MACCIS 2.0 – An Architecture Description Framework for Technical Infostructures and their Enterprise Environment

**Brian Elvesæter[1], Tor Neple[1], Jan Øyvind Aagedal[1], Rolf Kenneth Rolfsen[1],
Ole-Øyvind Stensli[2]**

[1]SINTEF ICT, P.O. Box 124 Blindern, N-0314 Oslo, Norway
Tel: +47 22 06 73 00, fax: +47 22 06 73 50
{brian.elvesater | tor.neple | jan.aagedal | rolf.k.rolfsen}@sintef.no

[2]Norwegian Defence Logistics Organisation, P.O. Box 43, N-1306 Bærum, Norway
Tel: +47 67 86 30 00, fax: +47 67 86 30 79
oostensli@mil.no

MACCIS 2.0 – An Architecture Description Framework for Technical Infostructures and their Enterprise Environment

Brian Elvesæter[①], Tor Neple[①], Jan Øyvind Aagedal[①], Rolf-Kenneth Rolfsen[①], Ole-Øyvind Stensli[②]

[①]SINTEF ICT, P.O. Box 124 Blindern, N-0314 Oslo, Norway
Tel: +47 22 06 73 00, fax: +47 22 06 73 50
{brian.elvesater | tor.neple | jan.aagedal | rolf.k.rolfsen}@sintef.no

[②]Norwegian Defence Logistics Organisation, P.O. Box 43, N-1306 Bærum, Norway
Tel: +47 67 86 30 00, fax: +47 67 86 30 79
oostensli@mil.no

Abstract

This paper presents the MACCIS 2.0 framework. MACCIS is an architecture description framework that defines architectural artefacts that are used to describe technical information infrastructures (infostructures) and their enterprise environment. The MACCIS framework has been developed for the Norwegian Defence Logistics Organisation (NDLO) and its intended use is to describe the architectures of C2IS systems and their C2 environments. MACCIS 2.0 is a framework that is built on standards and experiences from more than five years of framework development and use. The framework contains two parts with different foci; MACCIS 2.0 Infostructure Edition (M2IE) is concerned with descriptions of technical infostructures, while MACCIS 2.0 Enterprise Edition (M2EE) is concerned with descriptions of the enterprise environment of the infostructures. In addition to identifying architectural artefacts, MACCIS also contains guidelines for how to produce and document the architectural artefacts. The UML modelling language with standard extensions mechanisms is used to define the notation for the architecture descriptions. This paper also summarises some of the experiences that have been made using MACCIS in projects within the NDLO.

1. Introduction

The last years defence organisations around the world have increasingly been faced with expectations to change. There are different drivers for these changes; the tasks that armed forces are set to carry out have changed to become more adaptive and defence budgets have been cut while the performance requirements are upheld. Due to this, focus has been set on creating highly effective and flexible armed force units.

1.1 Problem area

Information technologies are important to facilitate the required changes. There is a need for flexible systems that can support the new and flexible modes of operation. The current view is that component-based information systems can be used to support component-oriented organisations, meaning organisations that can be assembled from standard organisational components to meet specific tasks. To ensure consistency

between an organisation and its information systems, knowledge of information systems and organisational architectures is necessary. Such knowledge can be obtained by the creation of architectural descriptions, and we believe such descriptions should be created based on common guidelines. This is important for several reasons; one ensures that the descriptions contain essential information, the descriptions become comparable, advice on creating such descriptions can be given, and common architectural patterns can be proposed.

Architecture descriptions provide a valuable tool for analysing and understanding Command and Control (C2) and Command and Control Information Systems (C2IS) integration. This is achieved through description of "component" capabilities and how components work together to form a coherent whole. In order to be able to maximise the potential effects of architecture descriptions, it is necessary to have frameworks, tools and procedures to create and maintain such artefacts in a standardised manner across different parts of the defence organisation.

1.2 MACCIS 2.0

The Norwegian armed forces have been faced with the same challenges as mentioned above. In order to handle these challenges, projects have been initiated by the Norwegian Defence Logistics Organisation (NDLO) to develop a framework for architecture descriptions with tool support. This initiative has been going on since 1998 and the architecture description framework, MACCIS, is now in its second generation.

MACCIS of today (version 2.0) consists of two parts; MACCIS 2.0 Infostructure Edition (M2IE) is concerned with descriptions of technical infostructures, while MACCIS 2.0 Enterprise Edition (M2EE) is concerned with descriptions of the enterprise environment of the infostructures. Figure 1 summarises the history of the MACCIS framework along a timeline.
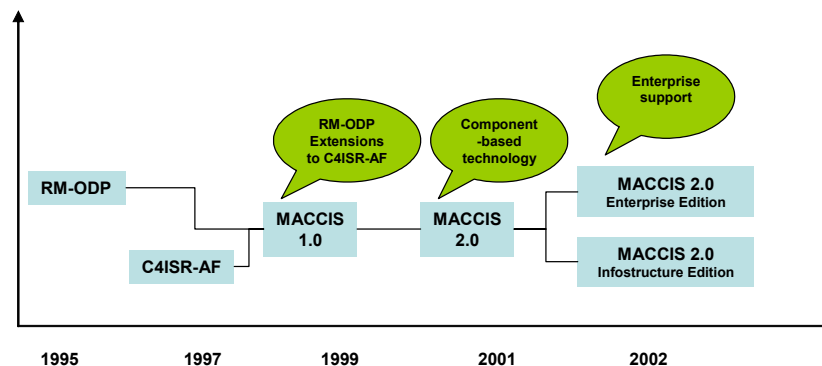

Figure 1. MACCIS development history

In 1999, MACCIS 1.0 [1] was developed, targeted specifically towards the description of Command and Control Information Systems (C2IS). The first version of MACCIS extended the C4ISR Architecture Framework [2] with elements from RM-ODP [3-6]. In the period from 1999 to 2002, the framework was extended with component-oriented concepts to handle both technical information infrastructures (infostructures) and the enterprises they support. The evolution of MACCIS has been based on feedback and experiences from NDLO that have used the framework to address different aspects of C2IS and C2 integration. MACCIS 2.0 was first

introduced in 2001 [7, 8] as an update to the original MACCIS 1.0 framework. When M2EE was introduced in 2002 [9], the MACCIS 2.0 of 2001 was revised and renamed M2IE. In this paper we use the term MACCIS 2.0 to refer to the latest versions of the framework, consisting of both the infostructure edition and the enterprise edition.

1.3   *Framework objectives*

When creating something it is usually done for a purpose, or an objective, so also for MACCIS. At a high level the objective is clear; create a framework that allows users in different communities to create architecture descriptions using a common terminology and set of concepts. This high level objective hides a set of more detailed goals. Since MACCIS is a result of evolution not all goals were present at the outset, some needs did arise based on experience in use of the framework. The key goals are summarised in this chapter and will be used as points for discussion and measurement in the closing parts of this paper.

1. The framework should provide means to describe the same as C4ISR AF (and later DoD AF [10-12]), but should add concepts to specifically describe distributed and component-based systems. Civilian and military standards should be used as the basis whenever possible.
2. The Unified Modeling Language (UML) should be used as the language for notation, in addition to structured text. Where more expression power is needed one should opt to utilise the standard extension mechanisms provided by UML.
3. The framework should be simple enough to actually be used, yet have enough expression power to be useful for describing C2IS and C2 architectures.
4. The framework should support both the process of defining and procuring a new system, and maintaining and extending existing systems.
5. The main focus of the framework is to be description of the architectures of information infrastructures and their enterprise environments.
6. The main problem domain is to be Command and Control, Communications, Computers and Intelligence (C4I).

1.4   *Structure of paper*

This paper is structured as follows: Section 2 presents an overview of the MACCIS framework, its background and history, the key concepts used and the components of the framework. Section 3 describes the application of the parts of the architecture description framework. Section 4 summarises some of the experiences gained when using the MACCIS framework. Section 5 presents the conclusions and discusses plans for future work. The last two sections contain acknowledgements and a list of references.

2.   Overview of the MACCIS framework

MACCIS is an architecture description framework targeted towards describing the architecture of technical information infrastructures and their enterprise environment. An architecture description framework defines how architectures should be described. In addition to identifying architectural artefacts, MACCIS also contains guidelines for how to produce and document the architectural artefacts. Figure 2 depicts the foundation on which MACCIS 2.0 was developed and its main components.
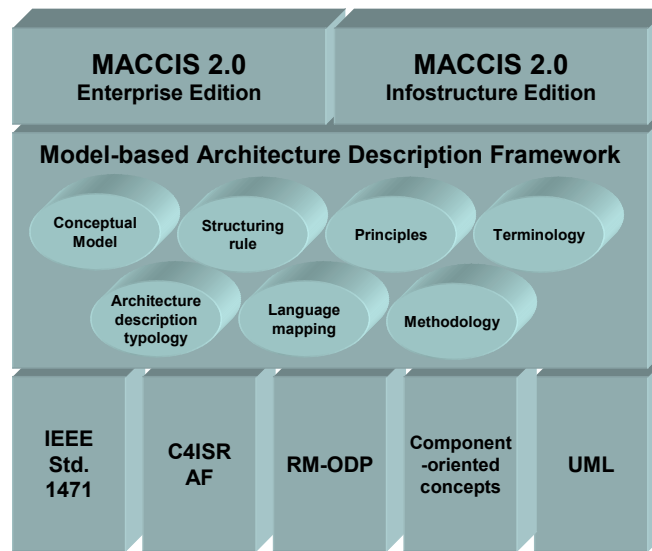
Figure 2. MACCIS 2.0 foundation and contents

The MACCIS framework contains two parts; MACCIS 2.0 Enterprise Edition (M2EE) which is a framework for describing enterprise architectures, and MACCIS 2.0 Infostructure Edition (M2IE) which is a framework for describing technical information infrastructures. The term *infostructure* is used as an abbreviation of the term *information infrastructure*.

The M2EE and M2IE editions of MACCIS define a model-based architecture description framework that contains

- a *conceptual model* of architectural descriptions;
- a set of *structuring rules*, specifying which views and models that should be included in an architectural description;
- *principles* of conformance, consistency, specialisation and realisation;
- a set of *language mappings,* i.e. mappings between the conceptual model and the modelling languages to be used to create architecture descriptions;
- a set of *architecture description typologies*, according to which architecture descriptions may be characterised, i.e. "as-is" or "to-be";
- a *methodology*, specifying how to go about developing architecture descriptions;

for describing enterprise and infostructure architectures, respectively.

2.1 MACCIS Foundation

MACCIS builds upon a well-based foundation of standards and related architecture frameworks. MACCIS aims to tie the results together in a useful and useable framework, and adds domain-specific architecture extensions based on research projects and industry experiences.

2.1.1 IEEE Std. 1471

The recommendations of IEEE 1471 [13] include definitions of important concepts and relate these in a conceptual model of architectural description. The terminology

defined in IEEE 1471 is adopted by the MACCIS framework. Some of the most important terms and their definitions are given below:

- *Architecture*: The fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.
- *Architectural description*: A collection of products to document an architecture.
- *Concern*: Those interests to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders.
- *Model*: A representation of an entity in the real world.
- *View*: A representation of the whole system from the perspective of a related set of concerns.
- *Viewpoint*: 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.
- *Stakeholder*: An individual, team, or organisation (or classes thereof) with interests in, or concerns relative to, a system.
- *System*: A collection of components organised to accomplish a specific function or set of functions.

According to the IEEE conceptual model, any system of interest has an architecture that can be described by an architectural description. The structuring rules defined in MACCIS for architectural descriptions have been based on the IEEE 1471 standard. In MACCIS the system represents either an enterprise or an infostructure.

2.1.2   C4ISR AF

The C4ISR AF [2] defines three different architecture views that correspond to viewpoints in the MACCIS terminology: *operational architecture view* (focuses on the operational aspects of a military operation), *systems architecture view* (focuses on the infostructure warfighting functions), and *technical architecture view* (focuses on technical standards and conventions).

Each of these viewpoints defines a set of architecture products. These products are catered for in MACCIS by different models that describe different parts of an enterprise or an infostructure. In addition, MACCIS specifies additional models that describe other missing enterprise and infostructure aspects of importance to stakeholders that have been identified based on framework use within the NDLO.

2.1.3   RM-ODP

The Reference Model for Open Distributed Processing (RM-ODP) [3-6] is an ISO standard focusing on open distributed processing systems. RM-ODP divides the specification of ODP systems into five different, but related, viewpoints. The viewpoints in RM-ODP are: *enterprise viewpoint* (focuses on purpose, scope and policies), *information viewpoint* (focuses on information processing and relationships between information objects), *computational viewpoint* (focuses on functional specification and decomposition), *engineering viewpoint* (focuses on how to solve distribution issues), and *technology viewpoint* (focuses on specific technology and solutions).

The MACCIS principles of conformance and consistency are based heavily on RM-ODP. In MACCIS, as in RM-ODP, viewpoints are based on a common foundation, which in addition to a core set of modelling concepts, provides a set of viewpoint consistency rules based on correspondences between the core modelling concepts of the different viewpoints. One is therefore able to correlate different formulations of the same or related abstract concepts in different views of the system.

2.1.4 Component-oriented concepts

Component-orientation is a way of thinking of systems that has evolved from the object-oriented paradigm. A component is an autonomous unit that has specific properties and provides a specific service or function to the environment in which it is located. In MACCIS we have adopted component-oriented concepts both for describing enterprises and infostructures.
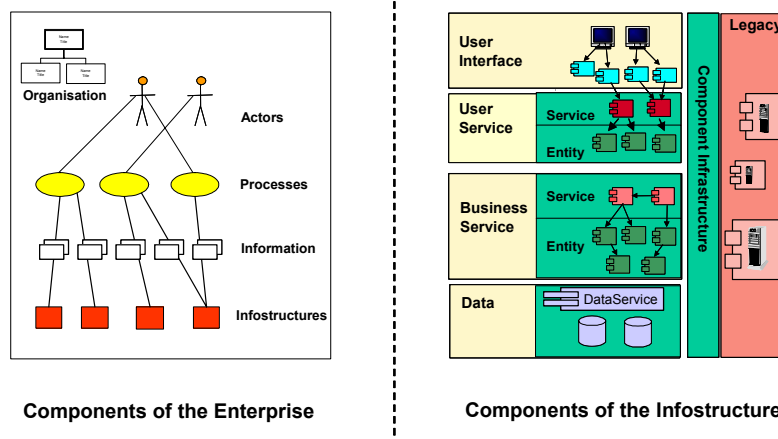

Figure 3. Component-orientation applied to Enterprise and Infostructure

Applying component-orientation to enterprises, we find that the constituent parts that make up an enterprise consist of amongst other things organisational units, process units, information units and infostructures as illustrated in Figure 3. In the military domain, C2 in particular, the concepts of components can be applied to describe task organisations in which organisational units, e.g. infantries or artilleries, can be organised to meet specific tasks. The paper [14] describes one approach to business modelling with components that has been adopted by the M2EE framework.

Component-orientation is a well-disciplined art in the software and hardware industry. MACCIS has defined a 4-tier reference model that can be used in order to describe the architecture of component-based information systems. The reference model shown in Figure 3 defines a set of logical tiers, each of which consists of a set of components. The four tiers are: *User Interface tier* (focuses on interaction logic through a set of user interface components), *User Service tier* (focuses on user session logic through a set of user service and entity components), *Business Service tier* (focuses on business functionality through a set of business service and entity components), and *Data tier* (provides data service components for accessing and updating data resources).

Information systems developed using modern component-based technologies such as J2EE, WebServices, .Net and CORBA are deployed in a *component infrastructure* that uses an underlying communication infrastructure. The component infrastructure is also used to integrate legacy systems.

2.1.5 UML

MACCIS 2.0 prescribes the use of the Unified Modeling Language (UML) as the primary notation for documenting models that describe the architecture of an enterprise or an infostructure. The UML language subset that is used for MACCIS is based on the 1.4 specification of UML [15] which provides modelling constructs to develop and document component-based architectures. Since UML is a widely used standard for developing models, there is a wide support in tools and technologies.

Standard UML extension mechanisms can be used to extend UML with new concepts and notations. MACCIS provides mappings from the conceptual models for architecture descriptions defined in M2EE and M2IE to the UML language. In addition to MACCIS-specific mappings, OMG standards such as the "UML Profile for Enterprise Distributed Object Computing" [16] for describing enterprise components, and working standards such as the "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms" [17] for describing quality of service and risk assessment, have been adopted by the MACCIS framework.

2.2 Model-based Architecture Description Framework

The target of consideration during modelling is often referred to as "system". "System" is used to denote entities on many levels, from virtual enterprises where clusters of businesses compose the "system" down to software objects with data attributes and operations on these constitute the "system". These system levels are related. A virtual enterprise consists of interacting businesses. A business consists of interacting actors and technical infostructures. A technical infostructure can be viewed in terms of technical components. Furthermore, one can have an arbitrary number of levels by using recursion. For instance, a business may consist of businesses or a technical infostructure may consist of technical infostructures. Figure 4 shows the common system levels. The recursion is illustrated by the curled arrows pointing back to the same level.
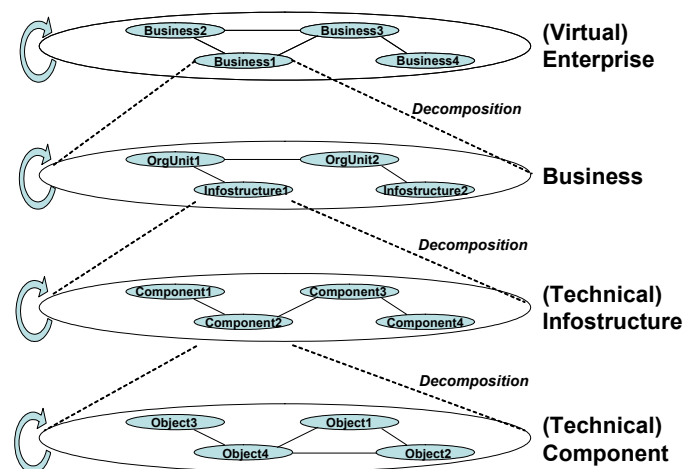

Figure 4. System levels

A model should either be a model of entities on one such level and how these interact or are related, or a model of the relationship between two adjacent levels. The MACCIS framework provides constructs and guidelines to manage the relationships

between these system levels. Enterprise architecture descriptions, defined by M2EE, focuses on the virtual enterprise and business levels shown in Figure 4, and infostructure architecture descriptions, defined by M2IE, focuses on the technical infostructure and component levels. The two architecture descriptions are tied together on the business level, which defines the context of the technical infostructure. This is further illustrated in Figure 5 which shows the relationships between the enterprise architecture description and the infostructure architecture description, and how these reflect entities in the real world.
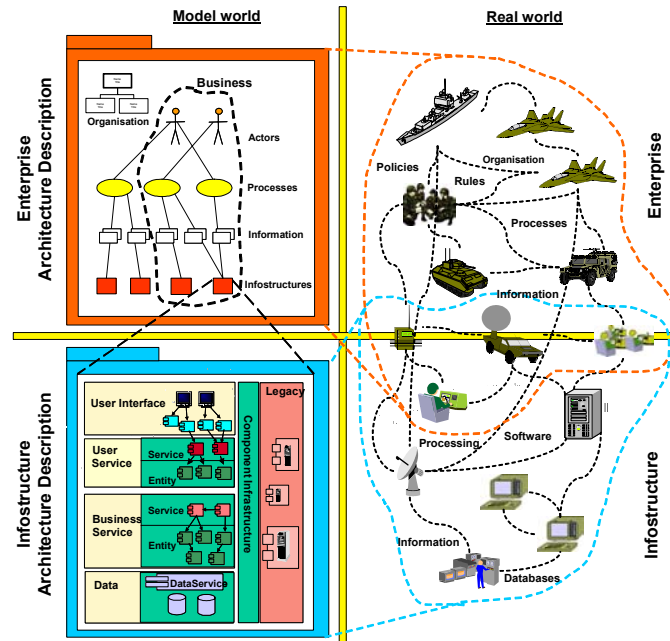

Figure 5. Architecture descriptions

An enterprise architecture description is formalised as an enterprise model that highlights the essential characteristics of the real world enterprise, while an infostructure architecture description is formalised as an infostructure model that captures the essential aspects of the technical infostructures in the real world. The real world consists of both non-physical and physical entities. An enterprise or infostructure model describes both non-physical entities (e.g. organisation, processes and software) and physical entities (e.g. vessels, terminals, and descriptions of both non-physical and physical entities) and relates these entities together in a model that one can analyse in order to better understand and evolve the real world. As pointed out in Figure 4, the enterprise and infostructure levels are related to each other through the business level. The business level is maintained in the MACCIS framework by links between the enterprise models and the infostructure models. The links define how the businesses of the enterprise are supported by infostructures (top-down view) and the business context of the infostructure (bottom-up view).

2.2.1   Model structure

The MACCIS framework provides guidelines for how to develop enterprise and infostructure architecture descriptions in a coherent way. The principles of MACCIS ensure that all models that are part of the architecture description must be related. This

ensures an integrated architecture description where changes and decisions made in one description propagates and shows up in another related description.

The different models are structured according to stakeholders, viewpoints, concerns and views as defined in IEEE Std 1471. A system on any level may be viewed from different viewpoints. The motivation for using viewpoints comes from the fact that a complete specification of any non-trivial system involves a very large amount of information. This information can be separated into views that address different areas of concern for different stakeholders. The number of viewpoints to use depends on the nature of the system and its stakeholders.
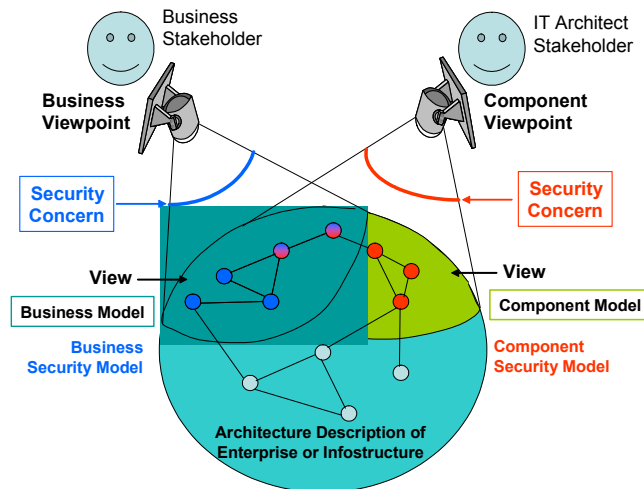

Figure 6. Stakeholders, Viewpoints, Concerns, Views and Models

Each viewpoint is an abstraction that yields a specification of the whole system. Viewpoint specifications may partly overlap and may also include different aspects of the same system component, making the consistency of the viewpoint specifications crucial. Figure 6 illustrates the viewpoint metaphor by showing how different aspects of a system are illuminated when seen from different viewpoints belonging to different stakeholders. A business stakeholder sees the system from a business viewpoint. The view that is illuminated through this viewpoint is described as a business model. An IT architect is interested in another view of the system, and uses a component viewpoint to see the component model of the system. Both stakeholders may be concerned about security, but want to see different descriptions of how the security is maintained in the system. The security concern, when applied to the different viewpoints, addresses both stakeholders, and is described as a business security model or component security model respectively.

The correspondence between viewpoints and concerns can be described in a matrix as shown in Figure 7. From the matrix we see that one specific view can be seen as a cross product of one specific viewpoint and a set of concerns; we can have views that span multiple concerns and filtered views that address one particular concern. Each view is described as a set of models according to the MACCIS framework.

Concerns

Concern1 Concern2 Concern... ConcernN

Viewpoints

Viewpoint1
Viewpoint2
Viewpoint...
ViewpointN

View
= Viewpoint2 x (Concern1...N)
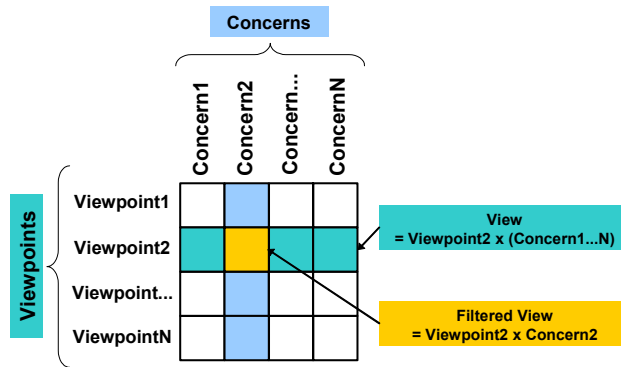
Filtered View
= Viewpoint2 x Concern2

Figure 7. Viewpoint x Concern matrix

The structuring rules of MACCIS allow for an infinite set of models to be created. In order to make the MACCIS framework usable, it provides a starting point with a finite set of defined viewpoints and concerns that can be applied. Each viewpoint defines a set of modelling constructs that can be used to describe the different concerns as formal models. These models can overlap in the same way as views overlap with respect to concerns. Furthermore, the MACCIS framework does not prescribe that all models should be developed every time; this depends on the stakeholders involved and the intended use of the models.

A more detailed description of the viewpoints and concerns defined in the MACCIS framework is given in the two following sections for the enterprise and the infostructure architecture description, respectively.

2.2.2 Enterprise Architecture Description

An enterprise architecture description is based on the following viewpoints and concerns defined in MACCIS 2.0 Enterprise Edition (M2EE).

- The *context viewpoint* focuses on the environment of the enterprise.
- The *enterprise viewpoint* focuses on the structure of the enterprise.
- The *realisation viewpoint* focuses on the technical information infostructure used or required by the enterprise.
- The *operational concern* specifies how to describe key information about how the enterprise works.
- The *distribution concern* specifies how to describe geographical locations and distributions.
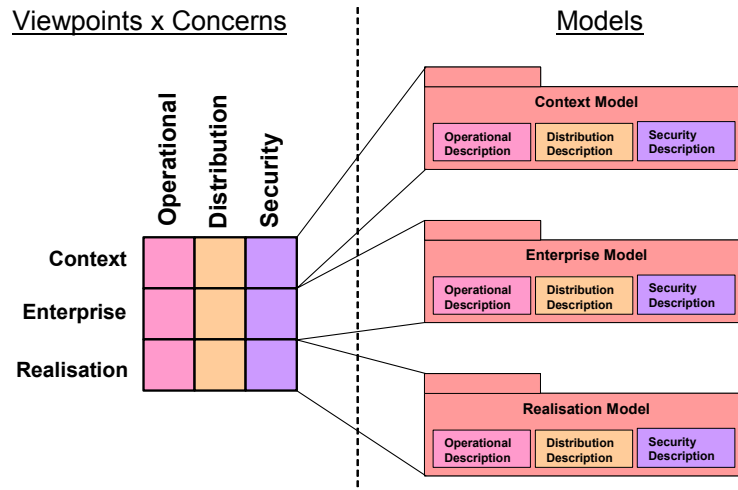- The *security concern* specifies how to describe risk and security issues.

Figure 8. Enterprise Viewpoints, Concerns and Models

Each of the viewpoints is described as a separate model in UML. Each model addresses all of the concerns defined as shown in Figure 8.

- **Context Model:** The context model describes the relevant stakeholders of the enterprise and their interests with respect to the operational, distribution and security concerns.
- **Enterprise Model:** The enterprise model describes how the enterprise works, the decomposition of the enterprise into components and how they are organised, with respect to the operational, distribution and security concerns.
- **Realisation Model:** The realisation model describes the technical infostructure with respect to the operational, distribution and security concerns.

2.2.3   *Infostructure Architecture Description*

An infostructure architecture description is based on the following viewpoints and concerns defined in MACCIS 2.0 Infostructure Edition (M2IE).

- The *business viewpoint* focuses on the context and purpose of the infostructure within the enterprise.
- The *requirements viewpoint* focuses on the business and user requirements of the infostructure.
- The *component viewpoint* focuses on the structure of the infostructure.
- The *platform viewpoint* focuses on the physical structure and technology used to implement the infostructure.
- The *functionality concern* specifies how to describe the services provided by the infostructure that are needed by the business.
- The *distribution concern* specifies how to describe logical and physical distributions of the infostructure.
- The *security concern* specifies how to describe security issues related to the infostructure.
- The *QoS concern* specifies how to describe quality of service requirements and fulfilment contracts for the infostructure.
- The *usability concern* specifies how to describe usability issues related to using the infostructure.
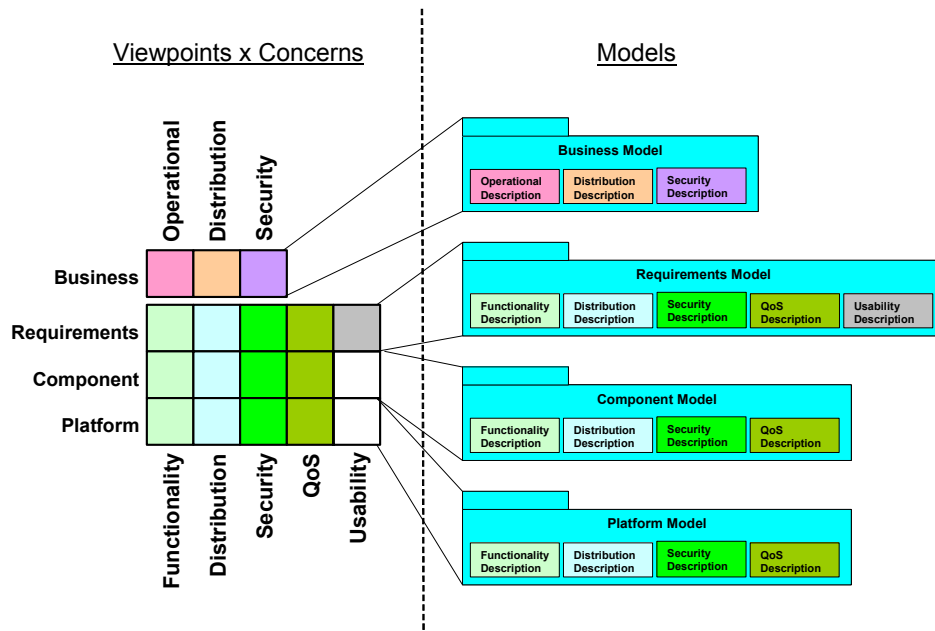
Figure 9. Infostructure Viewpoints, Concerns and Models

Each of the viewpoints is described as a separate model in UML. Each model addresses all of the concerns defined as shown in Figure 9.

- **Business Model:** The business model describes the business context of the enterprise that is relevant to the technical infostructure under consideration. This is typically a subset of models from the enterprise architecture description, where the models have been developed according to the operational, distribution and security concerns defined in M2EE. The business model defines the business requirements for the infostructure.
- **Requirements Model:** The requirements model describes the functional, distribution, security, quality of service and usability requirements for the infostructure. The initial requirements are derived from the business model, and further elaborated and refined.
- **Component Model:** The component model describes the infostructure in terms of its subsystems and information objects, and documents how subsystem interaction and information processing are carried out in order to provide desired effects.
- **Platform Model:** The platform model describes the realisation of the infostructure in terms of its implementation technology and its underlying communication infrastructures with respect to the functionality, distribution, security and QoS concerns.

3.   Applying the MACCIS framework

In order for the MACCIS 2.0 framework to be useful to the Norwegian Defence Logistics Organisation, we have developed a set of methodology handbooks and implemented MACCIS add-ins for standard UML tools used within NDLO. The add-ins defines model templates, visual enhancements, new functionality and consistency checks that allow military personnel to create architecture descriptions using standard UML tools. The methodology handbooks provide an architecture development

process which contains useful guidelines and techniques for how to develop the various models specified in the MACCIS framework.
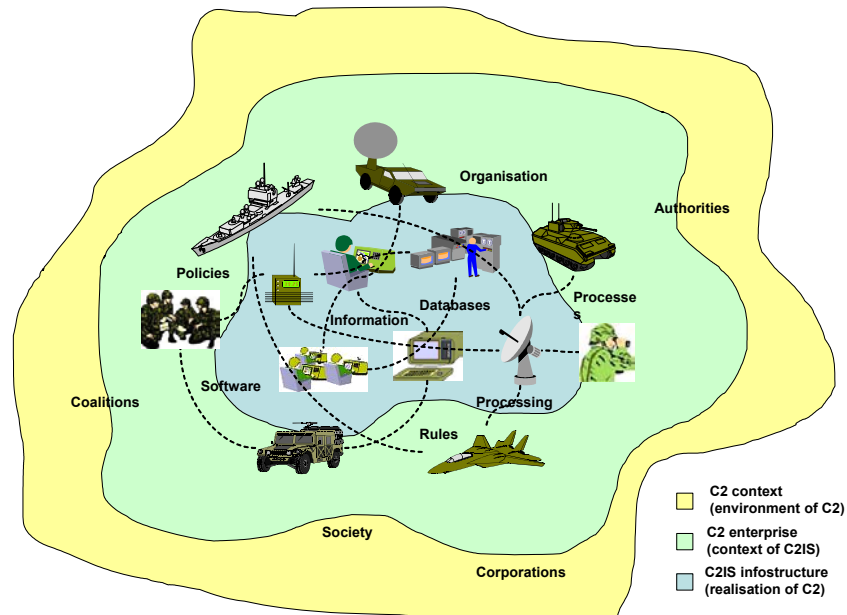

Figure 10. C2 and C2IS environment

Figure 10 depicts a C2 and C2IS environment of the real world which is used as an illustrative example on how to develop an architecture description. The C2IS infostructure operates in the context of a C2 enterprise, which in turn has its own environment denoted as the C2 context. The C2 enterprise is realised partly by the supporting C2IS infostructure.

When developing the architecture of a system, enterprise or infostructure, one must first identify the stakeholders and their concerns so that we focus on developing the relevant models. Figure 11 illustrates how two different enterprise stakeholders are concerned about separate issues in the real world that they would like to see described as models that can be used for analysis and planning purposes. It is important that the enterprise architecture description addresses all of the stakeholders and their concerns in order to be useful beyond a small segment of an organisation.
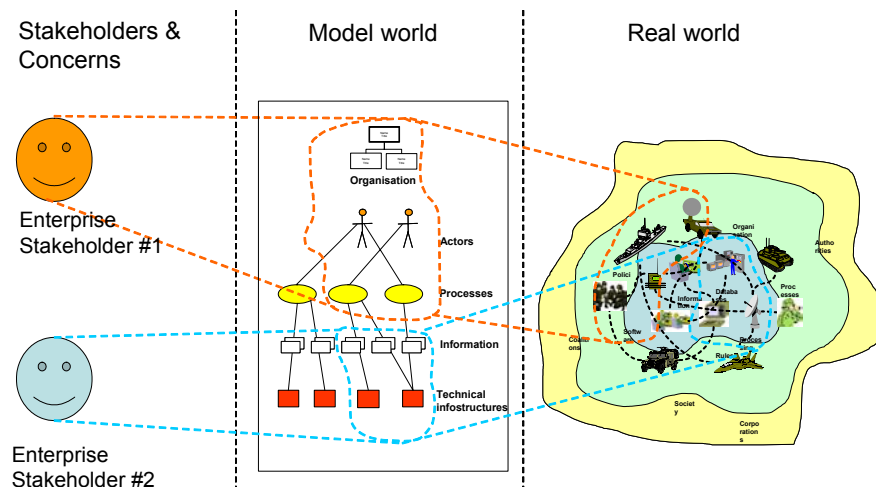

Figure 11. Defining the stakeholders and their concerns

Architecture descriptions are typically developed by trained architecture experts that have a good knowledge of the MACCIS framework and the tools being used. The Norwegian Defence Logistics Organisation has trained their own architecture group of experts that participate in workshops and interviews with various stakeholders such as domain experts, IT architects, user groups and industry suppliers when developing the models. The models are refined with more details over time and reviewed by the stakeholders in order to ensure that the models truly reflect the real world.

3.1 Enterprise Architecture Description

The MACCIS 2.0 Enterprise Edition (M2EE) defines a set of model artefacts that are used to describe the operational, distribution and security concerns. Each of the concerns is modelled according to viewpoint structuring rules. Based on the relevant stakeholders identified and their concerns, one can start to develop the appropriate sub-models shown in the table below. The table summarises the different sub-models that can be developed using the M2EE framework. The *concern column* indicates which concerns the different sub-models support: (O)perational, (D)istribution and (S)ecurity.

Model	Sub-model	Concern	Description
Context Model	Overview and Summary	O	The overview and summary model is an informal model that focuses on the purpose and scope of the enterprise, the actors and artefacts involved, and the operational concepts. This model may include descriptions of missions, products, society, authorities and corporations that are relevant to the enterprise.
	Distribution Context	D	The distribution context model describes the geographical locations and distribution of the enterprise.
	Security Context	S	The security context model describes security policies and regulations that enterprise must adhere to.
Enterprise Model	Vision	O	The purpose of this model artefact is to identify and describe the enterprise vision.
	Enterprise Goals	O	The purpose of this model is to describe a hierarchical goal structure that can be agreed upon with the stakeholders so that a set of required high-level processes can be identified for further analysis in the areas of responsibility.
	Organisation Structure	O, D	The purpose of this model is to describe the organisation of the enterprise, both formal and informal structures.
	Stakeholders & Concerns	O	The purpose of this model is to identify the stakeholders and their concerns for the enterprise in order to clarify the scope of the enterprise and possibly resolve potential conflicts of interest.
	Organisation roles & Responsibilities	O	The purpose of this model is to describe organisation roles and corresponding areas of responsibilities in order to relate enterprise activities, enterprise goals, organisation structures and enterprise resources.
	Enterprise Processes & Process Roles	O	The purpose of this model is to describe the enterprise processes and the corresponding process roles required to perform the activities of the processes.
	Enterprise Resources	O, D, S	The purpose of this model is to describe enterprise resources such as information and equipment that are needed by the enterprise processes. This model can be extended with distribution and security descriptions of the resources.
	Role-Distribution	O, D	The purpose of this model is to describe physical distribution of the different business roles, e.g. organisation roles.
	Role-Activity-Distribution	O, D	The purpose of this model is to describe the distribution units with its roles and the activities in each of the units.
	Activity-Distribution	O, D	The purpose of this model is to describe the distribution units and the activities from each of the units.
	Security	S	The purpose of this model is to describe security constraints related to how actors carry out their duties and how information is treated.

	Risk Assessment Model	S	The purpose of this model is to describe the strengths, weaknesses, opportunities, threats, unwanted incidents and risks related to the enterprise.
Realisation Model	Work Analysis Refinement Model	O	The purpose of this model is to refine the enterprise processes and the enterprise resources in order to identify the activities where infostructures are used and what information objects they manage.
	Infostructure Distribution Model	D	The purpose of this model is to describe the distribution of the information systems and the communication infrastructure of the enterprise.
	Security Threatment Model	S	The purpose of this model is to describe the security measures and treatments that must be provided by the infostructure.

The M2EE framework also provides visual extensions to the UML language for expressing enterprise models in a manner more recognisable to the military domain. The visual extensions provided are based on standardised UML extension mechanisms which can be implemented in most UML tools available. An example of such visual enhancements is given in the table below, where the APP-6A [18] notation can be used to model organisation structures. The table only shows a limited subset of the mapping.

Concept	Description	UML mapping	Icon
Artillery	Ground Track Unit Combat Field Artillery	Class <<OrgUnit_Artillery>>	
Support Electronic Warfare	Ground Track Unit Combat Support Military Intelligence Electronic Warfare	Class <<OrgUnit_SupportEW>>	EW
Support Explosive Ordnance Disposal	Ground Track Unit Combat Support Explosive Ordnance Disposal	Class <<OrgUnit_SupportEOD>>	EOD
Support Military Intelligence	Ground Track Unit Combat Support Military Intelligence	Class <<OrgUnit_SupportMI>>	MI
Infantry	Ground Track Unit Combat Infantry	Class <<OrgUnit_Infantry>>	
Support Information Warfare Unit	Ground Track Unit Combat Support Information Warfare Unit	Class <<OrgUnit_SupportIW>>	IW
Engineer	Ground Track Unit Combat Engineer	Class <<OrgUnit_Engineer>>	
Service Support Supply	Ground Track Unit Combat Service Support Supply	Class <<OrgUnit_ServiceSupply>>	
…	…	…	...

3.2   Infostructure Architecture Description

The table below summarises the different sub-models that can be developed using the M2IE framework. The *concern column* indicates which concerns the different sub-models support: (O)perational, (F)unctionality, (D)istribution, (S)ecurity, (Q)oS and (U)sability.

Model	Sub-model	Concern	Description
Business Model	Enterprise Processes, Enterprise Resources, Distribution, Security, Work Analysis	O, D, S	The business model contains the subset of model artefacts that are relevant to the infostructure we are describing the architecture of. The sub-models listed are those we have found relevant in most cases.

Requirements Model	Requirements	F, D, S, Q, U	The purpose of this model is to describe the system boundaries, the main actors and their responsibilities, and the main services offered by the system. Requirements related to distribution, security, QoS and usability can also be added to this model.
	User Interface Model	F, Q, U	The purpose of this model is to define the boundaries of the information system towards the user. This model can be extended with quality of service requirements put forward by the users and user interface sketches that addresses system usability.
Component Model	System Dependency Model	F, D	The purpose of this model is to describe how the system at hand fits into the set of existing systems that are currently in use.
	System Decomposition Model	F	The purpose of this model is to describe the system as divided into different subsystems or components, and how these are related to form a coherent whole.
	Interface Description Model	F, Q	The purpose of this model is to describe the interfaces of a component or subsystem in a manner that the software artefact can be understood and possibly reused. The interface descriptions can be annotated with QoS specifications.
	Structure Model	F, D, S	The purpose of this model is to specify relationships between information objects that must always be true (invariants). The structure model can also be used to describe the distribution and security classification of the information objects.
	Instance Model	F	The purpose of this model is to expresses assertions that must be true at a single point in time. Typically, an instance model is used to specify the states of information objects.
	Processing Model	F	The purpose of this model is to specify how the information can evolve as the system operates.
	System Security Model	S	The purpose of this model is to describe the different security mechanisms that are used in the system.
	Distribution Model	D, Q	The purpose of this model is to describe logical units consisting of a set of subsystems that must be distributed and deployed together. QoS requirements can be described for the communication links.
	Distribution Patterns	D	The purpose of this model is to describe general solutions to RM-ODP defined distribution transparencies or identified distribution concerns for the system.
	Distributed Component Profile	F, D	The purpose of this model is to describe the design specification for the implementation of the system components in a technology-independent manner.
Platform Model	Standards	F, D, S, Q, U	The purpose of this model is to provide a normative reference list of the standards being used.
	Deployment Model	D, Q	The purpose of this model is to describe the physical relationships among software and hardware components in the system.
	Architecture Extension Model	F, D, S	The purpose of this model is to document non-standard technology-specific extensions that are used.
	Technology Component Profile Model	F, D, S	The purpose of this model is to give a detailed view of the design and implementation of the system components in the chosen component technologies.
	Data Storage Model	F, D, S	The purpose of this model is to describe the implementation of the data model at a level of detail that makes it possible to maintain and change the system implementation as the system evolves over time.

4. Experiences with the MACCIS framework

As mentioned, MACCIS 2.0 is an evolution of MACCIS 1.0. This evolution has been based on experiences when using the MACCIS 1.0 framework through several projects the last years at the Norwegian Defence Logistics Organisation (NDLO). NDLO started using the M2IE part of MACCIS 2.0 in beginning 2001, and M2EE in

beginning 2003. The following sections summarises some experiences with the MACCIS framework.

4.1 Infostructure procurement projects at NDLO

The NDLO has been using the MACCIS framework in several C2IS procurement projects over a five year period. MACCIS establishes a set of models that are used in documenting the business context, requirements, components and platform realisation of a C2IS.
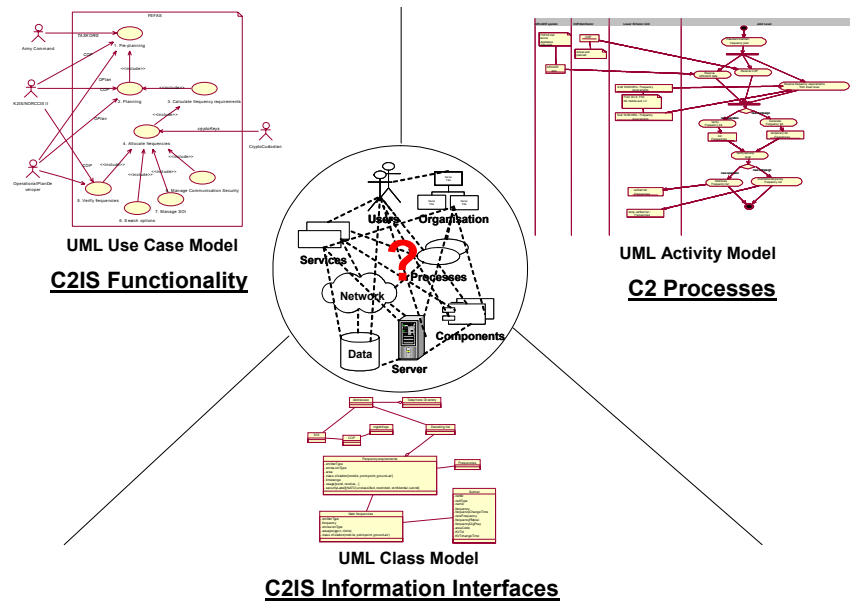

Figure 12. Important C2IS aspects are formalised as models in MACCIS

The MACCIS framework prescribes a set of essential models that amongst other things relate how C2IS functionality and C2IS information interfaces supports the C2 processes that are carried out in the enterprise. Figure 12 illustrates how these three aspects of a C2IS were documented as UML use case models, UML class models and UML activity models respectively. The activity models were used in the communication with the business experts to capture business requirements, while the use case models were used in the communication with the users in order to capture user requirements. The business and user requirements provided valuable input to the development of the information interfaces that the infostructures were to support.

The use of MACCIS has allowed the NDLO to better track the progress of the infostructures being acquired compared to earlier. The use of models has resulted in improved insight into the new C2IS systems being developed as well as existing C2IS systems, and has been a valuable tool in addressing how to integrate new and existing C2IS systems.

4.2 Logistics processes for electronic surveillance

The Enterprise Edition of the framework has been used to document the logistics processes *management*, *supply* and *operation* for electronic surveillance within the Norwegian Defence. The M2EE framework proved to be good tool for documenting and presenting the problem issue to all involved parties. It also proved to be a good

tool for describing a set of alternative solutions to the problem, which could be documented and presented in a standardised form.
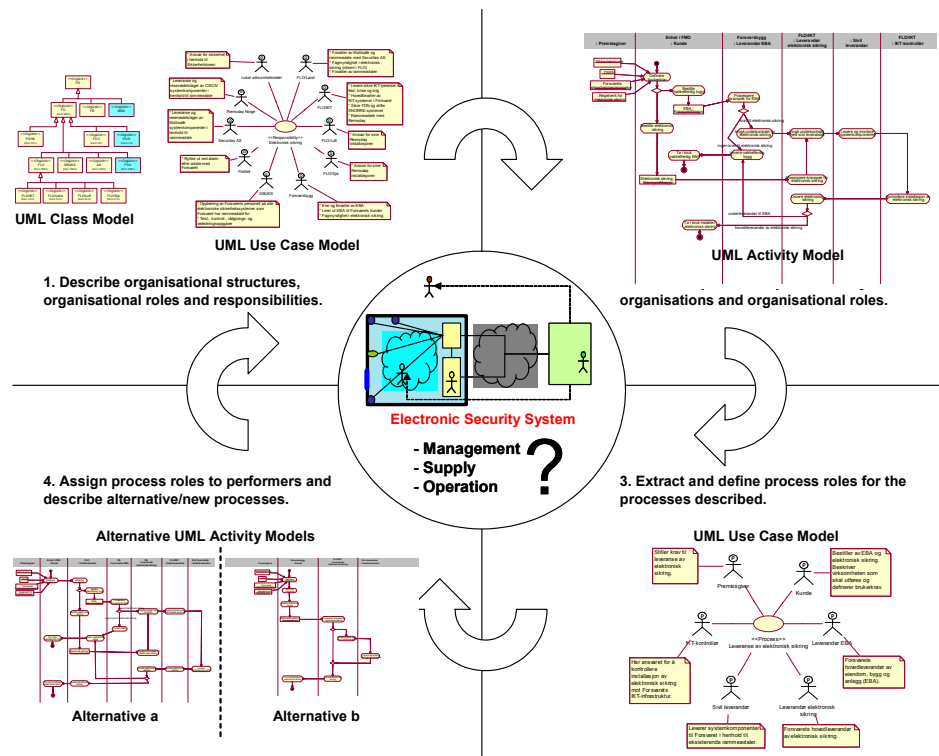

Figure 13. M2EE architecture description process

The M2EE architecture description process used to develop the models is illustrated in Figure 13.

1. An investigation of the current organisational roles and their responsibilities was carried out and presented using UML class and UML use case models.
2. More detailed models of the *management*, *supply* and *operation* processes were described using UML activity models. This provided a fairly perspicacious presentation of the current situation and also clearly pointed out the need for some changes.
3. Based on the "as-is" model that was developed in step 1 and 2, one could now start to identify organisational-independent roles for the processes in question. A selection of process roles were described for each process and presented as UML use case models.
4. In the subsequent discussions which took place one could use the identified process roles as a basis for assigning responsibilities to the most appropriate organisational unit. However, since no clear solution materialised in the discussions, the various alternative solutions were described as separate UML models.

4.3   Target architecture for Network Centric Warfare

MACCIS 2.0 has also been used in a project that set out to define a target architecture for Network Centric Warfare (NCW) for the Norwegian armed forces. The target architecture described a "to-be" model of how the Norwegian armed forces were to operate within the year 2008. In this project, MACCIS was used in combination with

the DoD AF version 1.0 draft [10-12]. DoD AF is an evolution of the C4ISR AF and intends to replace C4ISR AF as being the prime description framework for technical systems within the US DoD.

The target architecture was structured according to the three levels described in [19]; an *enterprise grid* (focusing on enterprise processes), an *information grid* (focusing on information services), and a *communication grid* (focusing on communication services). M2EE was used to develop an enterprise model describing the enterprise grid, while M2IE was used to develop a requirements and component models that focused on services and capabilities in the information and communication grid. Figure 14 illustrates this approach on a very high level of detail.
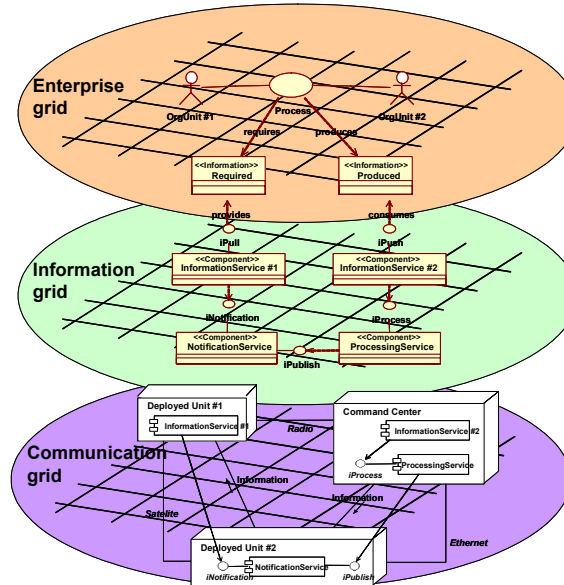

Figure 14. Target architecture for NWC

The MACCIS models were later used as a basis for documenting the work products according to the DoD AF. The component-oriented paradigm which the MACCIS framework builds upon provided a useful supplement to DoD AF, and made it easier to describe modern enterprises and infostructures. Further work around the relationship between DoD AF and MACCIS is planned.

5.   Conclusions and Future Work

5.1   Conclusions

Having presented the MACCIS framework through the last sections, this section will assess the framework related to the objectives listed in section 1.3.

1. MACCIS 1.0 was designed to describe what the C4ISR AF described with the addition of some RM-ODP concepts. The MACCIS 1.0 report includes a full mapping to C4ISR AF. MACCIS is based on multiple civilian standards; ISO RM-ODP, IEEE 1471 and UML to mention some. In addition to C4ISR AF, APP-6A has been referenced from the military domain.
2. UML has been used as the notation (in addition to small amounts of structured text) throughout the framework.

3. The usability of the framework is a hard objective to assess, as "simple enough" is a subjective statement. Experiences from usage tell us that there seems to be too many models to choose from, but that all of the models are required by some user, albeit no user requires all models. This leads to the conclusion that we need to focus on what models are needed for different user roles.

4. MACCIS has been used for both description of new systems (to-be) and existing systems (as-is). In a definition and procurement scenario the description of requirements is more important than in the maintenance scenario.

5. Since MACCIS 2.0 provides both an infostructure and enterprise version, the focus of the framework has been (and still is) description of the architectures of information infrastructures and their enterprise environments.

6. MACCIS will probably be useful for describing most software intensive systems. C4I is also a software intensive system, albeit with strict requirements to accuracy and security. The MACCIS framework does provide modelling terms that are specific to C4I.

All in all, MACCIS has covered most of the goals it was meant to cover. This is probably mostly because use and development of the framework have been relatively closely connected. Experiences in usage of MACCIS and in usage of related frameworks have been fed into the evolution of MACCIS.

5.2   Future Work

As mentioned, MACCIS, as it is defined today (version 2.0), is a result of evolution based on experiences made and new knowledge and requirements. The framework is still evolving. Some of the topics listed below are under work and some are wishes from the researchers.

- More support for enterprise aspects, not only limited to software intensive business processes. This feature has been requested due to high focus on business re-engineering over the last years. The work has started, but will for the time being be a stand-alone framework, not directly incorporated in MACCIS.

- Refinement of the infostructure aspects, e.g. with regards to security modelling. The description of security related issues are becoming even more important through the NCW paradigm. The MACCIS framework needs to extend its description power in this dimension.

- Integration, or rather harmonisation, with the DoD AF. This work has started and the results will be a part of subsequent versions of the MACCIS framework. Preliminary results indicate that the functional focus of DoD AF and the component focus of MACCIS need to be harmonised.

- Tool support and repository for models and analysis. Since UML is used as the notation for almost all of the products produced from using the framework, there is wide tool support for basic usage of MACCIS. However, in order for the framework to be a useful one, tool support specific for the framework is needed. Up until now this has been done by implementing frameworks and add-ins in IBM Rational Rose for modelling purposes. The goal is to create a repository that will allow for storage, traversal and data mining of MACCIS models. Emerging and existing standards such as UML 2.0, MOF and UML 2.0 Diagram Interchange Format will probably be highly relevant in this work.

- Some of the concepts introduced in UML 2.0 seem to be very useful in the MACCIS framework. The composite structure is an example of such. This mechanism will allow the creation of models that can easily bee "zoomed" between different levels of detail, using standard UML.
- In order to meet the usability requirements of using the framework, regarding easy-to-choose models, we will examine what models are needed for the different user roles.

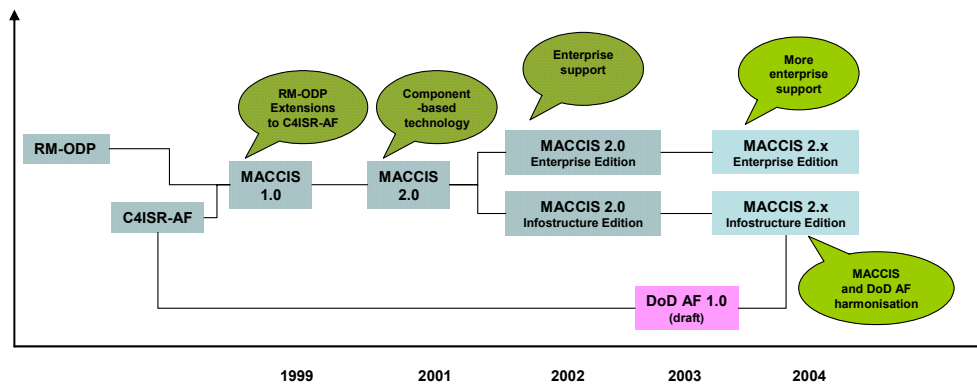The way forward for MACCIS has been summarised along a timeline in Figure 15.

Figure 15. Future direction of MACCIS

6. Acknowledgements

7. References

[1]   J. Ø. Aagedal, A.-J. Berre, B. W. Bjanger, T. Neple, and C. B. Roark, "ODP-based Improvements of C4ISR-AF," presented at Command & Control Research & Technology Symposium, U.S. Naval War College, Rhode Island, 1999.

[2]   C4ISR Architectures Working Group, "C4ISR Architecture Framework Version 2.0," Department of Defense, 18. December 1997.

[3]   ITU-TS, "Basic Reference Model of Open Distributed Processing - Part 1: Overview and guide to use the Reference Model," Rec.X901 (ISO/IEC 10746-1), 1995.

[4]   ITU-TS, "Basic Reference Model of Open Distributed Processing - Part 2: Descriptive model," Rec.X902 (ISO/IEC 10746-2), 1995.

[5]   ITU-TS, "Basic Reference Model of Open Distributed Processing - Part 3: Prescriptive model," Rec.X903 (ISO/IEC 10746-3), 1995.

[6]   ITU-TS, "Basic Reference Model of Open Distributed Processing - Part 4: Architectural Semantics," Rec.X904 (ISO/IEC 10746-4), 1995.

[7]   B. Elvesæter, J. Ø. Aagedal, A.-J. Berre, and T. Neple, "MACCIS - Minimal Architecture for C2IS - Part A: Overview and guide to use the MACCIS framework, Version 2.0," SINTEF Telecom and Informatics, SINTEF Report STF40 F01067, 15. December 2001.

[8]   J. Ø. Aagedal, B. Elvesæter, A.-J. Berre, T. Neple, J. Oldevik, and A. Solberg, "MACCIS - Minimal Architecture for C2IS - Part B: Specification of the MACCIS framework, Version 2.0," SINTEF Telecom and Informatics, SINTEF Report STF40 F01068, 15. December 2001.

[9]   R. K. Rolfsen and B. Elvesæter, "M2EE: MACCIS 2 Enterprise Edition," SINTEF Telecom and Informatics, SINTEF Report STF40 A02068, 12. December 2002.

[10]   DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.0 - Volume I: Definitions and Guidelines," Department of Defense, 30. August 2003.

[11]   DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.0 - Volume II: Product Descriptions," Department of Defense, 30. August 2003.

[12]   DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.0 - Deskbook," Department of Defense, 30. August 2003.

[13]   IEEE, "IEEE Std 1471-2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," IEEE, October 2000.

[14]   S. Tyndale-Biscoe, O. Sims, B. Wood, and C. Sluman, "Business Modelling for Component Systems with UML," presented at Sixth International Enterprise Distributed Object Computing Conference, Lausanne, Switzerland, 2002.

[15]   Object Management Group, "OMG Unified Modeling Language Specification Version 1.4," Object Management Group, formal/01-09-67, September 2001.

[16]   Object Management Group, "UML Profile for Enterprise Distributed Object Computing Specification," Object Management Group, ptc/02-02-05, 2002.

[17]   Object Management Group, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (Revised Submission)," Object Management Group, realtime/03-05-02, 2002.

[18]   NATO, "APP-6A: Military Symbols for Land Based Systems," 1998.

[19]   O. B. Eldar Aarhold, "Information grid in support of crisis management," presented at Command & Control Research & Technology Symposium, Naval Postgraduate School, Monterey, California, 2002.