

The superior success criterion:

AMBRA is a supporting architectural framework for model-based risk analysis

a) AMBRA facilitates reuse of best experiences from risk analysis.
b) AMBRA provides a common basis for model-based risk analysis within different domains.
c) AMBRA is a reference framework for the development of risk analysis methodology.

3.1.2 The 1st success criterion

In a risk analysis situation models are often used to describe the target system. We claim that if one uses well designed and precise models with appropriate level of details, it will be easier for the participants in risk analysis to contribute since they understand the system better. Using incomplete or complicated models may lead to misunderstandings and mistakes. This applies in particular in situations involving IT-systems because of their abstract nature which may be hard to overview. In order to ensure good comprehension in models we find it wise to employ the conventions for architectural descriptions found in a recognized standard.

The decision for incorporating a well known standard is in general that the use of standards simplifies the collaboration between different actors. Although we make use of a software standard, its notion of system is very wide and its terminology carries over to most domains. It also may reduce the cost related to integration of new components and personnel training. For small actors and enterprises there are specific reasons for using international accepted standards. By basing products on accepted standards their chances of gaining market share and achieve success are increased. Small companies are generally not in the position to successfully base themselves exclusively on their own developed standards, in contrast to powerful and multinational companies like Microsoft Inc. Compared to the rest of the world, Norwegian actors are small and benefit from using international standards in order to have a chance to succeed with their products on the international market.

There exists more than one potential standard to rely on, and the one you choose to use depends on the system and what type of context it will be used in. A standard like C4ISR-AF (Section 2.3) is especially developed for use in the military domain and their C4ISR systems. Other standards for system description are IEEE-1471, RM-ODP and TOGAF (Section 2.3). In this thesis we chose to follow IEEE-1471 ‘Recommended Practice for Architectural Description of Software-intensive Systems [IEEE1471].
This decision is based on our understanding of IEEE-1471 as a more high-level standard, or framework, posing less strict rules for use and conformance on the user than the other three which is more like complete frameworks. C4ISR-AF, RM-ODP and TOGAF could all be seen as a further refinement of IEEE-1471 adding more details and restrictions. IEEE-1471 is not a replacement for C4ISR-AF, RM-ODP or TOGAF; it is an organizing framework intended to supplement such standards, by providing specific content requirements on architectural descriptions.
Since IEEE-1471 is very general leaving more up to the user the other three, we recognize it as more suitable to use in the type of superior framework AMBRA is meant to be. Another argument for using IEEE-1471 is the general agreement that having a good architecture is critical to system success, and IEEE-1471 presents consensus elements of good practice in describing architectures.

IEEE-1471 standardizes a basic framework for the content of an architectural description with focus on stakeholders, in particular giving specific meaning to what are called *views*, *stakeholders* and *concerns*. Different stakeholders may have different views of the system, and IEEE-1471 emphasises the importance of using models of the system according to what the stakeholder is interested in, i.e. what the stakeholder is concerned about. This standard is also the foundation for OMG's [OMG] work with model-driven architecture (MDA) (Section 2.5). Since AMBRA will be model-based we believe it is sensible to incorporate some of the same underlying ideas as MDA.
Based on this we specify the first success criterion in detail:

The 1st success criterion:

AMBRA should be IEEE-1471 compliant

a) AMBRA extends IEEE-1471's conceptual framework of an architectural description in Figure 2-3 to include reusable aspects from risk analysis terminology.
b) AMBRA may take the liberty to rename aspects of IEEE-1471 if this increases the comprehension of the framework.
c) AMBRA may take the liberty to rename aspects of IEEE-1471 if this increases the focus on risk analysis.

3.1.3 The 2nd success criterion

An important aspect of AMBRA is to combine a well defined terminology for risk analysis with concepts from architectural descriptions. In order to ensure that we included all relevant terms from model-based risk analysis we found it useful to consult an existing model-based risk analysis methodology. The methodology we chose to use is CORAS 'A platform for risk analysis of security critical systems' (Section 2.4). During the development of the CORAS methodology for model-based risk analysis, several existing methods for risk analysis were evaluated in order to extract the best parts into a foundation for the CORAS' methodology. CORAS defines a complete approach to risk management and includes details about every aspect concerning this, but we will only adopt the high level aspects of CORAS. AMBRA will adopt the risk analysis terminology from CORAS and its way of organizing the risk management process.

We have previously stated that AMBRA should extend the conceptual specification of an architectural description in IEEE-1471 with terminology from risk analysis. This extension we aim to achieve by embedding the CORAS conceptual specification into the IEEE-1471 conceptual specification. There are several similarities between the terminology in CORAS and IEEE-1471. In cases where the two frameworks disagree about the interpretation or name of a term, we will focus on the interpretation found in IEEE-1471. This decision is motivated by the initial requirement to make AMBRA a specialization of IEEE-1471 and therefore include all terms from IEEE-1471.
We aim to comprise the most important risk analysis concepts by focusing on the CORAS documentation framework. These are terms and relationships CORAS recognizes as important to document during risk analysis and therefore also important in AMBRA. CORAS emphasizes storing successful risk analysis results in a structured way to facilitate later reuse and efficient utilization. AMBRA focuses on providing a well-founded terminology to the people who develop risk analysis methodologies, not deciding the exact techniques (HazOp [HazOp], FMEA/FMECA [FMEA], Markov analysis [Markov], FTA [FTA] or CRAMM [CRAMM]) to use.

Based on this we specify the second success criterion in detail as:

The 2nd success criterion:

AMBRA should be an abstraction of the CORAS methodology

a) The risk analysis terms extends the conceptual specification of IEEE-1471 in a natural manner.
b) The risk analysis terms should be extracted from the conceptual specification of the CORAS risk documentation framework in Figure 2-4.
c) The risk analysis terms included in AMBRA maintains their original interrelationships from Figure 2-4.
d) AMBRA adopts the conceptual specification of the risk management process in CORAS

3.1.4 The 3rd success criterion

How useful AMBRA is recognized to be depends on the users' opinion and whether they find it useful from a practical point of view. AMBRA may have more than one area of use and therefore different types of users; so far we have identified two different groups of intended users.

The first group of intended users we call '*direct users*'. They will mainly be people who develop model-based risk analysis methodologies and tools for use in risk analysis. This group will use AMBRA as a vehicle in the development process, a useful reference framework to work within ensuring that the new method incorporates important concepts and their relationships, and is interoperable with already existing methods. These are the ones who directly use AMBRA, but also two types of *indirect users* exist. '*Indirect users I'* are mostly people who use methodologies developed according to AMBRA and '*Indirect users II*' are those affected by the results from using of these methodologies.

Members of 'Indirect users I' are typically risk analysts and target stakeholders providing target input and documentation:
For the *risk analyst* it is important to be sure of the validity of the concepts in the risk analysis methodology. If the method is imperfect or insufficient, the risk analysis may lead to incorrect or insufficient results. The risk analyst will also find the methodology useful if results from earlier use of the method are stored in a way that facilitates reuse. By consulting results from similar risk analysis the analysts may use them to draw conclusions about the target system more easily.
The *target stakeholders* are familiar with the system and provide useful target input, i.e. documentation of target system, for the risk analysis. This may be models of the whole or parts of the system like interaction diagrams displaying communication between different parts of the system. The target stakeholders need to deal with AMBRA's conventions for architectural descriptions. The degree of usefulness AMBRA will have for this group depends on how intuitive the terminology is, i.e. to which degree it matches the interpretation in natural language. Also the target stakeholders will benefit from storing risk analysis results in a consistent manner. Often it is possible to reuse parts or whole models from earlier risk analysis on similar or identical systems, saving both time and effort.

The reason for dividing the 'indirect users' in two is the distinction between the people who use a methodology and the people who is affected by the actual use of the methodology. 'Indirect users II' includes those who certify organizations and people responsible for the quality assurance (QA) in the company.

The former investigates whether the company is using their standards properly, e.g. in conformance with their recommendations. He or she typical conducts tests on the system, inspects system documentation and routines in the organisation to see if they conform to the standard.
The QA people in the company will find AMBRA useful if it turns out that using a methodology developed according to AMBRA contributes to improve productivity and better quality.

Even if we divide the users of AMBRA into direct and indirect users, this is not meant to be a strict separation of user types; members of one group might also participate in the other group, and vice versa. Figure 3-3 illustrates the different groups of intended users of AMBRA.

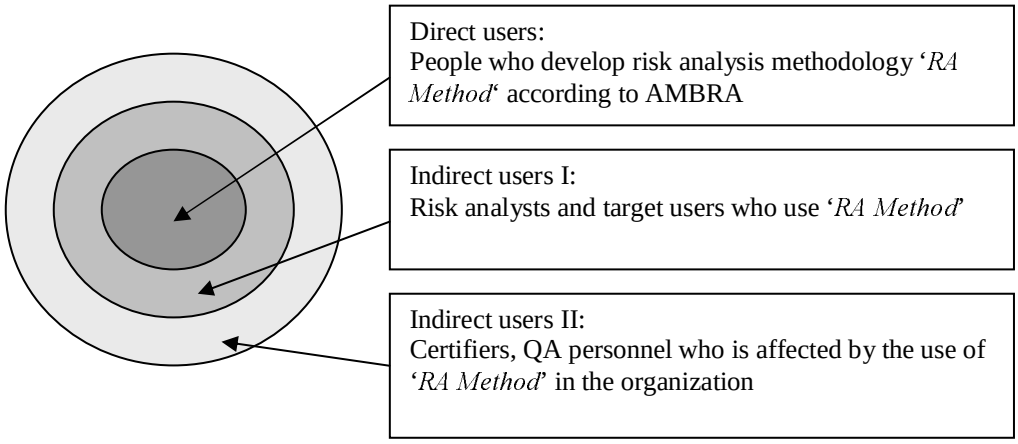

Figure 3-3 Intended users of AMBRA

Based on this we specify the third success criterion in detail as:

The 3rd success criterion:

AMBRA should be useful from a practical point of view

a) Direct users find AMBRA useful as a reference framework for development of model-based risk analysis methodologies.
b) Indirect users I like target users and risk analysts, find the terminology from AMBRA intuitive.
c) Indirect users II, like QA personnel and certifiers, approve AMBRA as a good basis for the risk analysis methodologies used in the organization.

3.1.5 The 4th success criterion

There are clear similarities between the MDA framework in [Runde,Stølen 03] and the framework AMBRA is intended to be. They are both examples of 'top-level' architecture, a sort of 'framework for developing methodologies'. AMBRA will adopt the same structure as [Runde,Stølen 03], in order to comply with it as an extension with focus on risk analysis. The structure consists of a terminology part, a library of reusable elements and a methodology part (displayed in full version in Figure 2-7). By organizing AMBRA in clearly separated parts where each Section focus on different aspects like in [Runde,Stølen 03], we also believe we make it easier for the user to get an overview of our framework.

MDA is meant to be a basis for creating model-based system development methodologies. MDA focuses on the use of models and is by many expected to become the dominant idea in

system development, and we think it is important to situate AMBRA in the context of MDA. This will facilitate a tight integration of risk analysis in the system development process.

The part about reusable elements in MDA should according to OMG include a library of predefined models for use across different applications. This thesis will not go into the details of this part, only discuss it briefly as a subject for thorough research in future work.

The third part of the MDA, the methodology part, should include a concrete methodology of MDA. In the sequel we choose to call our corresponding part for 'AMBRA Guideline', as the term 'methodology' is often used in this thesis and could lead to confusion. The AMBRA Guideline should be a user guide, including aspects like compliance checking, conformance checking and instantiation of library elements. This thesis will discuss this part only superficially, because like the authors of [Runde,Stølen 03] point out, the specification of the methodology part by OMG is weak and not yet satisfactory as a complete part of the framework. According to this only the terminology part is subject for completion in this thesis. The content of the remaining two parts, library and guideline, will be discussed in Chapter 7 'Discussion' but the actual development is left as subjects for further work with AMBRA.

Based on this we specify the fourth success criterion in detail as:

The 4th success criterion:

AMBRA should comply with MDA

a) The terminology of AMBRA complies with the terminology of MDA.
b) AMBRA allows for an AMBRA guideline and a library of reusable elements compliant with the organization of MDA

3.1.6 The 5th success criterion

A Norwegian writer would probably benefit from publishing his work in a 'world-language' like English in order to reach a larger audience than only the Norwegian-speaking. In the same manner we think AMBRA should be expressed in a well-known and widely accepted modelling language. AMBRA will in great extent present conceptual models of the terminology in risk analysis and architectural description.
We choose to use UML (see Section 2.6) a widely used modelling language within the system engineering industry. UML is by many people recognized as *the* modelling language and AMBRA will most likely benefit from being expressed in this world-wide accepted notation. UML is also used in the official specifications of both CORAS, IEEE-1471 and MDA.

When developing a methodology according to AMBRA, we will not impose on the new methodology to use UML, but we would not hesitate to make such a recommendation as we believe it is a well suited modelling language for this purpose. The question of which modelling language to use have to be addressed in each individual case with respect to the domain the methodology should be used in.

Based on this we specify the fifth success criterion in detail as:

The 5th success criterion:

AMBRA should be documented in UML

a) The conceptual models in AMBRA are specified in UML

3.2 Validating the success criteria

There are several strategies for validating research hypotheses, each one suitable for different research settings. Joseph E. McGrath has presented what he reckons as a complete set of strategies in his book "*Groups: Interaction and performance*" [McGrath 84]. According to him you are faced with the following dilemma when choosing the appropriate strategy: "*all methods have inherent flaws – though each has certain advantages. These flaws cannot be avoided. But what the researcher can do is to bring more than one approach, more than one method, to bear on each aspect of a problem*".

In a research study you want to maximize the generalizability of the evidence over populations of actors (A), meaning you look for results which are possible to apply on a broad population. You also want to get the highest precision of measurement of the behaviours (B) and realism of the situation or context (C). Unfortunately, increasing one of these three factors decreases the other two. It is impossible to maximize all at the same time, and we need to decide which one is the most important, in order to choose the right type of research strategy. The Figure below from [McGrath 84] shows the different strategies and their relationships to the three factors mentioned (A, B, C).

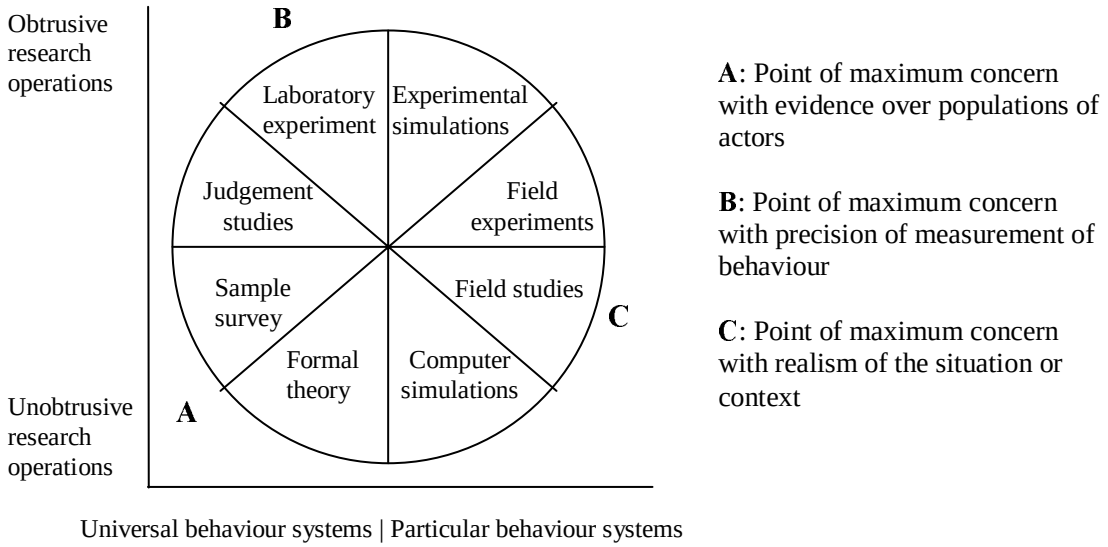

Figure 3-4 Research strategies

When deciding which strategies to use we will discuss each one of the different types displayed in Figure 3-4. The definitions of the strategies we use here are all from [McGrath 84].

A *field study* is to observe a system in natural settings without disturbing it. As claimed earlier we believe there are mainly two types of users of AMBRA: *direct users* and *indirect users*. A field study including *direct users* could be to *develop* a methodology according to AMBRA. This is an extensive study, but could help discover elements which should be included, or excluded, from AMBRA due to e.g. ambiguities.
A field study with *indirect users* could be a situation where a methodology developed according to AMBRA is *tested*. This could help validating how intuitive the terminology defined in AMBRA is, i.e. does the interpretation of terms and definitions match natural language? If there is possible to misunderstand terms or relationships, it could lead to faulty results and should therefore be thoroughly checked.

The field study strategy could be interesting if we had access to a larger group of people to experiment on, but unfortunately we only have access to a limited number of test personnel.

Due to the limited number of test personnel, we also find it difficult to conduct a *sample survey*. A sample survey is 'an effort to get information from a broad (and well devised) sample of actors, usually in the form of verbal responses to a relatively small set of questions'. This is typically done in situations where a manufacturer handles out samples of his product, asking people to respond to some questions about e.g. flavour, appearance and smell. Since this strategy requires a broad sample of actors, we need to postpone a sample survey about AMBRA until it hopefully gets a larger user-base.

A *field experiment* is a field study where one 'deliberate manipulates a feature in order to study its effects'. An interesting experiment could be to see if a change in a definition of a term or a relationship in AMBRA would make a difference for the group creating or using risk analysis methodologies.
In this thesis there will be conducted neither a field study nor a field experiment due to time and resource limitations. These are both left as subjects for further work even if they undoubtedly could lead to interesting results.

Computer simulations are 'attempts to model a specific real life system or class of systems'. This topic will not be investigated further in this thesis as we neither have enough available time nor the required resources. Furthermore, it is hard to see any potential for such an approach in our case.

Laboratory experiments are 'attempts to create the "essence" of some general class of systems in a context in which the researcher can control all (or at least very many) of the extraneous features of the situation, in order to be able to maximize the essential features with precision'. Conducting a laboratory experiment with AMBRA would be difficult as there is little or no chance of obtaining control over all features in a situation where AMBRA is used.

Experiment simulations are 'laboratory studies in which an effort is made to create a system that is like some class of naturally occurring systems, but which artificial in that they are created by the researchers for study, and people perform in them for research purposes rather than for purposes stemming from their own lives'. Experiment simulations could be used in a structured brainstorming like HazOp in order to monitor all participants and their understanding of the terminology, but this is too detailed for AMBRA. The strategy requires also use of laboratory, something we do not have access to, and therefore considered as inappropriate for our use.

As a result of the previous discussion all of the five research strategies: field study, sample surveys, field experiment, computer simulations, laboratory experiments and experiment simulations are considered as out of the scope of this thesis and suggested as possible tasks for future work with AMBRA. In our opinion one of the most interesting of these suggestions for further work is to conduct a field study of AMBRA where one actually could test if AMBRA works for its purpose. For a more detailed discussion about possible subjects for further work we refer to Section 8.2.

The use of *Formal theory* normally implies argumentation based on mathematical representations of the actual problem, predicate logic or similar formal methods. In this thesis we think of formal theory in a more general way. We think it should be sound and theoretical

argumentation, expressed in natural language, for AMBRA's conformance with formal specifications of IEEE-1471, CORAS and MDA. According to this interpretation, we believe the best way to verify **Success criterion 1,2,4,5** (IEEE-1471, CORAS, MDA and UML) is *formal theory.* They all deal with verifying AMBRA's conformance with different formal specifications and standards. We believe either a criterion of success is fulfilled or not, i.e. the notation used for expressing AMBRA either conforms to UML or not. In the same manner we will check the remaining three criteria by verifying their conformance to the specifications of IEEE-1471, CORAS and MDA respectively.

In order to validate the usefulness of AMBRA (**Success criterion 3)** we think *judgment studies* are most suitable. A judgment study is close to a sample survey, but requires a smaller group of selected sample of "judges". The rather small group of persons with knowledge about CORAS, IEEE-1471 and MDA we plan to use as judges, will hopefully provide us with more extensive information than we would get in a more superficial sample survey. We believe we make better use of the judge's competence by including them in a judgment study where they need to understand the thesis more carefully than else would be required in e.g. a sample survey.
Judgment studies are described in [McGrath 84] as '*efforts to get responses (usually from a very small and somewhat casually selected sample of 'judges') about a systematically patterned and precisely calibrated set of stimuli. […] Judgment studies have less generalizability over actors (A), but retain considerable precision of measurement (B). Both surveys and judgment studies try to deemphasize context - actually, to uncouple the behaviour (judgment) from the context in which it is done. Thus both are very low on realism of context (C)*'.

As stated earlier there is a goal to use different research strategies emphasising each of the three areas of concern (illustrated in Figure 3-4):

A) generalizability of the evidence over populations of actors
B) precision of measurement of behaviours (and precision of control over extraneous facets or variables that are not being studied)
C) realism of the situation or context (in relation to the context to which you want your evidence to refer)

Judgement studies are quite close to *laboratory experiment* where control over measurement and behaviour is at the maximum (B). *Formal theory* is optimal for maximum generalizability over populations (A). Both judgment studies and formal theory are low on realism of context (C). From a statistical viewpoint this contributes to undermine the results as one violates with the goal of maximizing all three factors. To support our results one should chose an additional research strategy which emphasizes maximum of realism of context. Such a strategy could be a *field study* or *field experiment*. As stated earlier this is recommended as subjects for further work with this framework and discussed in Section 8.2.

Based on this discussion we have decided to use judgment studies and formal theory in order to evaluate AMBRA against our criteria of success in this thesis. We do not consider computer simulations, laboratory experiments, experimental simulations, field experiments or field studies as appropriate at this time.

A summary of the verification plan:

The superior success criterion:

AMBRA is a supporting architectural framework for model-based risk analysis

a) AMBRA facilitates reuse of best experiences from risk analysis.
b) AMBRA provides a common basis for model-based risk analysis within different domains.
c) AMBRA is a reference framework for the development of risk analysis methodology.

This will be substantiated by successfully satisfying the five criteria of success for AMBRA.

The 1st success criterion:

AMBRA should be IEEE-1471 compliant

a) AMBRA extends IEEE-1471's conceptual framework of an architectural description in Figure 2-3 to include reusable aspects from risk analysis terminology.
b) AMBRA may take the liberty to rename aspects of IEEE-1471 if this increases the comprehension of the framework.
c) AMBRA may take the liberty to rename aspects of IEEE-1471 if this increases the focus on risk analysis.

This will be verified by use of formal theory. We will check AMBRA's conformance with the official specification of IEEE-1471.

The 2nd success criterion:

AMBRA should be an abstraction of the CORAS methodology

a) The risk analysis terms extends the conceptual specification of IEEE-1471 in a natural manner.
b) The risk analysis terms should be extracted from the conceptual specification of the CORAS risk documentation framework in Figure 2-4.
c) The risk analysis terms included in AMBRA maintains their original interrelationships from Figure 2-4.
d) AMBRA adopts the conceptual specification of the risk management process in CORAS

This will be verified by use of formal theory. We will check AMBRA's conformance with the official specifications of CORAS.

The 3rd success criterion:

AMBRA should be useful from a practical point of view

a) Direct users find AMBRA useful as a reference framework for development of model-based risk analysis methodologies.
b) Indirect users I like target users and risk analysts, find the terminology from AMBRA intuitive.
c) Indirect users II, like QA personnel and certifiers, approve AMBRA as a good basis for the risk analysis methodologies used in the organization.

This will be verified by use of judgment studies. We will get people who are familiar with risk analysis, CORAS risk documentation framework, IEEE-1471 or MDA to respond to a questionnaire about AMBRA. Details about the study will be discussed in the Chapter "Evaluating AMBRA against criteria of success".

The 4th success criterion:

AMBRA should comply with MDA

a) The terminology of AMBRA complies with the terminology of MDA.
b) AMBRA allows for an AMBRA guideline and a library of reusable elements compliant with the organization of MDA

This will be verified by use of formal theory. We will check AMBRA's conformance with [Runde,Stølen 03].

The 5th success criterion:

AMBRA should be documented in UML

a) The conceptual models in AMBRA are specified in UML

This will be verified by use of formal theory. We will check AMBRA's conformance with the official specification of UML

4 Relationships in Terms and Concepts between IEEE-1471 and CORAS

In this Chapter we will discuss the similarities in terms and concepts between IEEE-1471 and CORAS. The purpose is to develop a terminology for AMBRA with foundation in both CORAS and IEEE-1471. In the following we assess the main terms shared between the two frameworks, presenting each of the interpretations and concluding with the interpretation we will use in AMBRA. The terms addressed are *stakeholder/role, concern, element, model, view/viewpoint,* and *intended users.*

4.1 Stakeholder and role

IEEE-1471: IEEE-1471 defines a system stakeholder as: *an individual, team or organization (or classes thereof) with interests in or concerns relative to, a system*.
According to the recommended practice a system has one or more stakeholders and a stakeholder may have different roles regarding the system; each role is treated separately as a new stakeholder in the architectural description.

CORAS: CORAS' definition of a stakeholder is taken from the Australian Standard for Risk Management [AS/NZS4360]. There a stakeholder is defined as '*those people and organizations that may affect, be affected by, or perceive themselves to be affected by a decision or activity in the risk management* process' *[AS/NZS4360]*.

If a stakeholder plays several different roles in the risk analysis process, the stakeholder is assigned different *role names* to separate the roles and belonging concerns in a clear manner. A risk analysis process may involve people with roles like project leader, risk analysis leader, risk analysis secretary, target owner, target developer, field expert and risk analysis expert etc. The risk management process consists of several activities and in the different activities of the risk analysis the participants may play different roles according to the purpose of the activity. This leads to the question whether a stakeholder has different types of roles in a risk analysis setting like CORAS.

Conclusion: In principle the two frameworks have the same interpretation of stakeholder, i.e. anyone in contact with the target of analysis. Both frameworks open for the possibility for a stakeholder to have multiple roles. A stakeholder could be both a person with responsibility of conducting risk analysis of a target (the system or part of the system) at the same time as the person is an operator of the target system. This makes it important to keep the two roles separated, not intervening concerns identified by one role with concerns identified by other roles. In an IEEE-1471 setting each role will be treated as a separate stakeholder, while in CORAS one operates with role name to separate the different roles of a stakeholder. In a risk analysis setting like AMBRA we identify two main types of roles, the roles of the risk analysis experts, and the roles of the target stakeholders assisting in the analysis. In order to emphasize the distinction between the role types we specialize the term stakeholder into two different types: *Risk Analysis Stakeholder* and *Target Stakeholder*. The *Risk Analysis Stakeholder* is related to the risk analysis process, interested in how to reuse models from previous, successful risk analysis and how to draw conclusions based on the results. The *Risk Analysis Stakeholder* may also be a customer or a decision maker interested in the results of the risk analysis of the system. The *Target Stakeholder* is normally familiar with the target system, often a regular operator of the system, responsible for e.g. providing target input to the risk analysis team and answer technical questions.

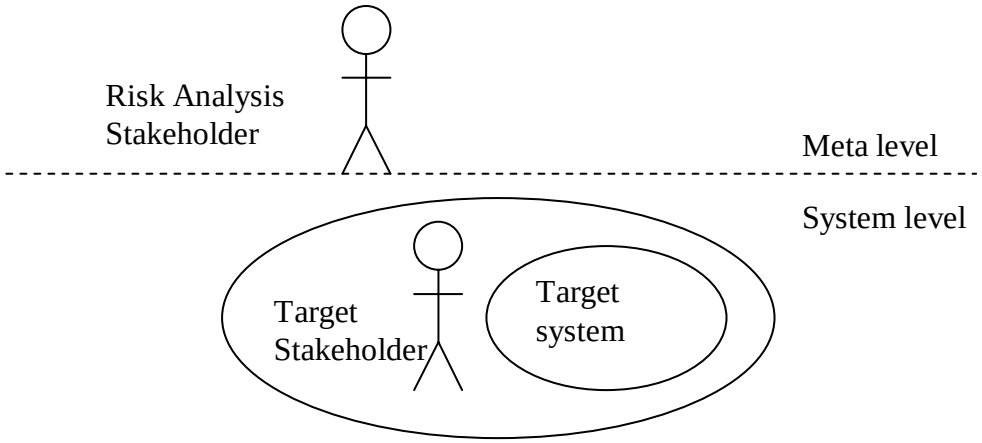

Figure 4-1 Stakeholders and roles

The definition of stakeholder in IEEE-1471 is somewhat broader than the one used in CORAS, but we believe this is due to CORAS' particular focus on risk analysis. We want AMBRA to be a specialization of IEEE-1471 extended with terminology from CORAS, therefore we will use the definition from IEEE-1471.

4.2 Concern

IEEE-1471: Concerns are '*those interests which relates to the system's development, its operation or any other aspect that are important to one or more stakeholders*'. As discussed earlier, a stakeholder may have more than one role related to the target and according to which role the stakeholder plays, the same stakeholder may have multiple and possible different concerns related to the same object. Concerns are discovered and formulated through interaction with stakeholders.

CORAS: The risk documentation framework in CORAS divides the five RM-ODP viewpoints into several concerns. Each concern belongs to a specified risk analysis activity and is a cross-viewpoint perspective linking together related information within different viewpoints (Figure 4-2). Each concern may consist of a set of specified elements like risk analysis tables or trees, models, logs and other documentation [Braber et.al. 03].

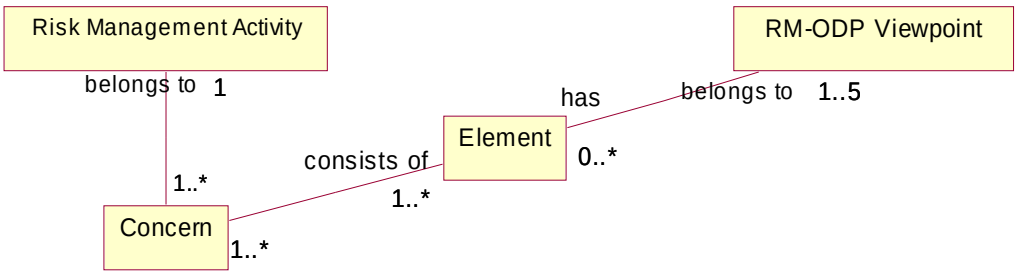

Figure 4-2 Concern in CORAS

Conclusion: The process of identifying concerns in IEEE-1471 resembles the activity "Identify and value assets" in the risk management process in CORAS [Braber et.al. 03]: for each stakeholder the assets of relevance for the target are identified and given a value. Therefore it is possible to assume a relationship between concerns in IEEE1471 and assets in CORAS. *Asset* is described in [Braber et.al. 03] as 'the carriers of the value of the system. This can be anything from knowledge to physical hardware, information etc.'. In IEEE-1471 a

stakeholder with concerns about a specific part of the system would probably recognize this as an asset.

Figure 4-3 Example: Asset and stakeholder in CORAS

The use of *Concern* in CORAS differs from IEEE-1471 where a concern is directly related to the specific stakeholders' system interests. CORAS uses it for linking related information from a risk analysis activity to the five different viewpoints as depicted in Figure 2-4. The *Concern* in CORAS is important to the risk analyst and addresses aspects on a sort of a meta-level of the target of analysis. If we recognize the fact that each stakeholder may have multiple roles and concerns in risk analysis, we need to specialize *Concern* into '*Concerns system-level*' and '*Concerns meta-level*'. They both represent concerns about the system, but different levels of abstraction. In our interpretation, CORAS-concerns are related to the 'meta-level' of the system in a risk analysis context and IEEE-concerns are related to the actual 'system-level'.
This is an extension of IEEE-1471 where a concern is related to the system stakeholders and only addresses elements on system level.

Based on our experience with the use of *Concern* in the two frameworks, we try to illustrate how they relate to each other in Figure 4-4. This interpretation will be used in the further work with AMBRA where the two types of concerns will be named *Risk Analysis Concern* (referring to *Concern* in CORAS) and *Target Concern* (referring to *Concern* in IEEE-1471).

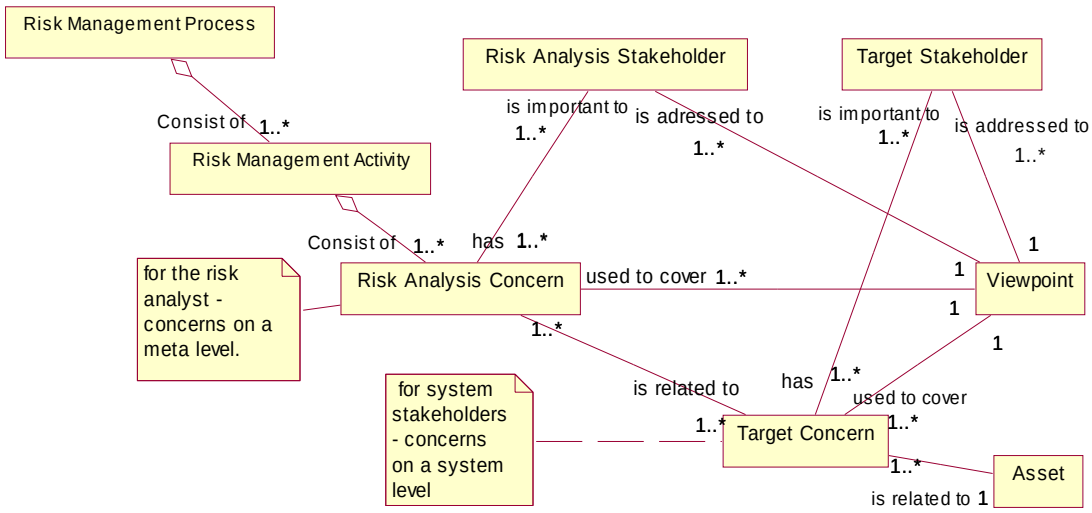

Figure 4-4 *Concern* in CORAS vs. *concern* in IEEE-1471

4.3 Element

IEEE-1471: This term is not used in IEEE-1471.

CORAS: A concern is decomposed into elements which contains information from the belonging risk analysis activity. Elements can be non-CORAS specific documentation, modelling elements, logs and risk analysis tables/trees [Braber et.al. 03].

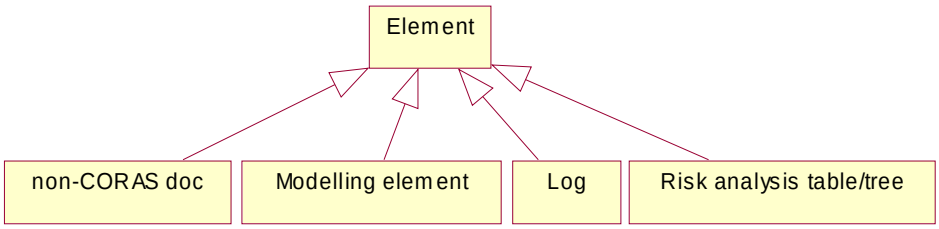

Figure 4-5 Types of elements

For regular users of CORAS, elements are only visible through integrated tools or through special platform interfaces.

Conclusion: It is reasonable to say that *elements* in CORAS correspond to *models* in IEEE-1471. According to IEEE-1471, models constitute a view, and a view expresses the system's architecture in some way. The Figure above shows what CORAS reckons as element: non-CORAS documentation, modelling element, logs and risk analysis tables/trees; each one of them is a model according to IEEE-1471's definition of models where an architectural model express the whole, or a part of, the system in order to address a concern. Since the definition of model in IEEE-1471 also covers the term *element*, we will use *model* to denote an element in AMBRA.

4.4 Model

IEEE-1471: Views consist of architectural models according to methods defined by its associated architectural viewpoint. The models are used to express the system architecture in order to answer one or more concerns.

CORAS: In CORAS a model refers to a specification of an entity in the real world, which has a commonly agreed semantics and a well-defined syntax. These are specifications that have a certain format (e.g. UML) and hold the same interpretation for all readers. CORAS uses models to describe system, to communicate with stakeholders and to document risk analysis results.

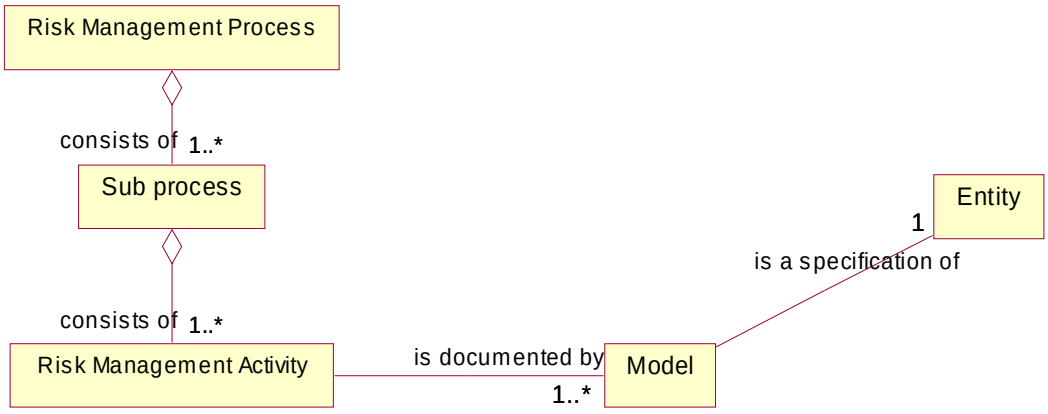

Figure 4-6 Relationships between concepts in CORAS

The models may be used to illustrate different levels of the system described, like enterprise level, business level, software system level, software component level and software object level. Viewpoint models may be used to communicate with stakeholders with another perspective on the system. Models are also useful in describing the levels of refinement of the

system from requirements through architectural and detailed design to realisation. A model belongs to one or more of the predefined viewpoints. [Lund et.al. 03]

Conclusion: The way the term *model* is interpreted in CORAS is not identical to the interpretation in IEEE-1471. In IEEE-1471 models are all figures and diagrams comprising a view on the system, used to illustrate the system architecture. In CORAS a model is a special predefined diagram belonging to an activity in the risk management process, often defined in CORAS' UML profile. They are made to ensure that the right information is captured at the right place in the risk management process. Models in CORAS are therefore more like a subset of what IEEE-1471 defines as models. A corresponding term in CORAS' terminology for an IEEE1471-model is *element*.
As we concluded in the previous discussion about elements, if CORAS and IEEE-1471 do not agree in the interpretation of a term AMBRA will rely on the one found in IEEE-1471. By using IEEE-1471's interpretation of model, this also covers CORAS' meaning of model.

4.5 Viewpoint and view

IEEE-1471: According to this standard an architectural description is organized by one or more *views*. A view conforms to a set of rules defined by its corresponding *viewpoint*. The viewpoint establishes how a view is created, depicted and analyzed. The view addresses one or more of the stakeholders concerns and consists of models which express the system architecture. When constructing an architectural description, the architect first selects a set of viewpoints to use, and thereafter constructs the corresponding views. The standard imposes no restrictions on which viewpoints to use.

CORAS: CORAS is based on RM-ODP which divides the system documentation into five viewpoints, i.e. abstractions, which yields a specification of the whole system according to the type of viewpoint. RM-ODP defines a viewpoint as: 'Viewpoint (on a system) - A form of abstraction achieved using a selected set of architectural concepts and structuring rules (a viewpoint specification), in order to focus on a particular concern within the system' [ISO10746]. The five different viewpoints used in CORAS are: enterprise viewpoint, information viewpoint, computational viewpoint, engineering viewpoint, and technology viewpoint, for more details see Appendix B or [Putman01].

From a viewpoint you get a view of the system which addresses the areas of concern for a specific stakeholder. Each cross-pair of concern/viewpoint illustrates, with elements, information related to the viewpoint in the belonging activity.

Conclusion: CORAS is claimed to be a specialization of RM-ODP and therefore uses RM-ODP's interpretations of view and viewpoint. In terms of RM-ODP, a viewpoint provides the concepts and the structure for any system specification from that viewpoint. The use of that viewpoint, and refinement of the architecture concepts of that viewpoint, is considered a view' [Put01 p737]. 'The viewpoint s of RM-ODP can be somewhat related to IEEE-1471 viewpoint templates, and the *use* of the RM-ODP viewpoints related to a set of IEEE-1471 views' [Put01 p734]. While *view* is an important term in IEEE-1471, used for organizing the architectural description, it is only used briefly in CORAS to explain what a viewpoint is. CORAS is based on the five viewpoints defined in RM-ODP, whereas IEEE-1471 does not impose any restrictions on which viewpoints to use. The reason for RM-ODP to predefine five viewpoints is based on the assumption that these viewpoints are sufficient to meet most demands, but it is possible to use additional viewpoints.

Based on the fact that IEEE-1471 has a broader and less strict interpretation of view and viewpoint, AMBRA will use this in its architecture.

4.6 Intended users

Even though the term 'intended users' is not included in the terminology in AMBRA, we feel it is interesting to compare the intended users of the two frameworks. This will help us clarify, and possible verify, our assumptions about AMBRA's intended users.

.

IEEE-1471: The first class of intended users of IEEE-1471 are stakeholders in system development and evolution, typically:

- Those that use, own and acquire the system (users, operators, and acquirers or clients).
- Those that develop, describe and document architectures (architects).
- Those that develop, deliver and maintain the system (architects, designers, programmers, maintainers, testers, domain engineers, quality assurance staff, configuration management staff, suppliers and project managers or developers).
- Those who oversee and evaluate systems and their development (chief information officer, auditors and independent assessors).

The secondary class of users are:

- Those involved in the enterprise-wide infrastructure activities that span multiple system developments, including methodologists, process and process-improvement engineers, researchers, producers of standards, tool-builders and trainers.

CORAS: As opposed to IEEE-1471 we do not consider CORAS to have more than one class of intended users. The users of CORAS is divided into two groups and related to the use of the CORAS methodology for risk analysis:

- The risk analysis team, including the risk analyst.
- The normal stakeholders of the target of analysis, according to [AS/NZS4360] this type of stakeholders may be:
    - Individuals inside the organization, such as employees, management, senior management and volunteers.
    - Decision-makers, business or commercial counterparties, employee groups, union groups, financial institutions, insurance organisations, regulators and other government organisations that have authority over activities.
    - Politicians (at all level of government) who may have an electoral or portfolio interest, non-government organisations such as environmental groups and public interest groups, customers, suppliers, service providers and contractors to the activity.
    - The media who are potential stakeholders as well as conduits of information to other stakeholders.
    - Individuals or groups who are interested in issues related to the proposal, local community and society as a whole

The two groups may overlap as it is normal to include users of the target of analysis in a risk analysis.

Conclusions: In the question of which group of users AMBRA will address it is natural to consider the secondary class of IEEE-1471-users. This class includes methodologists, process and process-improvement engineers, researchers, producers of standards, tool-builders and trainers. AMBRA intend to be a reference framework facilitating development of model-based risk analysis methodologies and therefore useful for this group of users. Our main focus

is on this group, but also people from the first class of IEEE-1471-users, who use a methodology developed according to AMBRA could be interested in the framework itself, e.g. for understanding the terminology better.

Figure 4-7 tries to illustrate how the users of CORAS and the users of AMBRA relate to the users of IEEE-1471. CORAS is a methodology and therefore typically address a similar group of users as the first class of IEEE-1471 users. In addition to addresses the first class of IEEE-1471 users, AMBRA mainly addresses users in the second class who e.g. develop methodologies.

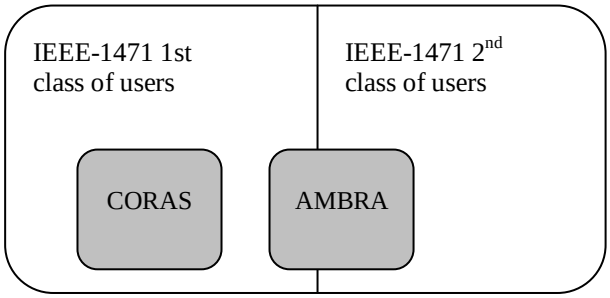

Figure 4-7 Users of CORAS vs. AMBRA

5 Formalizing the AMBRA Terminology

The objective of this Chapter is to formally define the terminology using UML class diagrams.

Throughout this thesis we use the term *risk analysis* in a way that is not fully in conformance with the "Australian Standard for Risk Management" [AS/NZS4360] we claim to follow. A *risk analysis* is actually considered as a part of the *risk assessment* in a full *risk management process* as illustrated in Figure 5-1 from the [AS/NZS4360]. Still we continue to use our interpretation as we believe this makes the terminology more understandable for most people. The risk management process is defined as *'A systematic application of management policies, procedures and practices to the task of establishing the context, identifying, analysing, evaluating, treating, monitoring and communicating risks'.*

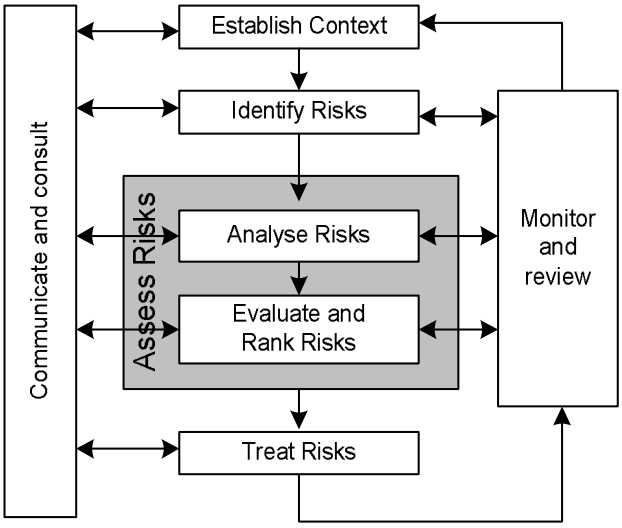

Figure 5-1 The Risk Management Process with sub processes

[AS/NZS4360] divides the risk management process into sub processes like *Establish context*, *Identify risks* and *Analyse risks* as illustrated in Figure 5-1. CORAS uses this structure, but specifies each of the sub processes to consist of one or more risk management activities, which are further specified into concerns.
Deciding the content of the activities, i.e. which risk analysis techniques to use, is not in the scope of AMBRA. This may vary from one risk analysis methodology to another, and is therefore left open in AMBRA. AMBRA will restrict itself to the superior risk management process with its sub processes and activities, giving a detailed conceptual specification of a risk management activity.

5.1 UML specification of AMBRA

The conceptual specification of architectural description in IEEE-1471 (Figure 2-3), is in the following combined with the conceptual specification of CORAS' risk documentation framework (Figure 2-5).

Figure 5-2 specifies the 'top level' of AMBRA, the *Risk Management Process* consists of *Sub processes* consisting of *Risk Management Activities*, and is another way of illustrating Figure 5-1.

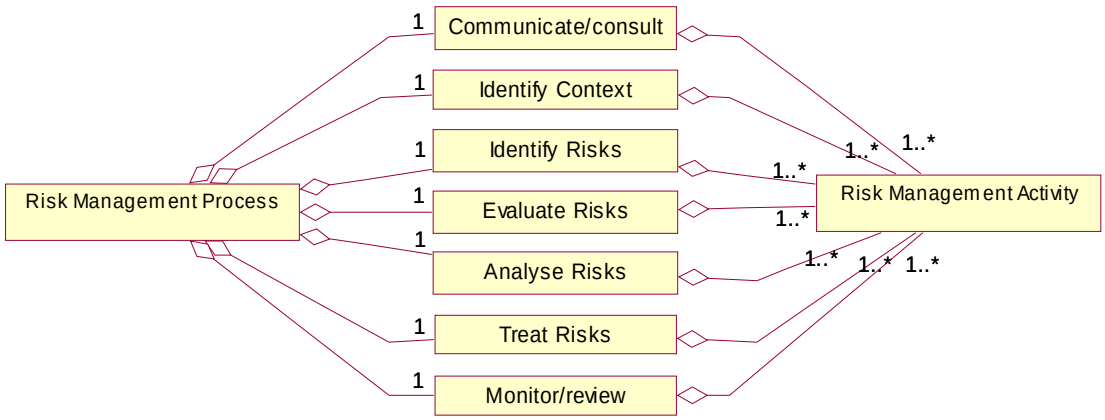

Figure 5-2 The risk management process

In the following we give an overview of general concepts addressed by each Risk *Management Activity*. According to IEEE-1471 every system has an architecture which can be viewed from different viewpoints, described by models in an architecture description. As we explained in the previous Chapter there are several similarities between the terminology in the documentation framework in CORAS and IEEE-1471. The goal is to create a specialization of IEEE-1471 which includes elements from risk analysis.

The original IEEE-1471 specification (Figure 2-3) needs some changes in terminology to adapt to the risk analysis context of this thesis. To make it more surveyable we start with a simplified version of the specification[5] in Figure 5-3, adding more elements as they become of interest.

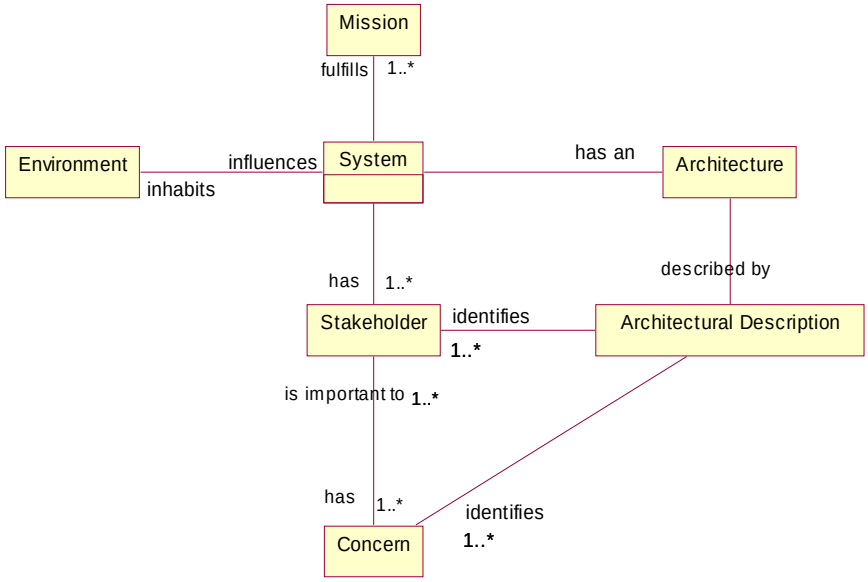

Figure 5-3 A simplified version of the IEEE-1471 conceptual specification

First we change the general term *System* to *Target of Risk Analysis*, but keep the original definition from IEEE-1471: '*the term system encompasses individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises and other aggregations of interest*'.

[5] For all specifications in this Chapter (except Figure 5-10) the following applies: if no cardinality is displayed the default value is 1.

The change from *System* to *Target of Risk Analysis* is made only to keep focus on risk analysis, the specification still conforms to IEEE-1471. The *Target of Risk Analysis* still fulfils a *Mission*, inhabits an *Environment,* has an *Architecture* and has *Stakeholders* like in the original specification.

The *Environment* in IEEE-1471 is now the environment for the *Target of risk analysis*. In risk analysis the *Environment* is often called *Risk Analysis Context*, which is the external conditions influencing the *Target of Risk Analysis*. Since these terms have the same interpretation, we will change *Environmen*t in the specification to *Risk Analysis Context*.

Then we include the term *Stakeholder* as in the original specification, where stakeholders can be everything from organizations or regular users to risk analysts. The stakeholders have different concerns regarding the *Target of Risk Analysis* related to what they find valuable or important by the 'target'. Concerns are adressed by using architectural models which express the parts of the 'target' answering the stakeholder's concern. We may also regard the parts of the system which the stakeholder find interesting or valuable as an asset for the stakeholder. Based on this we also add the notion of *Asset* to our conceptual specification. The relationship between *Concern* and *Asset* in the specification is called 'A Concern is related to an Asset'.

The original specification from IEEE-1471 has now changed in form of renaming some of its terms and gained one additional element as displayed in Figure 5-4:

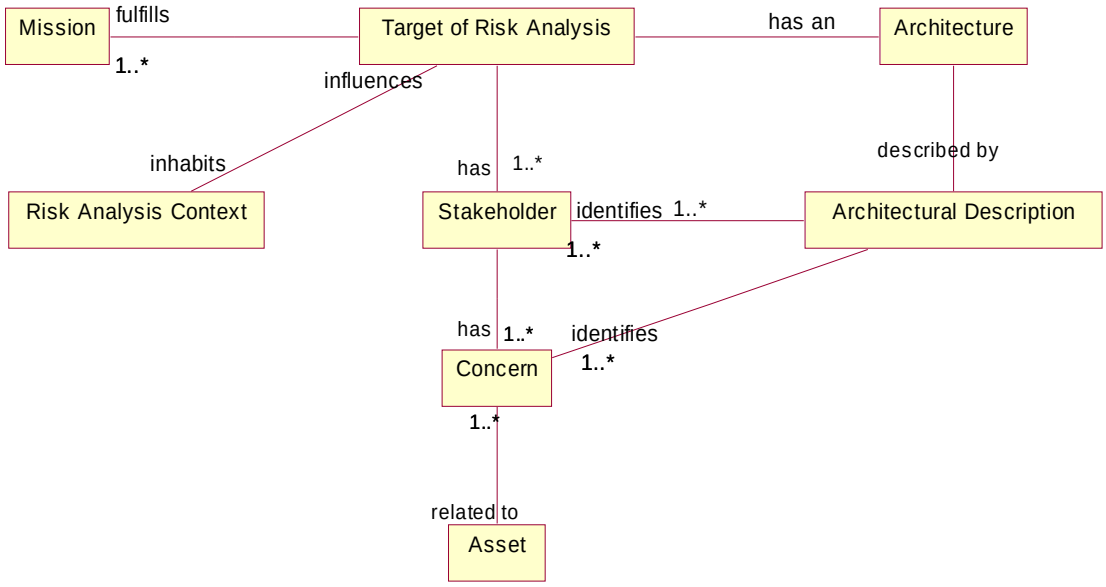

Figure 5-4 IEEE-1471 conceptual specification with changes

The conceptual specification in Figure 5-4 is fairly simplified and need to be extended with more details. The way this will be performed is first through specializing the term *Stakeholder*, then *Concern* and including these refinements in Figure 5-4.

As we have concluded in Chapter 4, we divide stakeholders in two types: the *Risk Analysis Stakeholder* who is concerned about the risk analysis itself and the *Target Stakeholder* who is mostly concerned about the target of risk analysis. Both types are stakeholders in an IEEE-1471-sense, but based on their difference in concerns we specialize the *Stakeholder* term into two types, illustrated in Figure 5-5.

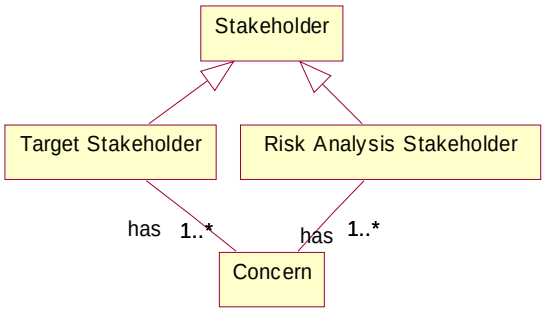

Figure 5-5 Specialization of stakeholder

A stakeholder concerned about the *Target of Risk Analysis* itself, may be a regular operator of the system assessed. The concerns are typically related to what he or she recognizes as *Assets* in the actual system, like the business rules for a transaction or communication between two system components. *Target concerns* are addressed by means of having the appropriate architectural description of the 'target'.

The other type of stakeholder, related to the actual risk analysis and the outcome of it, may be the risk analysis expert who organizes and conducts the risk analysis, the members of the organization or different types of decision makers. This stakeholder is concerned about the risk analysis results itself in order to draw the correct conclusions. This could be about getting sufficiently good results in each of the different steps in a risk analysis or exploiting the results to make the *Target of Risk Analysis* more secure.

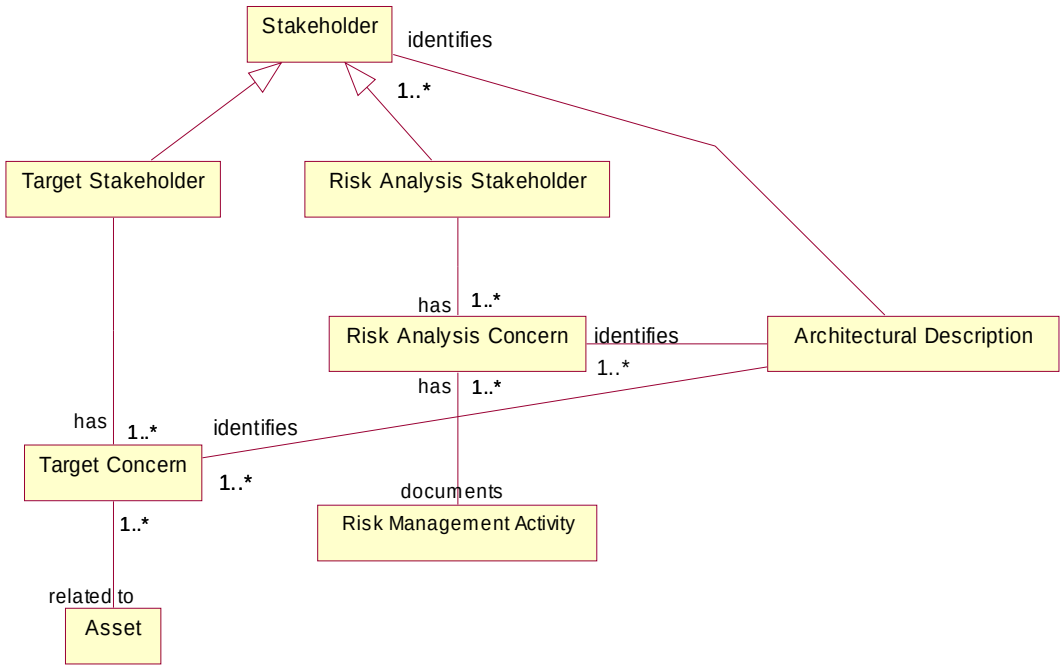

Figure 5-6 Specialization of *Concern* and *Stakeholder*

Figure 5-6 displays both the specialization of Stakeholder and Concern and how they relate to *Asset* and *Risk Management Activity*.

If we then combine the specialization of stakeholders and concern (Figure 5-6), with the general overview of a risk management activity (Figure 5-4), it results in Figure 5-7. Still the

term *Architectural Description* lacks refinement and only a few elements from risk analysis are yet included in the specification.

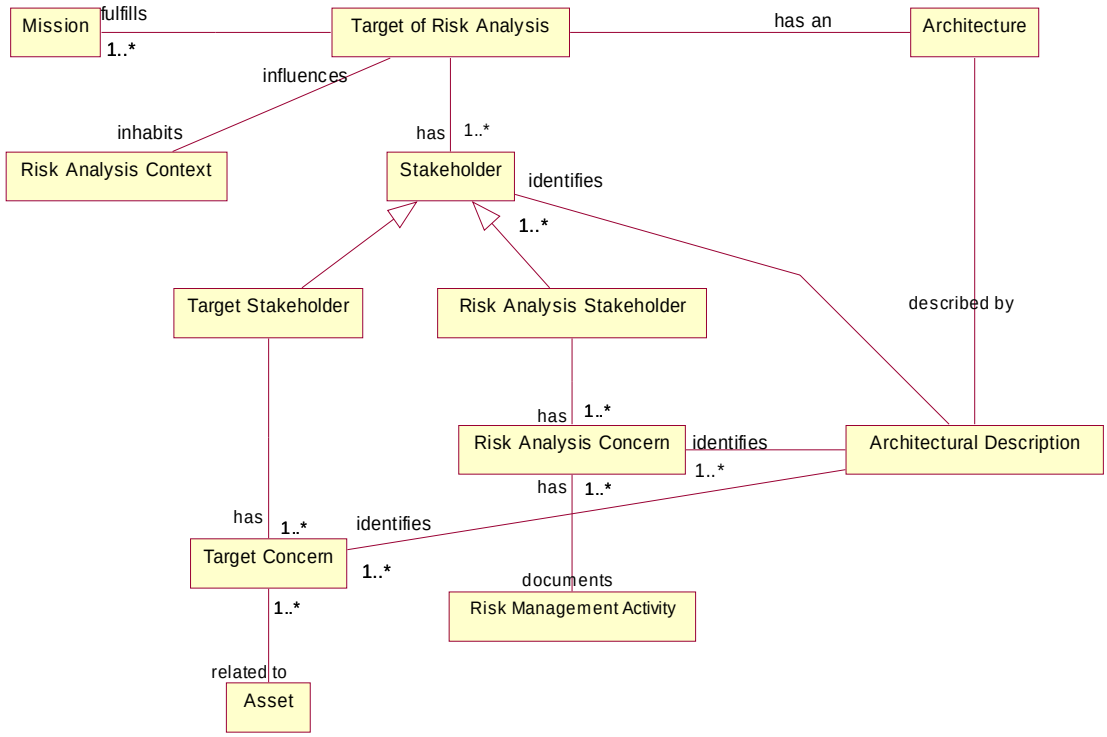

Figure 5-7 Combining Figures

IEEE-1471 specifies the content and structure of an architectural description Figure 5-8. This is not the complete conceptual specificationl from Figure 2-3, but it displays concepts directly related to the architectural description. This specification extends Figure 5-7 by specifying the term *Architectural Description*.

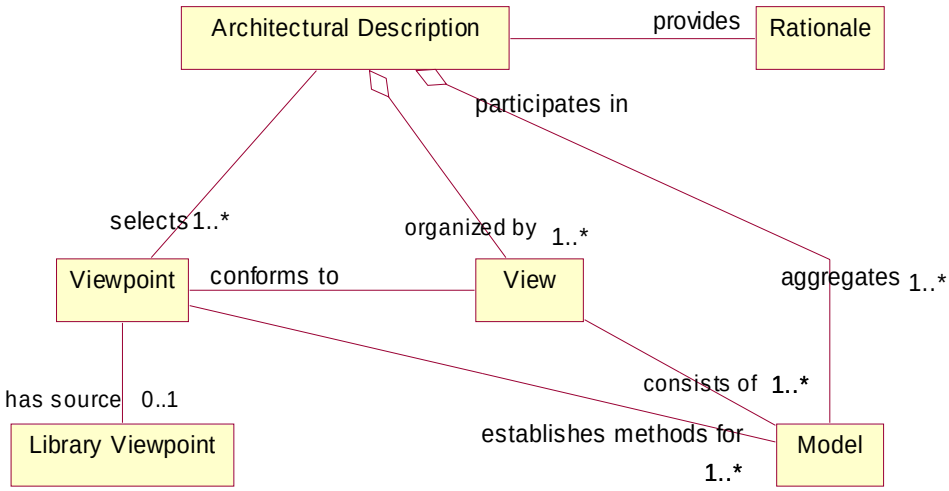

Figure 5-8 Concepts related to an architectural description from IEEE-1471

There are still several terms from risk analysis to be included. We have addressed the term *Asset*. *Assets* are entities related to target of analysis which a stakeholder finds valuable [Braber et.al. 03]. In a risk analysis context the goal is to discover risks, threats and vulnerabilities for the asset with appropriate treatment to improve the system. Related to the

Asset, CORAS defines several elements like *Risk*, *Threat* and *Treatment* as displayed in Figure 5-9 (except for *Stakeholder* which already is included in our specification). For a detailed explanation of the terms in the Figure see Section 2.4.

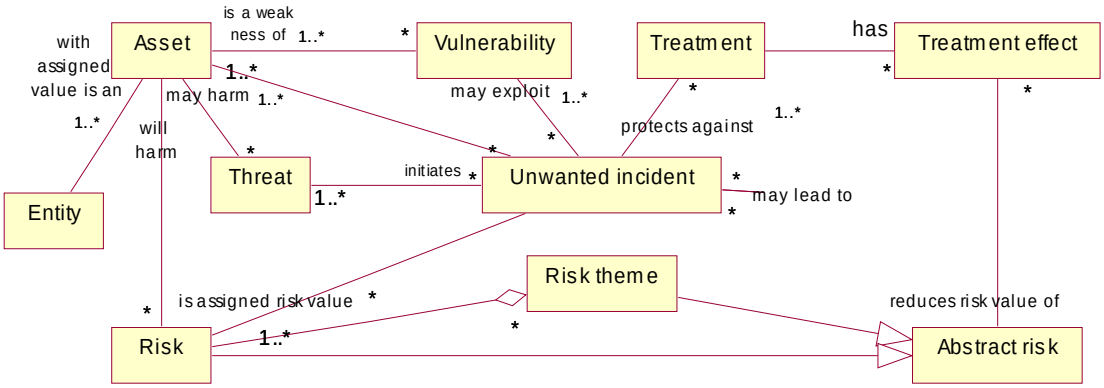

Figure 5-9 Asset, risk, threat and vulnerability

Next we combine the concepts from IEEE-1471 (Figure 5-8), and the elements from risk analysis (Figure 5-9) with our conceptual specification (Figure 5-7). The result is a conceptual specification for model-based risk analysis. The complete specification is illustrated in Figure 5-10. To increase the readability and making the specification less complex the two parts with concepts related to respectively architectural descriptions and risk analysis are shaded in different colours and framed with dashed lines. In addition only cardinalities of *new* relationships are displayed to avoid complicating the Figure.

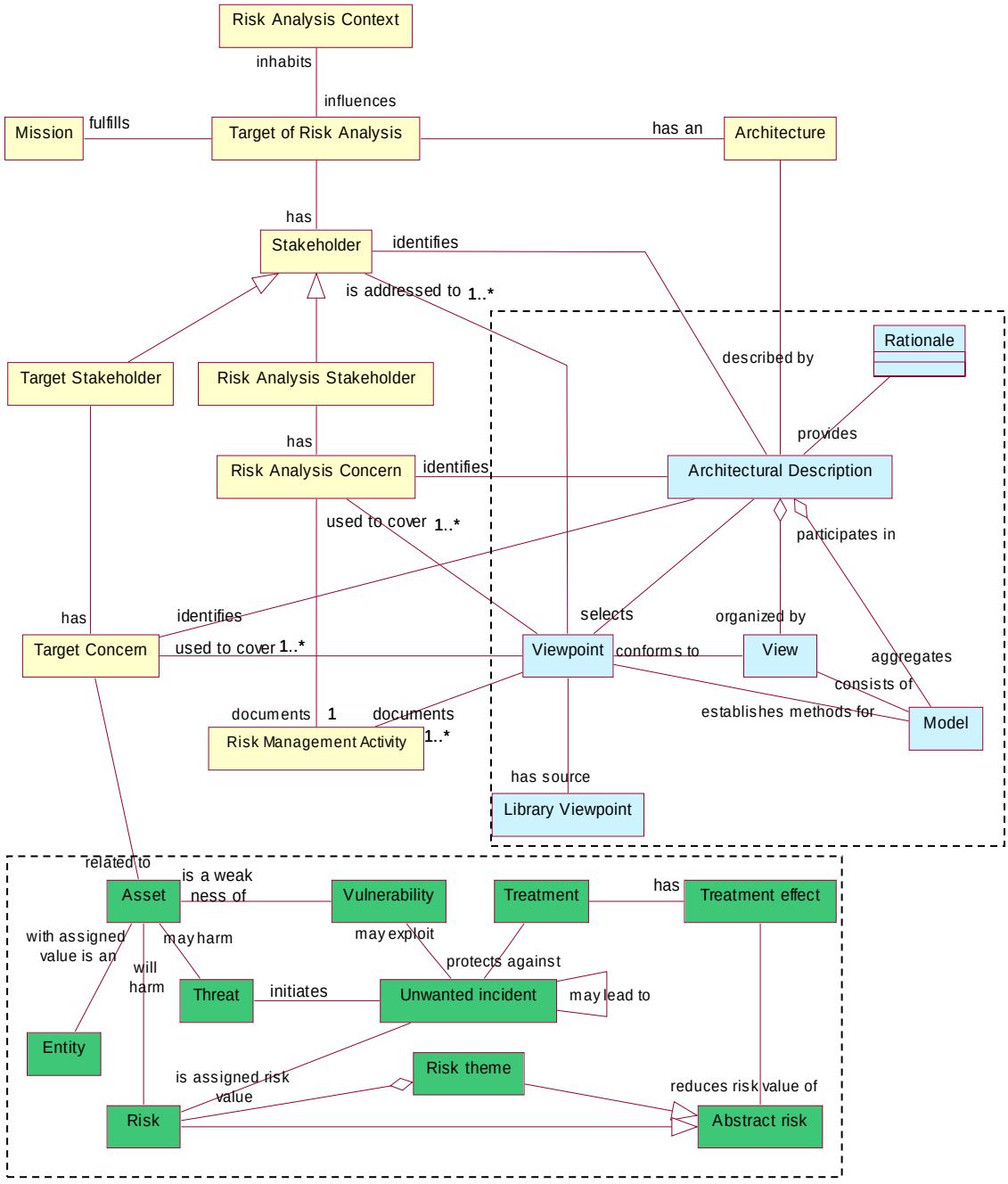

Figure 5-10 Terminology for a *Risk Management Activity* in AMBRA

5.2 Elements in AMBRA

This part contains an alphabetic list of the elements used in AMBRA.

1. **Abstract risk**
   An abstract risk is '*the common properties of risks and risk themes, such as risk value'* [Lund et.al. 03]. The term is taken directly from CORAS and has kept its original interpretation and name.
2. **Architectural Description**
   IEEE-1471 claims that every system (in this setting system is called Target of Risk Analysis) has an architecture which you can describe by an architectural description. The term is defined as '*a collection of products to document an architecture*' [IEEE1471].
3. **Architecture**
   According to IEEE-1471 every system (Target of Risk Analysis) has an architecture, which is defined as '*the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution*'[IEEE1471].
4. **Asset**
   An asset is anything that has value to the organization [ISO13335]. The assets of an organization include:
   - physical assets (e.g., computer hardware, communications facilities, buildings),
   - information / data (e.g., documents, databases),
   - software,
   - the ability to produce some product or provide a service,
   - people, and
   - intangibles (e.g., goodwill, image).
5. **Entity**
   'An entity is a part of or feature of the system. An entity becomes an asset when assigned value by a stakeholder' [Lund et.al. 03].
6. **Library Viewpoint**
   'A viewpoint definition may originate with an architectural description, or it may have been defined elsewhere. A viewpoint that is defined elsewhere is referred to as a library viewpoint' [IEEE1471].
7. **Mission**
   In IEEE-1471 every system (here: Target of risk analysis) has a purpose, a mission, which is to be recorded. A mission is defined as a *'use or operation for which a system is intended by one or more stakeholders to meet some set of objectives*' [IEEE1471].
8. **Model**
   According to IEEE-1471 '*a view may consist of one or more architectural models. Each such architectural model is developed using the methods established by its associated architectural viewpoint. An architectural model may participate in more than one view'* [IEEE1471].
9. **Rationale**
   From IEEE-1471's perspective an architectural description (AD) should include '*a rationale for the selection of each viewpoint. An AD should provide evidence of the consideration of alternative architectural concepts and the rationale for the choices made. The rationale should address the extent to which the stakeholders and concerns are covered by the viewpoints selected under this clause*' [IEEE1471].
10. **Risk**

A risk is '*the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization'*. [ISO13335]
A risk for an asset is *'an unwanted incident that has been assigned a consequence value, a frequency value, and a resulting risk value'* [Lund et.al. 03].

11. **Risk Analysis**
    'The process of identifying security risks, determining their magnitude, and identifying areas needing safeguards' [ISO13335]
12. **Risk Analysis Concern**
    This is concern on a meta-level of the target, mostly related to the risk analysis itself and the results.
13. **Risk Analysis Stakeholder**
    This is a stakeholder like the risk analyst whose main focus is on the risk analysis or the outcome of the analysis.
14. **Risk Assessment**
    '*Assessment of threats to, impacts on and vulnerability of information and information processing facilities and the likelihood of their occurrence'* [ISO17799].
15. **Risk Analysis Context**
    This was originally called 'Environment' in IEEE-1471, but renamed to Risk Analysis Context in order to conform to the risk analysis setting in AMBRA. The term has kept its original interpretation from IEEE-1471 where the Environment '*determines the setting and circumstances of development, operational, political, and other influences upon the system. The environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. The environment determines the boundaries that define the scope of the system of interest relative to other systems*' [IEEE1471].
16. **Risk Management**
    This is the *'process of identifying, controlling and minimizing or eliminating security risks that may affect information systems, for an acceptable cost'* [ISO17799].
17. **Risk Management Activity**
    This is a division of the sub processes into smaller components called risk management activities. [Lund et.al. 03]
18. **Risk Management Process**
    This is '*a systematic application of management policies, procedures and practices to the task of establishing the context, identifying, analysing, evaluating, treating, monitoring and communicating risks*' [AS/NZS4360].
19. **Risk theme**
    A risk theme is '*a categorisation of similar risks, assigned its own risk value*' [Lund et.al. 03].
20. **Stakeholder**
    '*A system has one or more stakeholders. Each stakeholder typically has interests in, or concerns relative to, that system and can be an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system*' [IEEE1471].
21. **Sub process**
    One or more sub processes constitute a risk management process. A sub process consists of one or more risk management activities [Lund et.al. 03].
22. **Target Concern**
    This refers to concerns about the actual target of risk analysis (e.g. a system or component). The term holds the same meaning as 'Concern' in IEEE-1471: '*are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders. Concerns include*

system considerations such as performance, reliability, security, distribution, and evolvability' [IEEE1471].

23. Target of Risk Analysis

This is the actual system or component assessed during the risk analysis. The term has been renamed from 'System' in IEEE-1471 to 'Target of Risk Analysis' in AMBRA. AMBRA uses the original interpretation where a system is '*a collection of components organized to accomplish a specific function or set of functions'* [IEEE1471].

24. Target Stakeholder

This is a stakeholder with concerns related directly to the target of analysis on system-level. This could be '*an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system'* [IEEE1471].

25. Threat

'A threat is a potential cause of an unwanted incident, which may result in harm to a system or organization' [ISO13335].

26. Treatment

'*Applying a treatment on the target is a way of reducing the risk value of a risk or risk theme'* [Lund et.al. 03].

27. Treatment effect

'The treatment effect is a treatments capability to reduce the risk value of a particular risk' [Lund et.al. 03].

28. Unwanted incident

'An *unwanted incident is an undesired event that may reduce the value of an asset* [Lund et.al. 03].

29. View

A view is '*a representation of a whole system from the perspective of a related set of concerns. Each view addresses one or more of the concerns of the system stakeholders. The term view is used to refer to the expression of a system's architecture with respect to a particular viewpoint.'* [IEEE1471].

30. Viewpoint

A viewpoint is '*a specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis'* [IEEE1471].

31. Vulnerability

'Vulnerability is a weakness with respect to an asset or groups of assets which can be exploited by one or more threats' [ISO13335].

6 Evaluation of AMBRA against Success Criteria

In Chapter 4 we formulated several success criteria for AMBRA. This Chapter will evaluate AMBRA against these criteria, discussing whether they are fulfilled or not.
In short the criteria of success are: conformance to IEEE-1471, conformance to CORAS, conformance to MDA, conformance to UML and usefulness.

6.1 Conformance to IEEE-1471

One of the requirements we put on AMBRA is for it to include the ideas from IEEE-1471 about how to create good descriptions of system architecture. IEEE-1471 includes central aspects related to architectural descriptions with focus on the individual stakeholder. An architectural description consists of models of the system the stakeholder have to understand. It is important to adjust the models according to the stakeholder's interests and knowledge. This is especially important in a risk analysis where you are dependent on the stakeholders understanding of the system to detect all system weaknesses and threats.

IEEE-1471 includes two main parts: a conceptual framework of an architectural description and a part describing practices for architectural description, i.e. rules for conformance with IEEE-1471. The latter part would be natural to include in a *methodology* developed according to AMBRA, but too detailed for AMBRA itself. We choose to concentrate on the conceptual model of an architectural description which in our opinion illustrates the most important aspects of IEEE-1471. Further on we extended this with concepts from risk analysis. We decided to connect the model from IEEE-1471 with terms from risk analysis by relating 'IEEE-1471-concern' to 'CORAS-asset' from risk analysis. Where a stakeholder considers a part of a system important it means the stakeholder is concerned about the part. This part could be characterized as something valuable, in risk analysis this would be recognized as an entity assigned value and called 'asset'. This motivates why we believe relating 'concern' to 'asset' is a correct decision.

When we formulated this success criterion we allowed for renaming terms in IEEE-1471 if this made it easier to understand AMBRA or increased the focus on risk analysis. We did not find any reasons for changing the original terms used in IEEE-1471 in order to increase the ability to *understand* AMBRA. On the other hand we found reasons for renaming a few terms to make them more influenced by the risk analysis terminology.

In IEEE-1471 'Environment' refers to *'the setting and circumstances of development, operational, political, and other influences upon the system. The environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. The environment determines the boundaries that define the scope of the system of interest relative to other systems'*. We felt this was too general without focus on risk analysis and did not feel we made any mistake by renaming it to 'Risk analysis context' in AMBRA. Of the same reason we changed IEEE-1471's 'System' into 'Target of Risk Analysis'. In IEEE-1471 the target is subject for a description, but in AMBRA the target will also be subject for a risk analysis, and the change in name emphasizes this. We have kept the original definition of 'system' from IEEE-1471 even if we have renamed the term.

Originally the conceptual model in IEEE-1471 only had a general *Stakeholder*, but we chose to specialize the term into two kinds, *Target Stakeholder* and *Risk Analysis Stakeholder*. This decision was based on the fact that we believe it is important to distinguish between stakeholders related to the target system at 'system-level' and stakeholders related to the

system on a ‘meta-level’ like the risk analyst. As a consequence of specializing the stakeholder term, we also had to divide ‘concern’ according to which stakeholder is concerned. The target stakeholder would typically have concerns at system-level, while risk analysis stakeholders could have concerns about the risk analysis process itself, thus a meta-level of target system. The specialization of stakeholder and concern has not changed any relationships in the original model. Stakeholders still exists in the model, representing attributes in common between Risk Analysis stakeholder and Target stakeholder. In the original model concern was related to stakeholder, architectural description and viewpoint. These relationships still exists in AMBRA, in addition we have connected Target concern to asset and risk analysis concern to risk documentation framework. Viewpoint is also connected to the risk documentation framework in order to make it possible to structure models according to which viewpoint they belong to.

Based on this we believe we have succeeded in achieving compliance with IEEE-1471.

6.2 Conformance to CORAS

AMBRA will be a high-level framework rather than a concrete methodology like CORAS. We had no intention of using every part of CORAS, because AMBRA should not include a detailed methodology but be a reference framework for development of model-based risk analysis methodologies. The parts of CORAS we found useful is the documentation framework and the organization of the risk management process. The first illustrates the artefacts CORAS think is important to document and store and the latter gives us the structure of a complete risk management process. AMBRA defines the concepts a risk managment activity should take into consideration, but not the exact techniques to use. To choose the techniques used in a risk analysis methodology is up to the developer of the concrete methodologies.

There are written several papers and reports about CORAS we chose to use mainly [Braber et.al. 03] and [Lund et.al. 03]. [Lund et.al. 03] describes the CORAS risk documentation framework (Figure 2-4). The risk documentation framework is a specialization of RM-ODP and helps linking together related information within the five RM-ODP-viewpoints (see Figure 2-4) A proper documentation framework is especially important for the risk analysis stakeholder to facilitate reuse and maintenance.

A conceptual model of this documentation framework (Figure 2-5) is the basis for extending the conceptual specification of an architectural description in IEEE-1471 (Figure 2-3). The complete specification was connected to the specification in IEEE-1471 by relating CORAS-asset to IEEE-concern.

Based on this we feel we have succeeded in achieving compliance with CORAS.

6.3 Usefulness

Based on our limited access to resources and the time constraints imposed on this work, we decided to conduct a judgment study of AMBRA by asking five, in our opinion well skilled, persons to judge AMBRA. We provided the participants with a questionnaire (see Appendix A) asking several questions about AMBRA and requiring detailed answers. Ideally we would have liked like to see an evaluation of AMBRA not only from this theoretical point of view, but rather in a situation where someone uses the framework for its intended purpose e.g. through a field study.

In order to evaluate AMBRA's usefulness for indirect users (both I and II) we first need a risk analysis methodology developed according to our framework. Accordingly there will not be made an evaluation of usefulness for this group, but we will strongly recommend this as subjects for future work. The evaluation of AMBRA with respect to usefulness will therefore mainly include direct users. The judges are all involved in research on topics like system development, risk analysis or architecture and suitable for judging AMBRA from the 'direct user' point of view.

The judges were asked to state whether they considered AMBRA as a suitable reference framework for developing risk analysis methodologies. Because they were asked to judge AMBRA in a somewhat early phase of development, they were not provided with a complete AMBRA framework. This made their job more difficult than if they were to evaluate the thesis after completion since the version they received had several unfinished parts. Some of the judges' main reasons for not assigning AMBRA top score on usefulness were:

- '*The generalization of CORAS may not be adequately general, restricting the flexibility of AMBRA.*' (This aspect is discussed in the next Chapter)
- '*It needs more description related to what AMBRA is, how it is used and what is accomplished by using it.*' (In later versions of this thesis these aspects have been addressed more clearly)
-

Most of the judges gave AMBRA a score between 4 and 5 (6 was maximum) on the question of whether they thought AMBRA was a suitable framework. Even though none of the judges recognized AMBRA as a perfect reference framework, we do not believe AMBRA has failed in its mission. We do not claim to have "reinvented the wheel", AMBRA is meant to improve and strengthen an already existing risk analysis methodology (CORAS) by including a standard for architectural description. This will make it less specialized, i.e. leverage it into a higher level of abstraction, and therefore more suitable as a reference framework. In the same manner as no one can expect the first version of a product to be perfect, we believe AMBRA needs to be further assessed and probably adjusted according to these assessment results.

Since we only rely on a theoretical evaluation, we cannot be sure if our assumptions about the practical usefulness of AMBRA are right. There may be aspects we have omitted or we may have included concepts with ambiguous interpretation which may lead to confusion. A trial where a group use AMBRA in development of a risk analysis methodology would be extremely valuable. Afterwards this methodology should be assessed by the indirect users (both group I and II), providing essential feedback helping to improve AMBRA.

6.4 Conformance to MDA

The reason for requiring compliance with MDA is to ensure focus on *model-based* risk analysis. This requirement originates from a desire to include concepts from model-based system development where the main feature is the extensive use of reusable models in the development process. The use of models in risk analysis could improve quality and, effectiveness by facilitate precision, communication and interaction between stakeholders. By creating general models applicable across different platforms it could also reduce development costs by increasing the possibility of reuse.

The official specification of MDA is claimed to be somewhat vague and incomplete, therefore we base us upon the explanation of MDA given in [Runde,Stølen 03] as our reference for MDA. In this article MDA is described as "*a framework for defining system architecture development methodologies by giving directions for making architectures, serving as a tool-*

box for system developers and system architects, and facilitating integration and interoperability between systems". AMBRA is meant to be a possible, future extension of a framework like this. In order to easily fit into the context of MDA have adopted the same organisation as MDA specified in [Runde,Stølen 03] where the MDA framework is divided in three parts: a terminology, a library of reusable elements and a methodology.

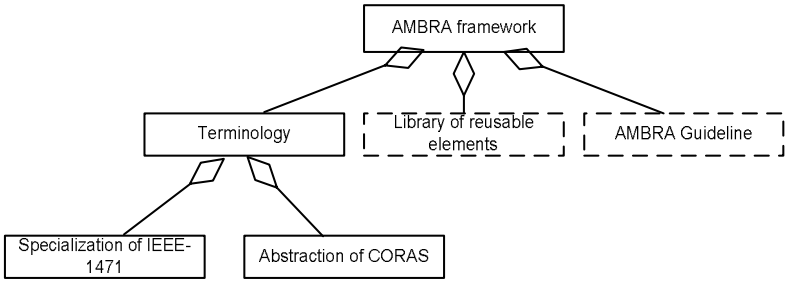

Figure 6-1 AMBRA, including the remaining parts

At this time AMBRA includes a complete terminology for model-based risk analysis based on the terminology in IEEE-1471 and CORAS. This terminology is compliant with the terminology in [Runde,Stølen 03]. The remaining two parts are proposed as subjects for further work and will be discussed in Chapter 7 "Discussion".

We claim AMBRA as defined to this point is compliant with MDA defined in [Runde,Stølen 03].

6.5 Conformance to UML

In Section 3.1.6 we explained why we think expressing AMBRA in a world-wide accepted modelling language like UML is important. The core part of AMBRA consists of conceptual specifications expressed in UML class diagrams followed by a textual explanation. Every Figure is illustrated using a combination of class entities and different types of relationships with associated role names and cardinalities. In cases where the cardinalities of relationships are not displayed the default value is "1" (explained in footnotes).
We have used pure UML notation (version 1.4) to define the different parts of AMBRA and therefore successfully fulfilled our criterion about conformance to UML.

6.6 Success of Superior criterion

When we defined the superior criterion of success we concentrated of the overall objectives for this thesis. It was stated that AMBRA should be a first step towards ***a supporting architectural framework for model-based risk analysis.*** This criterion was specified in more detail in terms of three requirements:

a) *AMBRA should facilitate reuse of best experiences from risk analysis*
   We believe this is substantiated by our requirements to conformance to MDA, and adoption of the concept of documentation framework from CORAS for storing risk analysis results. Reuse is also facilitated by the use of standardized UML.
b) *AMBRA should provide a common basis for risk analysis within different domains*
   We believe this is substantiated by conformance to MDA, the development of a general terminology based on CORAS and IEEE-1471, the use of UML and requirement of usefulness from a practical point of view.
c) *AMBRA should be a reference framework for the development of risk analysis methodology*

We believe this is substantiated by the requirements of AMBRA being useful from a practical view, conformance to MDA when completed with the two remaining parts, our foundation in CORAS and use of UML.

7 Discussion

Throughout the development of AMBRA we have made several design decisions. In the following we discuss the most important of these in further detail, formulated as questions and answers.

7.1 Why do we need AMBRA when we could adapt an existing methodology to fit the intended domain?

In many ways this could be the easiest, especially if the existing methodology is made for a very similar domain. The drawback is that many existing methodologies are highly specialized for the intended domain, e.g. CRAMM [CRAMM] for the health sector, making it difficult to adjust them to a new. Also it could possible lead to inheritance of the original methodology's faults and weaknesses. If the methodology was equipped with computerized tools, they may be expensive to change. Our goal is that a methodology developed according to AMBRA's model-based framework would be easier to adapt to other domains. This job would be facilitated through a reference framework, providing a terminology and structure in addition to a library of predefined and reusable elements or components to use in the methodology. The use of an architectural framework like AMBRA ensures interoperability between risk analysis methodologies.

7.2 Is it necessary to include a standard for architectural descriptions in risk analysis?

We think models are important in risk analysis, and we recognize architectural descriptions as very informative models of target system. We believe models are useful, as they help describing the target at right level of details, they facilitate communication between stakeholders, and they also promote easier reuse and maintenance. Even though we first think architectural descriptions as illustrations of the system, also log tables, risk analysis tables and other documentation are architectural descriptions and therefore models. IEEE-1471 comprises common and consensus elements from architectural descriptions and is claimed to gradually gain acceptance in the IT-industry. To use a standardized way of describing the architecture you are making your work more accessible and understandable for a larger audience.

7.3 Is AMBRA too general?

At this point AMBRA is not a complete architectural framework for model-based risk analysis e.g. the guideline is still to be defined. The scope of this thesis is to develop the terminology and the main structure, only outlining the parts we leave for further work. Some may therefore claim that AMBRA is too general. We do not agree, the terminology in AMBRA is meant to be general in order to be useful on different domains in risk analysis. One could claim that AMBRA will provide insufficient directions for the developer with respect to which techniques (e.g. HazOp or FMECA) a risk analysis methodology should include. We think this would be to specify too much, making the framework less general than desirable. We expect the developer to choose the right risk analysis techniques for the domain, not to follow a predefined set in an explicit order specified in AMBRA.

7.4 On the other hand is AMBRA too specific?

During the judgment study there was expressed concerns about whether AMBRA was too specific, restricting the flexibility of the framework or not. The best way to test this is through practical use of AMBRA. Like we discussed in Section 6.3, we only have basis in theoretical arguments for how useful AMBRA is. A practical study like a field study would probably

revealed whether AMBRA is too specific. Arguments against IEEE-1471 are often based on that people consider it to be too general. This might lead to the conclusion that terms in AMBRA which originate from IEEE-1471 are not to be recognized as too specific. Someone may think the terms originating from CORAS (risk related terms) are too specific, but we did not feel like merging or excluding any of them. We believe they have all been subject for thoroughly research in CORAS and if any of them were unnecessary or unclear this would have been discovered during development of CORAS. If we did not specialize stakeholder and concern we would, according to our opinion, loose focus on *risk managemen.* Our point is to state clearly the difference between concerns to the ones responsible for conducting the risk analysis (e.g. risk analysis expert) and the participants (e.g. target operators etc.).

7.5 What about the use of evaluation strategies, are they appropriate?

'Formal theory' and 'judgment study' were chosen to help evaluating our success criteria. The strategies were chosen among eight different and generally accepted methods. The reason for restraining us to use only two of them was mainly because the other six either required a larger group of test personnel, more time or extensive use of research equipment like laboratories. A strategy like a field study would have required a considerable amount of time and effort from the participants. We do not think the six research strategies are useless for evaluating AMBRA; on the contrary we believe some of them could be well suited for further work (discussed in Chapter 8). According to [McGrath 84] the different strategies have their strengths in different areas. 'Formal theory' and 'judgment study' emphasize respectively precise measurement of behaviour and generality over actors. In order to have a statistically valid result from our evaluation we ought to include a strategy with focus on testing AMBRA in a system realistic environment. Strategies with maximum concern with system character of context are e.g. field studies or field experiments. If we had access to satisfactory resources we believe a field study could have been interesting. Through a field study we could investigate how useful AMBRA is as a supporting framework and vehicle for model-based risk analysis methodology development.

7.6 Could IEEE-1471 have been used in a different way?

Could we have connected the conceptual specification of an architectural description in IEEE-1471 with the risk analysis specification in a different way? We do not believe IEEE-1471 could have been used differently, at least not without adding more terms to the terminology. The way we combine the two specifications was done by using already existing terms without introducing completely new. We found it useful to specialize 'stakeholder' and 'concern', but these are changes made to improve the comprehensibility of already existing terms. For a while we thought of connecting 'asset' directly to 'concern', but this would have neglected the concerns of the risk analyst.

We considered not changing *Environment* into *Risk Analysis Context*, and instead recognize *Risk Analysis Context* as a specialization of *Environment*. We also considered including *Target of Analysis* as a specialization of *System*. Both of these alternatives were rejected because they would have made the conceptual model too complex and reduced focus on risk analysis. Instead the names were changed from *Environment* and *System* to *Risk Analysis Context* and *Target of Analysis.*

7.7 Could we have used CORAS differently?

The parts of CORAS we have concentrated on in AMBRA are the documentation framework and the risk management process based on [AS/NZS4360] and [ISO17799].

In addition CORAS also includes techniques from risk analysis methods, patterns and methodology for UML oriented modelling, a platform for tool-inclusion and an integrated risk

management and system development process. In our opinion the risk management process and the documentation framework were the only natural parts of CORAS to use for our purpose. The rest are too specific and detailed for the type of "architectural framework" AMBRA is meant to be. The integrated risk management and system development process of CORAS could for instance be the result of using AMBRA.

7.8 What about the remaining parts of AMBRA, in order to comply with MDA?

The main feature in model-driven development is extensive use of models in system development. The aim is to raise the level of abstraction so that models, rather than implementation code, become the main units of reuse. In the same manner AMBRA emphasizes the use of models in risk analysis.

To make AMBRA compliant with MDA we have an overall objective to adopt the organization we find in the MDA framework in [Runde,Stølen 03]. Of the three parts this framework consists of, we have only developed the MDA compliant terminology in this thesis. If we were to describe what we think the two remaining parts (AMBRA guideline and library of reusable elements) should include, this would be based on the ideas of [Runde,Stølen 03]. According to [Runde,Stølen 03] the methodology part should include a concrete methodology of MDA with aspects like compliance checking, conformance checking and instantiation of library elements. We find this as a sensible starting point for our guideline, but it might need adaptations in order to change focus from system development to risk analysis. We will not investigate which adaptations are necessary now, but leave it for further work with AMBRA.

In [Runde,Stølen 03] the library is described according to MDA's suggestions where reusable elements are categorized as either "Pervasive services", "Domain specifications", "Transparencies" or "Quality of service". We believe these categories are all possible to reuse in the risk analysis domain. "Pervasive services" are described as *essential services needed by most applications*, this could for instance be *essential components needed by most risk analysis methodologies* in AMBRA. "Domain specifications" are *standardization of services and facilities in specific markets,* a possible adaptation could be *standardization of components and facilities in specific risk analysis domains.* "Transparencies" and "Quality of service" are *models of environments with specific hardware and software attributes",* since the intended risk analysis domain for AMBRA is IT, we would not need to change these categories. The different components would be stored in the library for use of the developers.

7.9 What about the use of UML?

What if AMBRA was expressed in a different notation than UML? Most probably this would have worked quite well, provided that the notation was capable of correctly describing the terminology and the relationship between the terms. In our opinion UML was the most natural choice of modelling language to use for our purpose. The 'Class diagram' of UML is perfect for our need. It lets us specify each term as a separate class and provides several different kinds of predefined relationships.

8 Conclusions

This Chapter summarizes the main conclusions of this thesis. It specifies the main contibutions and further work. It also presents related work.

8.1 Main contributions

The main contribution of this thesis is the adaptation, extension and integration of terminology from model-based risk analysis in IEEE-1471's terminology for architectural description.

Based on this foundation, we have:

- Outlined an architectural framework for model-based risk analysis called AMBRA;
- Situated our framework in the context of MDA, providing a basis for tight integration of risk analysis in system development methodology;
- Formally defined the terminology part of the framework in UML;
- Specified our expectations to the architectural framework in the form of concrete success criteria, and carefully argued that they have been fulfilled.

Further more we have positioned our work in the architectural hierarchy, explained the relationship to other architectural frameworks, and carefully described the most important design decisions made during this work.

8.2 Further work

AMBRA is not yet a complete framework. The guideline and the library of reusable elements are still to be developed. As discussed in the previous Chapter, the library helps the analyst by providing different types of general components suitable for reuse in the new analysis, while the guideline includes methods for compliance checking, conformance checking and instantiation of library elements.

In order to strengthen the evidence for the suitability of AMBRA there is need to evaluate AMBRA with focus on realism, e.g. a field study. We propose for further work to conduct two field studies: one where a methodology is developed following AMBRA and another one where this methodology is tested. These will provide valuable information on how useful AMBRA is with respect to all relevant user groups (direct users, indirect users I and indirect users II) as defined in Section 3.1.4.

8.3 Related work

AMBRA is a specialization of IEEE-1471, integrating terminology from model-based risk analysis in a way that is compliant with MDA. As far as we know there has not been made similar work. There are however other architectural frameworks as explained in Chapter 2. We refer to Section 2.2 for a characterization of the relationships with IEEE-1471 (described in Section 2.3) and MDA (described in Section 2.5).

For further information about the CORAS approach we refer to [Braber et.al. 03], [Dimitrakos et.al. 02] and [Aagedal et.al. 02]. There exist other approaches to model-based risk analysis, we believe the most interesting are CRAMM [CRAMM], SA [SA], RSDS [RSDS] and ATAM [ATAM]. Other more conventional approaches to risk analysis include HazOp, FMEA, FTA and Markov analysis.

References

Aagedal et.al. 02	Aagedal, J. Ø., den Braber, F., Dimitrakos, T., Gran, B. A., Raptis, D., Stølen, K.: Model-based risk assessment to improve enterprise security. In: Proc. EDOC2002. IEEE Computer Society (2002) 51–62
AS/NZS4360	AS/NZS 4369:1999 Risk Management, Australian Standard for Risk Management.
ATAM	Clements, P., Kazman, R., Klein, M.: Evaluating software architectures: methods and case studies. Addison-Wesley (2002)]
Atkinson et.al. 02	Colin Atkinson, Joachim Bayer, Christian Bunse, Erik Kamsties, Oliver Laitenberger, Roland Laqua, Dirk Muthig, Barbara Paech, Jürgen Wüst, and Jörg Zettel. Component-based Product Line Engineering with UML. Component Software Series. Addison-Wesley, 2002.
Bayer et.al. 01	Joachim Bayer, Dirk Muthig, and Brigitte Göpfert. The Library Systems Product Line. A KobrA Case Study. IESE-Report 024.01/E, Fraunhofer Institut Experimentelles Software Engineering, November 2001.
Braber et.al. 03	Folker den Braber, Theo Dimitrakos, Bjørn Axel Gran, Mass Soldal Lund, Ketil Stølen, Jan Øyvind Aagedal. The CORAS methodology: model-based risk management using UML and UP. Chapter in book titled UML and the Unified Process. Liliana Favre (ed), pages 332-357, IRM Press, 2003.
C4ISR-AF	Architectures Working Group: *Command,Control, Communication, Computer, Information, Surveillance, Reconnaissance – Architecture Framework (C4ISR-AF),* version 2.0, 18 Dec. 1997. www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/
CRAMM	Barber, B., Davey, J.: The use of the CCTA risk analysis and management methodology CRAMM. In: Proc. MEDINF. North Holland (1992) 1589–1593
Dimitrakos et.al. 02	Dimitrakos, T., Ritchie, B., Raptis, D., Aagedal, J.Ø., den Braber, F., Stølen, K.,Houmb, S.-H.: Integrating model-based security risk management into eBusiness systems development: The CORAS approach. In: Proc. I3E2002. Kluwer (2002) 159–175
FMEA	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.
FTA	IEC 1025: 1990 Fault tree analysis (FTA).
HazOp	Redmill, F., Chudleigh, M., Catmur, J. Hazop and software hazop. Wiley, 1999.
ieee.org	Official website of IEEE (Institute of Electrical and Electronics Engineers, Inc.) “The IEEE promotes the engineering process of

	creating, developing, integrating, sharing, and applying knowledge about electro and information technologies and sciences for the benefit of humanity and the profession." Retrieved 06.06.2003 http://www.ieee.org
IEEE1471	"Recommended Practice for Architectural Description of Software-intensive Systems" (IEEE1471-2000 Std), IEEE, 2000.
ISO10746	ISO/IEC 10746: 1995 Basic reference model for distributed processing.
ISO13335	ISO/IEC TR 13335-1:2001 *Information technology - Guidelines for the management of IT Security–Part 1: Concets and models for IT Security*
ISO17799	ISO/IEC 17799:2000 *Information thechnology – Code of practice for information security management*.
Lund et.al. 03	Mass Soldal Lund, Folker den Braber, Ketil Stølen. Maintaining results from security assessments. In Proc. 7th European Conference on Software Maintenance and Reengineering (CSMR'2003), pages 341-350, IEEE Computer Society, 2003.
Markov	Littlewood, B. A reliability model for systems with Markov structure. Journal of the Royal Statistical Society Series CApplied Statistics, 24(2):172-177. 1975.
McGrath 84	J. E. McGrath "Groups: interaction and performance" Prentice Hall, 1984 ISBN: 0-13-365700-0
MDA01	Model Driven Architecture (MDA). OMG document ormsc/01-07-01, July 2001.
MDA02	Text for an MDA guide. OMG document ormsc/02-10-01, October 2002.
meriamwebster.com	Online and free versions of the Merriam-Webster's Collegiate Thesaurus and Merriam-Webster's Collegiate Dictionary. Retrieved 06.06.2003 http://www.meriamwebster.com/
nr.no/coras	The official CORAS IST-2000-25031 website with information about the project. Retrieved 06.06.2003 http://www.nr.no/coras/
omg.org	General information about Object Management Group (OMG) from their official website. Retrieved 06.06.2003 http://www.omg.org/gettingstarted/gettingstartedindex.htm
omg.org/mda	General information about MDA published by OMG at their official website. Retrieved 06.06.2003 http://www.omg.org/mda/
Popper	Stanford Encyclopedia of Philosophy, information about the

	philosopher Karl Popper who has formulated the principle about falsification. Retrieved 06.06.2003. http://plato.stanford.edu/entries/popper/
Putman01	Janis R. Putman “Architecting with RM-ODP” Prentice Hall PTR, 2001. ISBN: 0130191167
Put01 p734	Janis R. Putman “Architecting with RM-ODP”, p 734: “18.2.4.6 How RM-ODP Enhances IEEE-1471 Architecture Description” . Prentice Hall PTR, 2001. ISBN: 0130191167
Put01 p737	Janis R. Putman “Architecting with RM-ODP”, p 737: “18.2.4.6 How RM-ODP Enhances IEEE-1471 Architecture Description” . Prentice Hall PTR, 2001. ISBN: 0130191167
Rausand 91	Marvind Rausand, ”Risikoanalyse. Veiledning til NS 5814” (norwegian), Tapir trykkeri, 1991. ISBN: 82-519-0970-8
RSDS	Lano, K., Androutsopoulos, K., Clark, D.: Structuring and design of reactive systemsusing RSDS and B. In: Proc. FASE 2000, LNCS 1783. Springer (2000) 97–111
Runde,Stølen 03	Ragnhild Kobro Runde, Ketil Stølen, “What is Model Driven Architecture?” University of Oslo, Department of Informatics, March 2003 ISBN: 82-7368-256-0, Research report 304
SA	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)]
SESC	General information about the committee, from the official website for Software Engineering Standards Committee (SESC). Retrieved 06.06.2003 http://standards.computer.org/sesc/sesc_geninfo/infondex.htm
Sommerville 95	Ian Sommerville “Software Engineering”, fifth edition Addison-Wesley, 1995. ISBN: 0-201-42765-6
TOGAF	Website aboute The Open Group Architectural Framework (TOGAF) Version 7. Retrieved 06.06.2003 www.opengroup.org/togaf
UML	James Rumbaugh, Ivar Jacobson, Grady Booch, “The Unified Modeling Reference Manual”, PEARSON PROFESSIONAL EDUCATION, 1999. ISBN: 0-201-30998-X

Appendix

A. Questionnaire for Judgment study

Purpose: to verify the criteria listed in Chapter 3 "Problem analysis", especially the one which says that AMBRA should be useful from a practical point of view.
In order to validate AMBRA we will use skilled people with knowledge about the background of AMBRA to evaluate the framework and give their judgment.

Organization of study

Our study will be based the response from five judges. They will be given a draft of the thesis about AMBRA and asked to answer a questionnaire about the framework.

Questionnaire

These are the questions posed to the judges:

1) Do you know if similar or identical framework been made? Y/N
2) Is the technical depth appropriate? Y/N (If no, please comment.)
3) Has any vital material been omitted? Y/N (If yes, please comment.)
4) Are there any serious flaws in the work? Y/N (If yes, please comment.)
5) Does the title "AMBRA –Architectural framework for Model-based Risk Analysis" accurately reflect the content of the framework? Y/N (If no, please comment.)
6) Is the length of the framework and the thesis justified by the content? Y/N (If no, please comment.)
7) Are there any figures/tables that can be omitted or need to be changed? Y/N (If yes, please comment.)
8) Is the English easily understandable? (grade 1-6, 1=poor, 6= excellent)
9) Do you think AMBRA is suitable as a reference framework in order to ensure that a new method for risk analysis incorporates important elements and their relationships? (grade 1-6, 1=poor, 6= excellent)
10) Do you agree in these 'intended users' of AMBRA?: Y/N (If no, please comment.)
    a. Users whose purpose is to develop methodologies for risk analysis (direct users)
    b. Regular participants in a risk analysis developed according to AMBRA like target stakeholders and the risk analyst(s) (indirect users).
11) Do you have any general comments and recommendations, please give them to me:

B. RM-ODP viewpoints

The five viewpoints in RM-ODP are defined as follows:
1. "The **enterprise** viewpoint, which examines the system and its environment in the context of the business requirements on the system, its purpose, scope and policies. It deals with aspects of the enterprise such as its organizational structure, which affect the system."

2. "The **information** viewpoint, which focuses on the information in the system. How the information is structured, how it changes, information flows, and the logical divisions between independent functions within the system are all dealt with in the information viewpoint."

3. "The **computational** viewpoint, which focuses on functional decomposition of the system into objects which interact at interfaces."

4. “The **engineering** viewpoint, which focuses on how distributed interaction between system objects is supported.”

5. “The **technology** viewpoint, which concentrates on the individual hardware and software components which make up the system.”
[ISO10746]

C. About the Object Management Group™ (OMG™)

This is what OMG present themselves as on their website [OMG]: “The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. Our membership includes virtually every large company in the computer industry, and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing today are represented on our Board of Directors.
Our flagship specification is the multi-platform Model Driven Architecture (MDA), recently underway but already well known in the industry. It is based on the modeling specifications the MOF, the UML, XMI, and CWM. OMG's own middleware platform is CORBA, which includes the Interface Definition Language OMG IDL, and protocol IIOP. The Object Management Architecture (OMA) defines standard services that will carry over into MDA work shortly. OMG Task Forces standardize Domain Facilities in industries such as healthcare, manufacturing, telecommunications, and others.”

Members of OMG:

AT&T	Glaxo	Microsoft	Rational
BEA	SmithKline	MITRE	SAGA Software
Borland	Hewlett Packard	MSC.Software	SAP
Boeing	Hitachi	NASA	SAS Institute
CA	Hyperion	NEC	Secant
Citigroup	IBM	NetGenics	Siemens
Compaq	IONA	NTT	Sprint
Ericsson	io Software	OASIS	Sun
Ford	Kabira	Oracle	Unisys
Fujitsu	Kennedy Carter	Pfizer	Vertel
	John Deere		