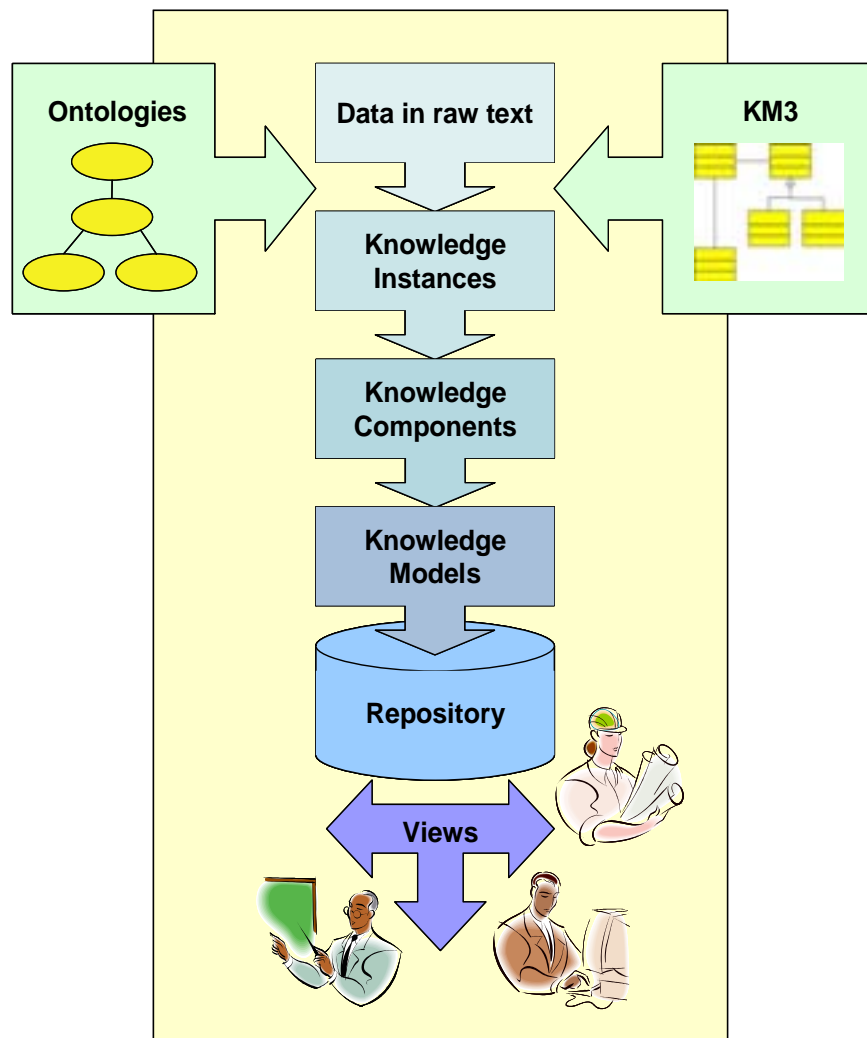


# DCMF – Defence Conceptual Modelling Framework

**Vahid Mojtahed, Marianela García Lozano, Pernilla Svan**  
FOI – Swedish Defence Research Agency

**Birger Andersson, Vandana Kabilan**  
KTH – Royal Institute of Technology



FOI is an assignment-based authority under the Ministry of Defence. The core activities are research, method and technology development, as well as studies for the use of defence and security. The organization employs around 1350 people of whom around 950 are researchers. This makes FOI the largest research institute in Sweden. FOI provides its customers with leading expertise in a large number of fields such as security-policy studies and analyses in defence and security, assessment of different types of threats, systems for control and management of crises, protection against and management of hazardous substances, IT-security and the potential of new sensors.



FOI  
Defence Research Agency  
Systems Technology  
SE-164 90 Stockholm

Phone: +46 8 555 030 00  
Fax: +46 8 555 031 00

[www.foi.se](http://www.foi.se)

# DCMF – Defence Conceptual Modelling Framework



<b>Issuing organization</b> FOI – Swedish Defence Research Agency Systems Technology SE-164 90 Stockholm	<b>Report number, ISRN</b> FOI-R--1754--SE	<b>Report type</b> Methodology report
	<b>Research area code</b> 2. Operational Research, Modelling and Simulation	
	<b>Month year</b> November 2005	<b>Project no.</b> E6035
	<b>Sub area code</b> 21 Modelling and Simulation	
	<b>Sub area code 2</b>	
<b>Author/s (editor/s)</b> Vahid Mojtahed Marianela García Lozano Pernilla Svan Birger Andersson Vandana Kabilan	<b>Project manager</b> Vahid Mojtahed	
	<b>Approved by</b> Monica Dahlén	
	<b>Sponsoring agency</b> Swedish Armed Forces	
	<b>Scientifically and technically responsible</b> Vahid Mojtahed	
<b>Report title</b> DCMF – Defence Conceptual Modelling Framework		
<b>Abstract</b> <p>The Defence Conceptual Modelling Framework has its origin in the CMMS concept presented by the US DMSO. DCMF is a framework for making conceptual descriptions and models of military operations. It consists of tools for their development and reusability, as well as standards for acquisition, representation, modelling, integration, management and preservation of knowledge.</p> <p>After a prestudy of the American concept, the Swedish FOI project suggested a long-term plan in order to successfully implement the DCMF concept within the Swedish Armed Forces. The DCMF project officially began 2003. This year we have tried to create our first conceptual model according to the DCMF. A DCMF process has been built, analysed and refined, and as a consequence, necessary tools and methodologies have been identified. The need of a (activity centric) tool for modelling the acquired knowledge, resulted in a new concept called, <i>KM3 – Knowledge Meta Meta Model</i>. Another important component of the framework is the Ontologies. An ontology suite and methodology have been developed for the DCMF purposes. All of this is discussed in the report.</p>		
<b>Keywords</b> Conceptual Modelling, Knowledge Acquisition, Knowledge Analysis, Knowledge Modelling, Knowledge Management, Ontology, Ontology Development Methodology, Knowledge Meta Meta Model, KM3, Interoperability, Reusability, Standardisation, Composability, CMMS, VV&A		
<b>Further bibliographic information</b>	<b>Language</b> English	
<b>ISSN</b> 1650-1942	<b>Pages</b> 146 p.	
	<b>Price acc. to pricelist</b>	

<b>Utgivare</b> FOI - Totalförsvarets forskningsinstitut Systemteknik 164 90 Stockholm	<b>Rapportnummer, ISRN</b> FOI-R--1754--SE	<b>Klassificering</b> Metodrapport
	<b>Forskningsområde</b> 2. Operationsanalys, modellering och simulering	
	<b>Månad, år</b> November 2005	<b>Projektnummer</b> E6035
	<b>Delområde</b> 21 Modellering och simulering	
	<b>Delområde 2</b>	
<b>Författare/redaktör</b> Vahid Mojtahed Marianela García Lozano Pernilla Svan Birger Andersson Vandana Kabilan	<b>Projektledare</b> Vahid Mojtahed	
	<b>Godkänd av</b> Monica Dahlén	
	<b>Uppdragsgivare/kundbeteckning</b> FM	
	<b>Tekniskt och/eller vetenskapligt ansvarig</b> Vahid Mojtahed	
<b>Rapportens titel</b> DCMF – Ramverk för militär konceptuell modellering		
<b>Sammanfattning</b> <p>Konceptet DCMF (Defence Conceptual Modelling Framework) har sitt ursprung i CMMS konceptet som presenterades av det amerikanska försvarets organisation för Modellering och Simulering (US DMSO). DCMF är ett ramverk för att skapa konceptuella beskrivningar och modeller av militära operationer. Det består dels av verktyg för deras utveckling och återanvändning samt dels av standarder för anskaffning, modellering, integrering, hantering och bevarande av kunskap.</p> <p>Efter en förstudie av det amerikanska konceptet, föreslog det svenska FOI projektet en långtidsplan för att lyckas implementera DCMF inom det svenska försvaret. DCMF projektet startades officiellt 2003. I år har vi försökt skapa vår första konceptuella modell enligt DCMF. En DCMF process har skapats, analyserats och förfinats. Som en konsekvens har nödvändiga verktyg och metodiker identifierats. Behovet av ett (aktivitetscentriskt) verktyg för modellering av anskaffad data resulterade i ett nytt koncept, KM3 (KunskapsMetaMetaModell). En annan viktig komponent i ramverket är ontologierna. En ontologi struktur och metodik har utvecklats för DCMFs behov. Allt detta diskuteras i rapporten.</p>		
<b>Nyckelord</b> Konceptuell Modellering, Kunskapsanskaffning, Kunskapsanalys, Kunskapsmodellering, Kunskapshantering, Ontologi, Utvecklingsmetodik för Ontologi, Kunskapsmetametamodell, KM3, Interoperability, Återanvändning, Standardisering, Komponentssammansättning, CMMS, VV&A		
<b>Övriga bibliografiska uppgifter</b>	<b>Språk</b> Engelska	
<b>ISSN</b> 1650-1942	<b>Antal sidor:</b> 146 s.	
<b>Distribution enligt missiv</b>	<b>Pris:</b> Enligt prislista	

## ***Executive summary***

The increasing use of modelling and simulation in the military domain puts high demands on how knowledge is used and managed. Major challenges are how to acquire, validate and maintain knowledge and how to achieve this with the minimum effort. To address issues relating to knowledge bases for modelling and simulation, the US DoD introduced in 1995 a concept called Conceptual Models of the Mission Space (CMMS). For unknown reasons the concept was never completed and the activities in it seemed to end around the turn of the century. However, the Swedish Defence Research Agency (FOI), found the concept interesting and has, since 2002, done research on the concept to explore its potential.

Conceptual Models of the Mission Space, CMMS, are simulation- and implementation-independent functional descriptions. These functional descriptions describe real world processes, entities, environmental factors, and associated relationships and interactions constituting a particular set of missions, operations or tasks. CMMS is also a framework for the development of models and it captures the characteristics of objects in a domain given by a scenario, such as their features, interactions and behaviour.

Hence CMMS is for all stakeholders a common description of what is to be simulated and serves as a bridge between the military experts and the developers. The military experts own the mission processes and are an authoritative source when validating the content of the conceptual models. CMMS also serves as a platform for communication among stakeholders working with these simulation models.

The work at FOI began in 2002 with an extended study of all known published material about CMMS up to that point. To understand the CMMS concept better with the aim of utilising it, a plan for a study with a focus on the early phases of the CMMS process was established in 2003. It was discovered early on that many of the specifications of the CMMS process were vague and unfinished. During the work it became more obvious that a lot of the necessary components, methodologies and tools to finish the process, were also missing. Examples of fundamental pieces missing included a structure by which Mission Space Models (MSMs – the final outcome of the process), were describable. A proposal for such a structure was made and it is called the Knowledge Meta Meta Model (KM3). Another fundamental piece missing was an ontology structure. Such a structure was not even mentioned in the original CMMS documents. This meant that to get a clearer understanding of the concept, a basis for a common methodological framework had to be developed. The main objective of this report is to present this framework, now called the DCMF – Defence Conceptual Modelling Framework.

The final result, MSMs - Mission Space Models - which are the kernel of both DCMF and CMMS, are defined as simulation and implementation-independent functional descriptions of the real world processes, entities, and environmental factors associated with a particular set of missions. These descriptions would be able to serve as a frame of reference for simulation development by capturing the basic information about the important entities involved in any mission, and their key actions and interactions. Thus *the overall objectives* for both DCMF and CMMS are: to capture authorised knowledge of military operations; to manage, model and structure the obtained knowledge in an unambiguous way; and to preserve and maintain the structured knowledge for future use and reuse. And *the premier aim* of doing so is to enable semantic and substantive interoperability of the future simulation models built on these descriptions.

However we added some requirements to the conceptual models developed by the DCMF in addition to what we could discover as requirements for the ones developed by the DMSO's CMMS. To summarise, the DCMF requirements for how the final conceptual models should be are as follows: (a) well documented, (b) readable and usable for a person as well as a machine, (c) composable, (d) traceable the whole way back to the original sources, and finally (e) useable as a basis for simulations models.

This year the work has mainly focused on making MSM prototypes and following through with the process to assess its feasibility and to gather experience. During this work we have also been able to further identify necessary tools, methods and techniques for analysing, representing and modelling knowledge. We have applied the DCMF process on a hypothetical scenario and thereby done some validation of developed tools and theories.

We had several candidate scenarios available. Evaluating the DCMF process by using the scenario has identified several issues in the different phases of the process. The scenario was given in a free text form which needed to be read (parsed) and interpreted. We found that the most appropriate parsing method depends on the purpose of the activity and who is performing it. The results will also be different depending on the methods used. For instance, if two method experts analyse and formalise a common scenario description they are more than likely to end up with two different formalisations. Future work involves designing stricter guidelines for the analysis and formalisation of information.

We have studied and analysed several of the contemporary ontology design and modelling methodologies. Based on this research and the requirements put forward for the DCMF, we have created a methodology, called MiSO (Military Specific Ontology development), to develop military specific ontologies. Using MiSO we proposed a multi layered architecture called Defence Conceptual Modelling Ontology Framework (DCMF-O) for modelling reusable knowledge for the military operations domain. We have also surveyed and compiled a collection of existing ontologies and other knowledge bases which may be included in the DCMF-O

The Knowledge Meta Meta Model (KM3) that was previously developed has been evaluated in several ways with mostly positive results. It has been found useful in a supporting role when interpreting data. This means that when confronted with ambiguous or otherwise unclear data, the KM3 can support an interpreter of the data by supplying concepts and structure. Furthermore, an interpreter may also use the KM3 to discover ambiguities in data. Another result is that the KM3 may be used to further structure semi-structured information. This ability is important when considering that one of the intended uses of the KM3, is to provide a way to structure information into a form that facilitates processing by machine.

Future work involves designing stricter guidelines for the analysis and formalisation of information. All steps in the proposed DCMF process have been done manually and we are looking into where and what tools are needed to automate the process to the largest possible extent. To formalise the methods for automatic tagging and extraction of explicit data from raw natural language or from other forms, and methodology for extracting implicit knowledge from domain knowledge are also suggested as future tasks.

Finally, by performing the DCMF work, we have gained valuable experience which takes the DCMF project one step closer to its goal and the task is considered to be very large, complex



and complicated with many challenges. We believe that a great effort is needed to reach the goal of knowledge reuse and interoperability. In case true interoperability between simulation models built on conceptual models is desired, then a real international cooperation in this field would be very valuable and welcome.



# Contents

<b>1</b>	<b><i>Introduction</i></b>	<b>11</b>
1.1	Background	11
1.2	Short introduction to the CMMS concept	11
1.3	Introduction to DCMF	13
1.4	Objectives and aims for the DCMF	15
1.5	Short introduction to this report	16
<b>2</b>	<b><i>The DCMF process</i></b>	<b>17</b>
2.1	The DMSO CMMS process	17
2.2	The DCMF process	17
2.3	Knowledge Acquisition (KA)	18
2.4	Knowledge Representation (KR)	21
2.5	Knowledge Modelling (KM)	22
2.6	Knowledge Use (KU)	23
2.7	Traceability and Management	25
<b>3</b>	<b><i>DCMF Components</i></b>	<b>27</b>
3.1	Overview of Tools and Methods for Knowledge Development	27
3.2	Data Flow in the DCMF-P	28
3.3	An example of the data flow in the DCMF	29
3.4	Components of the DCMF	30
<b>4</b>	<b><i>Ontology</i></b>	<b>33</b>
4.1	What is an ontology?	33
4.2	Types of ontologies	34
4.3	Ontology Design Criteria	36
4.4	Conceptual Modelling, UML and Ontologies	37
4.5	Ontology Design Methodologies	39
<b>5</b>	<b><i>Military Specific Ontology (MiSO) - Methodology and DCMF-O Framework</i></b>	<b>45</b>
5.1	MiSO in context of the overall DCMF Process	45
5.2	Requirement Criteria for DCMF-O	46
5.3	MiSO Development Methodology	47
5.4	DCMF-O Architecture	49
<b>6</b>	<b><i>The Knowledge Meta Meta Model - KM3</i></b>	<b>59</b>
6.1	Introduction	59

6.2	The KM3 .....	59
6.3	A High Level View of KM3 .....	62
6.4	Discussion and Summary .....	64
7	<i>Analysis Method (From KA to KR)</i> .....	67
7.1	Parsing methods .....	67
7.2	Chosen approach.....	70
8	<i>Case Study</i> .....	73
8.1	Objectives and aims with the scenario .....	73
8.2	Scenario candidates .....	73
8.3	Evaluation of the scenario candidates.....	74
8.4	The chosen scenario .....	75
9	<i>Case Study Analysis</i> .....	79
9.1	Overview of Scenario Analysis Process/ DCMF Process recap .....	79
9.2	Scenario Analysis Overview .....	80
9.3	Walk Through Example of Scenario.....	83
10	<i>Experiences gained</i> .....	93
10.1	KM3 Results Discussion .....	93
10.2	DCMF-O Design and Development Experiences .....	96
10.3	Merging the ontologies .....	101
11	<i>Discussion, Conclusions and Future studies</i> .....	105
	<i>Acronyms</i> .....	109
	<i>References</i> .....	111
	<i>Appendix A – Detailed KM3</i> .....	117
	<i>Appendix B – 05F-SIW-038</i> .....	123
	<i>Appendix C – 05F-SIW-040</i> .....	135
	<i>Appendix D – A graphical use case scenario diagram</i> .....	145

# 1 Introduction

## 1.1 Background

The ever increasing use of Modelling and Simulation in the Swedish Defence are placing large demands on knowledge management. Major concerns are how to secure, validate and maintain knowledge and how to achieve this with the minimum effort. In the progress of developing a practice facilitating this, key attention has been given to the Modelling and Simulation Master Plan (MSMP) developed by the US DoD. An important component of this vision was the CMMS concept, presented as an essential requirement for interoperability and reusability of *knowledge in the military domain*. [MSM05]

The Swedish Defence Research Agency (FOI) found the idea of the CMMS concept very promising. As a result, the Department of Systems Modelling carried out a study in 2001-2002, to investigate to what extent and under which circumstances the concept could be of interest for the Swedish Defence. The main emphases were on the aspects of reutilisation, interoperability and the maintenance of conceptual models, as well as on the applicability and relevance for the Swedish Defence. The result of the study indicated that CMMS was an interesting concept worth pursuing but that in its current state it was ambiguous, vague and partly unfinished [Lun04]. Thus, in continuation of the first study, FOI initiated an additional analysis in order to further develop the concept and to implement a framework that could support it.

A plan for future work was laid out, and we began in 2003 with work focused on the early phases of the CMMS-process such as Knowledge Acquisition. The year after, in 2004, we continued our work and focused on the following phases. During this work we found that a lot of the CMMS-process, such as components and tools were uncharted territory and therefore we had to basically start from the beginning. We soon discovered that several fundamental pieces were missing and this year, 2005, our work has mainly focused on developing the process further. We are doing this both by developing the identified and required process steps through theoretical analysis and simultaneously by means of a test case following through with the process to see if it is feasible and to gather experiences. Doing so, we have also been able to further identify necessary tools, methods and techniques.

This work is still in progress within the FOI CMMS project. But we are diverging so much from the original CMMS process that we have now decided to call our approach DCMF – Defence Conceptual Modelling Framework. So henceforth whenever we use the term CMMS we will be referring to the original CMMS concept introduced by the US DoD, and whenever we use the term DCMF we will be referring to the FOI CMMS project.

In this report we will give an overview and the current status of the DCMF project as well as providing a deeper look into ongoing activities and the experiences gained during this year in the project.

## 1.2 Short introduction to the CMMS concept

Conceptual Models of the Mission Space (CMMS) was put forward as a component of a vision for a future technical framework for Modelling and Simulation (M&S). CMMS is the second component of this M&S Common Technical Framework established by the US Department of Defence (US DoD) in their MSMP - M&S Master Plan in 1995. The main

purpose of the CMMS concept is to facilitate and support development and reuse and cooperation between simulation models. DMSO – Defence Modelling and Simulation Organisation defines CMMS as “First abstractions of the real world that serve as a frame of reference for simulation development by capturing the basic information about important entities involved in any mission and their key actions and interactions” [MSM05].

Conceptual Models of the Mission Space, CMMS, are simulation- and implementation-independent functional descriptions. These functional descriptions describe real world processes, entities, environmental factors, and associated relationships and interactions constituting a particular set of missions, operations or tasks. CMMS is also a framework for the development of models and it captures the characteristics of objects in a domain given by a scenario, such as their features, interactions and behaviour. The final conceptual models, or rather Mission Space Models (MSMs) should be generic and applicable to as many scenarios as possible without any loss of critical knowledge.

Hence CMMS is for all stakeholders a common description of what is to be simulated and serves as a bridge between the military experts and the developers. The military experts own the mission processes and are an authoritative source when validating the content of the conceptual models. CMMS also serves as a platform for communication among stakeholders working with these simulation models.

Since its introduction in 1995 CMMS has been replaced by other initiatives. For the sake of simplicity this report only refers to CMMS but presumably connections can be made to other concepts that we believe share a common goal as well, such as Functional Descriptions of the Mission Space (FDMS) and Knowledge Integration Resource Center (KIRC). Both FDMS and KIRC have been introduced by DMSO but there are no longer any links or references to these concepts apart from the very basic information about the FDMS which is available on the DMSO’s site [DMS05].

### **1.2.1 Goals for CMMS**

The goal for CMMS is to provide a basis for the development of consistent and authoritative simulation representations. In other words, CMMS will provide the basis for creating a real world associated knowledge base for military operations and processes which can be used for analysis, design, verification and validation and preferably also accreditation and certification. The concept tries to address the following problems, which are related to the development of Mission Space Models (MSMs) [MSM05]:

- To gain access to authorised knowledge of military operations is not easy, and modellers instead rely on various, sometimes not authoritative, sources for the same information, which leads to a confusion of ideas.
- The supplied knowledge (information) is not sufficient enough to be used in simulation model development, or it is ambiguous.
- The obtained knowledge, which often has been achieved at a very high price, is not preserved for future use, which leads to unnecessary repetition of work.

### **1.2.2 CMMS Components**

The primary and most important part of CMMS includes the domain specific conceptual models, so-called “Mission Space Models” (MSM). They are consistent, structured and functional descriptions of real military operations or processes. The other parts are the

“Technical Framework”, a common library with a database management system, and a group of supporting tools for different parts of the CMMS process.

There are four principal components of CMMS [CMM97]:

- *Mission Space Models (MSMs)*: a standardised way to create consistent functional descriptions of real-world military operations.
- *Technical Framework [CMM97]*: interoperability standards for Knowledge Acquisition (KA) and Knowledge Engineering (KE). The CMMS-TF provides among other things:
  - a *common semantics and syntax* for describing the mission space
  - a closed-loop *engineering process* for creating and maintaining conceptual models
  - *data interchange standards* for conceptual model integration and interoperability
- *Common Library*: a database management system for model registration, storage, management, and release.
- *Supporting Tools, Utilities and Style Guides*.

### 1.3 Introduction to DCMF

The rapidly increasing use of modelling and simulation in the Swedish Defence, as in any other defence organisation with high ambitions and ambitious goals, are putting taxing demands on knowledge management. On the other hand, as we mentioned earlier in the Background section, the CMMS concept was presented by DMSO as a component of an M&S framework, as an essential requirement for interoperability and reusability of knowledge in the military domain. This was sufficient reason for us to carry out a study to understand the concept and its details better. The result of this feasibility study was very encouraging, mostly because the purpose and goals of the CMMS concept were so promising.

The purpose of CMMS was to facilitate and support development, reuse and interoperability between simulation models. These simulation models are supposed to be built on conceptual models which are simulation and implementation independent. They are conceptual models of real world processes, entities, environmental factors, associated relationships and interactions constituting a particular set of missions, operations or tasks. Constructed models are supposed to be as generic and applicable to as many scenarios as possible without any loss of critical knowledge. One benefit of such models is that they act as a common description for all stakeholders of what is to be simulated, and thus serve as a bridge between the military experts and the developers.

The objective and the purpose of the CMMS, according to the above, were very tempting and the structure seemed to be so good and carefully prepared so we decided to initiate an additional analysis in order to further investigate the concept. For this reason a seven-step plan was developed to tackle the challenges, starting in 2003.

The CMMS process consists of a number of phases where knowledge is refined and enriched. The plan included an analysis of the early phases of the CMMS process, called the Knowledge Acquisition (KA) and Knowledge Engineering (KE) phases, a grading of the usefulness of the tools supporting the entire process, and a conducting of a first iteration of the complete process in order to create a real simulation model (a Mission Space Model – MSM).

A short version of the seven-step plan for the DCMF is:

1. Study the KA and KE phases more closely.
2. Analyse the CMMS process in greater depth to identify necessary tools and evaluate them based on requirements formulated in step 1.
3. Analyse the language issues such as terminology, ontology, common semantics and syntax.
4. Tackle the entire mode of procedure for developing MSMs. Develop new techniques where needed but try to adhere to existing standards.
5. Having come this far we should be able to analyse all parts of the CMMS concept from the point of view of the Swedish Defence's needs.
6. Implement a prototype of the complete CMMS process with the user and other stakeholder interfaces (e.g., VV&A-agents, model developers, SME, sponsors, etc.)
7. Integrate CMMS research into other parts of the Swedish defence, in particular the Swedish Network Based Defence research programme.

But the journey through these steps has not been easy. The more we worked and the deeper we came, the less support there was, and the more we penetrated the different CMMS process steps the more ambiguous and less concrete everything became. The first and absolutely most troublesome obstacle one encounters when dealing with CMMS is the lack of available documentation. There no longer seems to be a central authority actively administrating the concept. Perhaps this is not so surprising given the fact that the initiative appears to have ended some time around year 2000 and changed scope and name to Functional Descriptions of the Mission Space (FDMS). Short after that, in continuation of CMMS and FDMS, yet another concept Knowledge Integration Resource Center (KIRC) was presented. However, no formal, public document explains why these changes came about. Today the DMSO is back on the FDMS track and there seems to be no way of telling where the original CMMS currently stands. Furthermore, much of the available documentation appears to be unfinished and lacks references to its authors, making it hard to follow up and to initiate discussion about these topics.

By 2004 we had identified some more concrete problems and limitations of CMMS which we presented in a paper [Lun04]. These limitations and the corresponding solutions suggested are briefly:

- *Unsupported Knowledge Acquisition (KA)*: Develop a complete methodology for KA in this domain. This is explained further in Chapters 2 and 3 of this report.
- *Lack of clarity of modelling elements*: Create meta levels or abstraction layers to separate groups of modelling elements. This is explained in Chapters 4, 5 and in Part 6.
- *Need for alternative knowledge representations*: Introducing the Knowledge Meta-Meta Model (KM3). This is explained in Chapter 6 and Appendix A.
- *Limitations of processes*: An action centric approach to add Dynamic Knowledge without limiting the processes. This is explained in different places in this report, but mainly in Chapters 6 and onwards.

Since CMMS is concerned with military missions based on activities, we believe that it must not only show the structure and order of these activities but also their cause, conditions and effects. These aspects are not addressed much in CMMS and we believe the question still needs further work. Any model striving to capture knowledge should also denote a mapping to different knowledge representations or at minimum define a clear way to handle ways of



representing static and dynamic knowledge. (Read more about this and KM3 in Chapter 3 and Chapter 6.)

These limitations were the reason why we began to develop the concept further and to try implement a framework that could support it, which we today call the DCMF – Defence Conceptual Modelling Framework. We have tried to follow our seven-step plan, which is still valid. Focus has so far been on the early steps of the DCMF process, Knowledge Acquisition and Knowledge Engineering, and special attention has been given to new ways to represent knowledge and thus support substantive interoperability. Steps two and three (of the seven-step plan) have been partly taken care of by now. Some of our findings have now been presented in two papers [And05] and [Gar05]. During 2005 we have mainly been working on step 4 of our seven-step plan, which this report is largely about.

### **1.4 Objectives and aims for the DCMF**

One could say that our objectives and aims for the DCMF are generally the same as US DoD's for the CMMS. That is why we have been able to reuse as much as possible of what already has been introduced and done by DMSO. We have then tried to develop it further, in the hope and expectation that it can gradually be of benefit to others, perhaps even the DMSO.

The final result, MSMs - Mission Space Models - which are the kernel of both DCMF and CMMS, are defined as simulation and implementation-independent functional descriptions of the real world processes, entities, and environmental factors associated with a particular set of missions. These descriptions would be able to serve as a frame of reference for simulation development by capturing the basic information about important entities involved in any mission and their key actions and interactions. Thus *the overall objectives* for both concepts are: to capture authorised knowledge of military operations; to manage, model and structure the obtained knowledge in an unambiguous way; and to preserve and maintain the structured knowledge for future use and reuse. And *the premier aim* of doing so is to enable semantic and substantive interoperability of the future simulation models built on these descriptions.

However we have some more requirements on the conceptual models developed by the DCMF in addition to what we could discover as requirements for the ones developed by the DMSO's CMMS. Summarised these DCMF requirements for how the final conceptual models should be are: (a) well documented, (b) readable and usable for a person as well as a machine, (c) composable, (d) traceable the whole way back to the original sources, and finally (e) useable as a basis for simulations models.

The other objectives of DCMF (which we believe have also been part of the vision for DMSO's CMMS) are to achieve the following additional advantages. First, to produce a disciplined procedure by which the simulation developer is systematically informed about the real world problem to be synthesised. Second, to deploy a set of information standards the simulation subject matter expert employs to communicate with and obtain feedback from the military operations subject matter expert. Third, to provide a real world, military operations basis for subsequent, simulation-specific analysis, design, and implementation, and eventually verification, validation, and accreditation. Finally, to be the means for establishing reuse opportunities in the eventual simulation implementation by identifying commonality in the relevant real world activities.

DCMF constitutes an important step in the implementation of the Swedish Defence's modelling and simulation plan by initiating the first study of how a common library of verified and validated conceptual models of military operations can be developed. This could create the basis for the defence's future simulation models. In the long term it could help making the simulation software less expensive, easier to develop and maintain, and achieve both higher quality and a higher level of interoperability.

### ***1.5 Short introduction to this report***

The DCMF Project - and one of its results, the report, has been a commission by the Swedish Military Headquarters. This report is written in collaboration by FOI and DSV, a department of KTH, and has been categorised as a methodology report, describing the ongoing activities and the experiences during the work of this year. This report will present the properties, characteristics, design and experiences of the DCMF as a method. The report should be able to serve as a foundation anyone who wishes to understand the framework and method better, is interested to develop it further, or aims to use the framework and methods.

The scope of this report is on the whole limited to the activities of 2005, with focus on those parts of the process that mostly deal with how information is analysed, represented and modelled to suit our needs. The different analysis methods, necessary components, different roles involved, and process steps will be described through the report.

To obtain experience concerning practical application of DCMF we planned to create our first MSM – Mission Space Model, even if only in prototype format. Given a simple scenario, through limiting and controlling the work throughout the various steps of the DCMF Process, we could identify many pitfalls and the lack of important parts. We soon discovered that several fundamental pieces were missing and this year's activities have largely focused on developing those parts. With this work we have also been able to further identify necessary tools, methods and techniques, which are mainly about ontologies, meta modelling and other methods and techniques to model and structure knowledge.

Trying to create our first MSM prototype according to the DCMF has led us through the DCMF process, (1) to see if it is feasible, (2) to gather experiences and (3) to identify missing gaps where we need to focus our next research. It has given us a good idea of the supporting tools required for various phases of the DCMF Process, including a structure and management system for the end results, i.e., the MSMs. This experience will constitute a significant element in our future methodological research.

After this general introduction, the report will take you through the DCMF process both in general and in detail, and then discuss some of the most important parts of it, such as the KM3 and ontology structures. Then our Case Study and the methods and techniques used to transform scenario specific information into structured general knowledge, will be presented. We will conclude the report by discussing the results and outcomes so far, point out some difficulties and challenges, and state some future work in this area.

## 2 The DCMF process

This chapter is about the DCMF process. We will discuss the background e.g. the DMSO CMMS Process, before going into the general layout of the process. Finally we will look deeper into some of the more interesting details. An excerpt of this process was presented at the Fall SIW workshop of 2005; see Appendix B or [Gar05].

### 2.1 The DMSO CMMS process

In the CMMS Technical Framework Specification, see [CMM97], by DMSO the CMMS concept, components and process were outlined. The document described the framework, a number of different technical standards, administrative procedures, and layout of the infrastructure needed to build conceptual models. This document was a good starting point for us but it was unfortunately not enough, see [Lun04] (a general problem we had was to find any documentation). The CMMS-TF did not provide us with enough details to develop our own conceptual models of the mission space. The description was at an advanced level where some parts were described in more detail but others were vague and sometimes even ambiguous.

Another issue was that we had a slightly different focus than DMSO. In the CMMS-TF the resulting conceptual models, called Mission Space Models (MSMs), were oriented towards describing military processes. We found that approach to be too inflexible and not easily extendable. Take for example the process of driving a car or a tank, the process changes with the current weather conditions, the kind of vehicle, the kind of surface (tarmac, ice, gravel, etc), the environment (city, race track, etc), and so on. There are some basic activities that are common to all of these situations and by combining these in different configurations it is possible to deal with most of these without having to describe all the activities over and over again. We were therefore much more focused on an activity-centric approach.

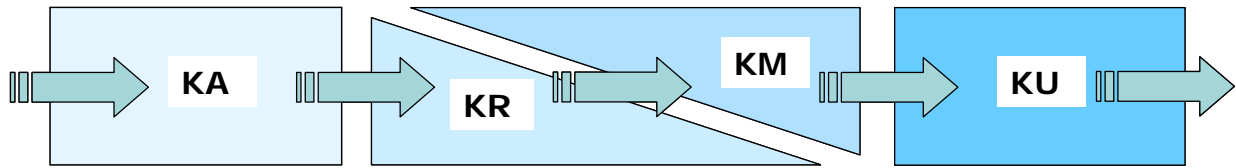
The DMSO CMMS process was described as a two phase process, where the two phases were Knowledge Acquisition (KA) and Knowledge Engineering (KE). In the KA-phase there were two main steps, where the development of a focused context (i.e. specifying the purpose, delimitations and focus) is the first and the gathering of information the other. The KE-phase consisted similarly of two steps where the formalising of data and construction of MSMs was the first and construction of CMMS<sup>1</sup> was the other.

### 2.2 The DCMF process

With a starting point from the DMSO CMMS process we began by splitting up the Knowledge Engineering phase into two new phases that we chose to call Knowledge Representation (KR) and Knowledge Modelling (KM). The reason for this separation was that we found the engineering phase to be very complex. After analysing the engineering phase we saw that there were two main parts where the first had to do with the analysing, formalising and representation of the acquired knowledge and the second had to do with the modelling of the acquired, formalised and structured knowledge.

---

<sup>1</sup> CMMS – is the name of both the concept and the conceptual model.



**Figure 2-1:** The four main parts are KA - Knowledge Acquisition, KR - Knowledge Representation, KM - Knowledge Modelling and finally KU - Knowledge Use.

We also chose to emphasise the importance of how to present, use and reuse the end products, and also how to present information in a user adapted way. There must also be a way to describe and insert the experiences from using conceptual models (that are the result from going through the process) into the DCMF knowledge base. Therefore we introduced another final phase, the Knowledge Use (KU) phase. With our changes the DCMF process consists in large of four main parts, see Figure 2-1.

The DCMF process is an iterative process where some parts and steps sometimes are done in parallel. By the end of every phase there are one or more products that are used as input for the next or a following phase. Depending of the purpose for acquiring knowledge some products may be ready for use after perhaps only one or two phases. We have tried to create a flexible process that gives the users and developers freedom of choice. The following sections will describe the individual phases in some more detail.

### 2.3 Knowledge Acquisition (KA)

The Knowledge Acquisition phase is the first phase of the DCMF process. The purpose of the Knowledge Acquisition phase is, as the name suggests, to acquire information and knowledge. In a wide sense it could be said that this phase is about learning. Historically, man has always tried to preserve established knowledge through a variety of different techniques and when the computers entered the scene during the 20th century new forms of storing information arose. In the middle of the 1980s the dream of artificial intelligence (AI) blossomed. The idea behind AI was to transfer knowledge within a domain to computer systems to enable intuitive data based systems to reason and make decisions about different issues. To make this dream possible, it was necessary to have correctly acquired and stored knowledge in these database systems. As a consequence a lot of interest was directed towards knowledge acquisition as a research area.

KA in DCMF consists of three main parts; the first is the focused context where the scope and delimitations of the knowledge requirements are decided, the second is the identification of authorised knowledge sources and the third is the actual engineering (acquiring, gathering and documentation) of data. The most important step is to know what the purpose of the KA is. Depending on the answer, different kinds of knowledge sources and methods may be used. Preferably, the sources used have been authorised by some organisation beforehand<sup>2</sup>. Information sources can be anything from books, web information, papers, regulations documents, pictures, maps, case studies to interviews with Subject Matter Experts (SME).

<sup>2</sup> The issue of how to authorize knowledge sources or who should do it is not something that we have studied.

A challenge of KA is that it often begins with the fetching of information from descriptions about a certain domain, for example through books, papers, tutorials, etc. All stored information is static while reality is dynamic and in constant change, which, if nothing is done, result in information that has been correctly gathered with time becomes out of date. Often the information that is needed for a certain purpose is not documented and is only available through SMEs. This knowledge is often expensive to gather, recount and store, since there is no easy way to do it. The art of gathering information from experts is usually called *knowledge elicitation* and is considered to be one of the greatest challenges of KA, among other things because there seldom exist a common starting point for *knowledge engineers* (the persons responsible for acquiring, gathering and documenting the data) and experts to reason about. The experts are also often themselves not actively aware of the knowledge they do possess. Expertise usually belongs to one of these four categories: domain, task, strategic or inference. Another challenge is to capture and obtain unspoken knowledge, things that the expert does routinely, without much thought and is considered obvious by the expert. The more knowledge an expert possesses, more is often considered obvious and it is usually more difficult for him to recount what he or she knows.

Some examples of techniques that can be used for knowledge elicitation are: (for more extensive lists with some discussions on the advantages and disadvantages, see [Bur05, Tke05]).

- *Interviews:*
  - *Unstructured:* The expert has a general discussion of the domain, designed to provide a list of topics and concepts.
  - *Structured:* The interviewer asks the expert or end user questions relating to a specific topic.
- *Problem-solving:* The expert is provided with a real-life problem, something that they deal with during their working life and are then asked to solve it. As the expert does so, he or she is required to describe each step, and the reasons for doing it.
- *Prototyping:* The expert is asked to evaluate a prototype of a system.
- *Simulation:* The expert is asked to use a simulator so that the expert's behavior can be observed.
- *Dialogue:* The expert interacts with a client, in the way that they would normally do during their normal work routine.
- *Sample lecture preparation:* The expert prepares a lecture, and the knowledge engineer analyses its content.
- *Questionnaires:* These are useful when the knowledge is to be gathered from several different experts.

A risk with using several experts is that each expert might use different terminology or emphasise different things. A method for solving this can be to have one expert write something and then use a system of peer-review to iteratively refine the data<sup>3</sup>.

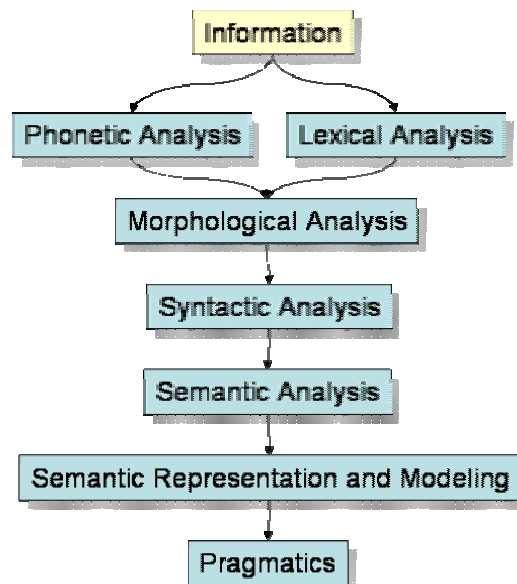
When acquiring information it is important to follow some well documented method that suits the purpose. Because when the information is gathered by interviews great care must be taken to avoid unintentional influence of the SME by the interviewer. Examples of methodologies for knowledge elicitation through interviews that we have studied are: CommonKADS, Generic Tasks and DESIREE, for more information on the different methodologies and our

---

<sup>3</sup> Real life examples of this is for example the Wikipedia project, it is a free encyclopedia that anyone can edit, see their main page at [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page).

study of them, see [Moj03, Bra95, Cha93, Sch00]. There are also some computer-based knowledge acquisition tools, but we have not evaluated any.

Another aspect of the KA-phase is that there are some important analysis steps that occur with the information when it is being gathered, see Figure 2-2. It has to do with the *linguistic process* that the information goes through (to read more about linguistics see [Wik05]). There is first a *phonetic* or *lexical* analysis depending if the information is acquired by interviews or by written form. Words or expressions that are in a different language or from another domain are translated or mapped to the current language or domain. The second thing that happens is the *morphological* analysis of the received information. That is, the information that is not relevant to our context is removed, documented in another context or simply not documented at all. Finally there is also some *syntactic and semantic* analysis done by the information engineer. These analysis steps are usually done quite unawares by the information engineer but since they are a source of error it is necessary to be aware of them. In that way pre-emptive measures can be taken, for example by recording an interview, asking control questions, having the expert approve of the gathered information etc.



**Figure 2-2:** Information is processed by the knowledge engineer in several steps before it is useful from a DCMF-perspective. Some of these steps (phonetic and lexical analysis, morphological, syntactical and semantic analysis) belong in the Knowledge Acquisition phase and others (semantic representation and modelling and Pragmatics) in later phases (Knowledge Representation, Knowledge Modelling and Knowledge Use).

- *Phonetic analysis* is done when information is gathered by interviews, discussions, talks or anything that is spoken. This is done quite unconsciously and allows us to interpret what is being said. This analysis is based on previous experience and may therefore lead us wrong sometimes. This is what helps us to determine if what was said was *rudder* or *robber*, *hat* or *hot*, and so on.
- *Lexical analysis* is like the phonetic analysis but for written information.
- *Morphological analysis* is what allows us to distil useful and plausible information. It helps us identify the internal structures of words and thus allowing us to further understand what is being said or written. This is also done quite unconsciously if one knows the language well.

- *Syntactic analysis* is used to decide how words are combined and arranged to form grammatically correct sentences. This helps us see how something is expressed and decide whether words are nouns, verbs, adjectives and so on which makes it easier to look them up and see what they are implying.
- *Semantic analysis* is used to decide the meaning of words and how they are combined to form meaningful sentences. It has more to do with the use of words than the nature of the entity being referenced by the word (this is the task of an ontology).

The persons involved in this phase need to have a deeper understanding of the problem domain and preferably experience in certain areas if information is to be gathered correctly. The roles involved in this phase and their tasks are:

- *Authorisations Agency*. Identification of suitable knowledge sources, authorising them and describing them.
- *Knowledge engineer*. Provides experience in acquiring, gathering and compiling information. For interviews knowledge about knowledge elicitation methods is important.
- *SME*. Provides the domain, task, strategic or inference knowledge.

## 2.4 Knowledge Representation (KR)

The aim of the Knowledge Representation phase is to analyse structure and formalise the acquired information. This is a step towards making the human readable information also machine readable. The structuring and formalisation should be done in such a way that no (or little) information is lost in the process and preferably so that the structured knowledge can be traced back to the source. The analysis and formatting of the information in the KR phase is a mainly syntactic and semantic representation and modelling process but *pragmatics* also influences the interpretations.

- *Pragmatics*: this is how language, sentences or knowledge is used, in different contexts. It helps us decide what actions to take or how to respond to information in a certain situation. For our purpose it helps us decide what knowledge is suitable for a certain need.

Structuring and formatting a text can be done with several methodologies; the important issue is *how* the structured text should be further analysed, processed and used. All methodologies for processing text and contents have a different focus and thus yield different results. We have mainly looked at three different methods and tools; SPO, 5Ws and KM3, read more about them in Chapter 7. After, or rather at the same time that the analysis and structuring is done, the information should also be mapped to a suitable *ontology*. The ontology gives the context of the domain, the terms and their relationships and interactions. The choice of ontology influences the analysis by giving a frame of reference for how the information should be interpreted.

In our case studies at FOI (read more about them in Chapters 8 and 9) we have studied a number of existing ontologies and information exchange models and have found two to be suitable for our purposes; IEEE's SUMO and NATO MIP's JC3IEDM. We have also seen the necessity of more specialised domain ontologies. Read more about Ontologies and our work with them in Chapters 4 and 5.

In the DCMF process *Verification, Validation and Accreditation* (VV&A) are an important help in the management of knowledge. Verification and validation aim to increase the

credibility of models and simulation results by providing evidence and indication of correctness and suitability. Correctness refers to concepts of consistency and completeness while suitability refers to capability, fidelity and accuracy.

- *Verification* deals with the examination of correctness and addresses the question: Is the model correct according to the specification? Has the model been built right? Verification is the process of demonstrating that a model is correctly represented and transformed correctly from one presentation form into another, according to all transformed and representation rules, requirements and constraints.
- *Validation* deals with the suitability of a model. Model validation is the process of demonstrating that a model and its behaviour are suitable representations of the real system and its behaviour with respect to an intended purpose of model application. Questions asked when dealing with validation are: Is the simulation model an accurate representation of the System, for the particular objectives of the study? Has the right model been built? Is the model suitable for the defined purpose? In what extent is it suitable? No model could be a copy of the referent. But is it good enough for the given purpose?
- *Accreditation* is the official certification that a model, simulation or federation of models and simulations is acceptable for use for a specific purpose. It is a bureaucratic act, during which a model results officially are declared as acceptable for a specific purpose.

Well performed VV&A in M&S increases the confidence in the model and reduces the risks in using the simulation results. If the VV&A information are stored it facilitates the VV&A process and also in the long term facilitates reusability of the model and the simulation results. For more information on VV&A see [Bra04, Law00].

As a conclusion, there are many different roles involved in this phase that require expertise and experience in various domains:

- *Analysis and formatting expert*: experienced in the appropriate formatting analysis method and technique.
- *Ontology expert*: knowledge in the used ontology, experience in mapping and interpretation of information to the ontology as well as knowledge in how to further develop and extend ontologies.
- *VV&A-agent*: experienced in the VV&A-process and methodology

## **2.5 Knowledge Modelling (KM)**

As previously mentioned the Knowledge Representation and Knowledge Modelling phases run pretty much in parallel, the difference between the two is the focus of each phase. In the KM phase the emphasis is on the semantic analysis and modelling of the information. In this phase pragmatics is also an important part of the analysis and modelling. After the Knowledge Acquisition and Representation phases the resulting products may already be usable for some applications. But, if the aim is to make general and reusable knowledge models then further knowledge modelling is needed. The largest difficulty of the knowledge modelling phase is to build a common general model at the right level of abstraction. Depending on the purpose, several different kinds of models can be the result of the same data. The previously mentioned Mission Space Models (MSM) are only one option. The result of the modelling may also be that certain information is lacking and a return to the KA or KR phases needed.

Another difficulty of the KM phase is how to make sure that knowledge is stored for future



use and reuse. As mentioned in Chapter 1 “storing previously acquired knowledge for future use” is one of the purposes with the DCMF. To make the conceptual models reusable we have thought in the lines of *divide and conquer*. By breaking down the knowledge into smaller components these knowledge components could then be reused in different configurations producing new conceptual models. A MSM can consist of one or more knowledge components. There are a number of advantages to this course of action: a) it is more flexible and easier to reuse the components b) they may be reused for other purposes than they originally were created for and c) components on different levels of abstraction could be combined. The challenge lies in creating well defined interfaces, descriptions and ontologies for the components, as well as finding the right scope for each component.

Another task of the Knowledge Modelling is to merge the new conceptual models or components with the ones previously created. Ideally the collection of acquired knowledge has been stored in some sort of knowledge base or knowledge repository. This repository is called the DCMF repository. There are several issues to consider:

- Data from the same domain may be merged
- The same fidelity level is necessary
- The same or similar purpose is necessary
- The same focus (i.e., activity centric, process centric, object oriented, etc.) is necessary

***With that said, it is not a bad idea to relate different views of the same scenario with each other. This only gives a fuller view of the domain in question. In this phase as well as in the Representation phase VV&A and Ontologies are essential parts.***

In this phase there are also many different roles that require expertise and experience:

- *Analysis and modelling expert*: people with experience in the appropriate analysis and modelling methods and techniques. They should also be familiar with what is in the knowledge base and the appropriate way of adding and merging modelled knowledge or components to the knowledge base.
- *Ontology expert*: here the ontology expertise is also necessary. Knowledge in the used ontology, experience in mapping and interpretation of information to the ontology as well as knowledge in how to further develop and extend ontologies.
- *VV&A-agent*: experienced in the VV&A-process and methodology.

## **2.6 Knowledge Use (KU)**

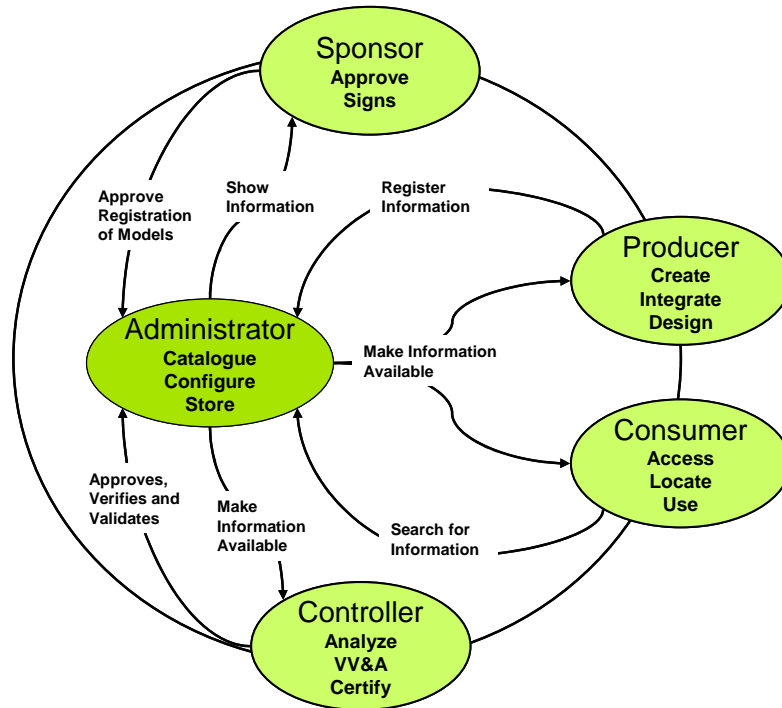
The final phase of the DCMF process is the actual use of the acquired knowledge. There should be one or more products that are the result of the three previous phases. If it is assumed that all of these products have been previously made and stored in some common DCMF repository, the challenge is to find the appropriate models for the users needs. In this phase the connection is strongest to the end-user and therefore it is of great interest that the knowledge can be visualised in different ways depending on the domain and the user's purpose or rights. This phase deals mostly with the pragmatics of the modelled knowledge.

There are a variety of users that may use the assembled information. The users can be divided into these four main groups; Sponsor, Consumer, Producer and Controller. There is also administrator functionality involved that acts as a broker and toolbox for the other users, see Figure 2-3. This area is not yet fully explored but work is in progress.

- *Sponsor*: he/she is the one who initiates the whole process and when an object (an MSM, knowledge component, knowledge instance, metadata, history, ontologies, etc.)

is registered in the DCMF Repository he must approve it and allow it to be made accessible to consumers and other users. Due to security and ownership issues there could also be some access control setting. If the object is not approved in its entirety it must go back to the producer for modifications.

- *Producer*: It is the producer that develops the objects that are then registered and stored in the repository. The producer has access to tools where objects can be described, registered, modified and sent for approval to the sponsor.



**Figure 2-3: The DCMF Users**

- *Consumer*: The consumer is the user that uses the DCMF Repository to locate and then use the MSMs. In the Consumer's view there should be tools to find the correct information. Information should be searchable in a semantic way and not only keyword based. As a result the consumer would get a number of models and components that could fit the query. The user could then proceed to choose and examine the most promising results. The results of the query should also contain metadata associated with the models and components, like ontologies, history, experiences or commentaries belonging to a model, etc. There should also be functions and tools for viewing these associated files.
- *Controller*: The Controllers are the persons involved in Verifying, Validating, Accreditation and Certifying objects that are stored in the repository. The controller has access to a set of tools and methodologies to aid him in his work. An example of this could be the REVVA-process, see [Tha04]. All notes and commentaries that the controller has made during this process should also be stored in relation to the object, to help ensure traceability.

The Knowledge Use phase can sometimes be a part of the Knowledge Acquisition phase. When knowledge that previously has been acquired, analysed, modelled and stored is used for new purposes it is important to be aware of the original purpose of the stored knowledge. It is also important to be aware of how (or if) the new purpose is in conflict with the original

purpose. Here the Controller has an important task to make sure that erroneous knowledge isn't used. Traceability is also important in this phase. All the products that have been the result of a Knowledge Acquisition effort should be traceable back to the original source and documents.

As mentioned in the previous chapter, the CMMS Concept that the DCMF concept has its roots in, was originally suggested as a framework to facilitate modelling and simulation. For that reason there has long been the question of what the connection is between the products that result from this process and an implemented simulation model. The Model Driven Architecture concept [OMG05], has been found to be a possible candidate for this connection. The concept is based upon the idea that from one and the same conceptual model, code can, through several transformation steps, be obtained for different platforms and languages. OMG05 is a complex concept that probably complements the DCMF concept very well. This matter needs further study, tests and evaluation. Other fields of application for the final products of the DCMF process, for example the MSMs, could be as references for VV&A-efforts, context sensitive reference manuals or even for the identification of needed services within the Network Based Defence. Below is a small list of potential users and their uses:

- *Sponsors, Orderers*: when specifying requirements, to describe extent and delimitations
- *End users, War fighters*: to formalise and describe their demands and to evaluate the results
- *Evaluators, VV&A-agents*: to verify and validate models and use the conceptual models as a basis for accreditation of simulation models
- *Researchers, engineers*: as a tool and aid in their modelling work
- *Analysers, designers, developers, programmers*: to understand the components of military operations and processes within the developed simulation

## **2.7 Traceability and Management**

Traceability and VV&A reoccur as a common feature throughout the entire process. In several steps there is some kind of feedback to earlier steps or phases if there has been any missing, misinterpreted or ambiguous information. It can also be that the analysis, formatting or modelling hasn't been done correctly and therefore needs more work before approving. Sometimes it has been an SME that has had to approve and sometimes a Controller, like a VV&A-agent, or some other authorised person or organisation. The issue of approval puts great demands on the person or organisation responsible for it and should be further investigated. As mentioned in the process description, there are many different roles and functions that are involved in the different steps of each phase. Further analysis of their authority and necessary competences should be done.

For VV&A-agents the DCMF process and the final products could be a great aid in the VV&A-process. The question of how cooperation can be done and how the connections between the two fields of VV&A and DCMF look like should also be further analysed and clarified.

Another important connection is to remember what initiates a run through the DCMF-process - the need of information for some specific purpose. Regardless of whether the sought information can be retrieved from previously acquired knowledge stored in the DCMF Repository or not, new models are created by performing the steps of the process. VV&A has a natural role here in securing that the correct information is acquired, that the correct model is produced, that it is valid for the purpose and used in the correct way, and for this

traceability is very important. All of these are reasons which indicate that there is a need for knowledge transfer between VV&A and DCMF.

For the traceability and management of knowledge we have seen the need for a repository that can aid in keeping track of all end products such as scenario descriptions, knowledge instances, knowledge components, MSMs, related metadata, use history, VV&A-comments, and so on. To keep track of all this data ontologies and information models are necessary.

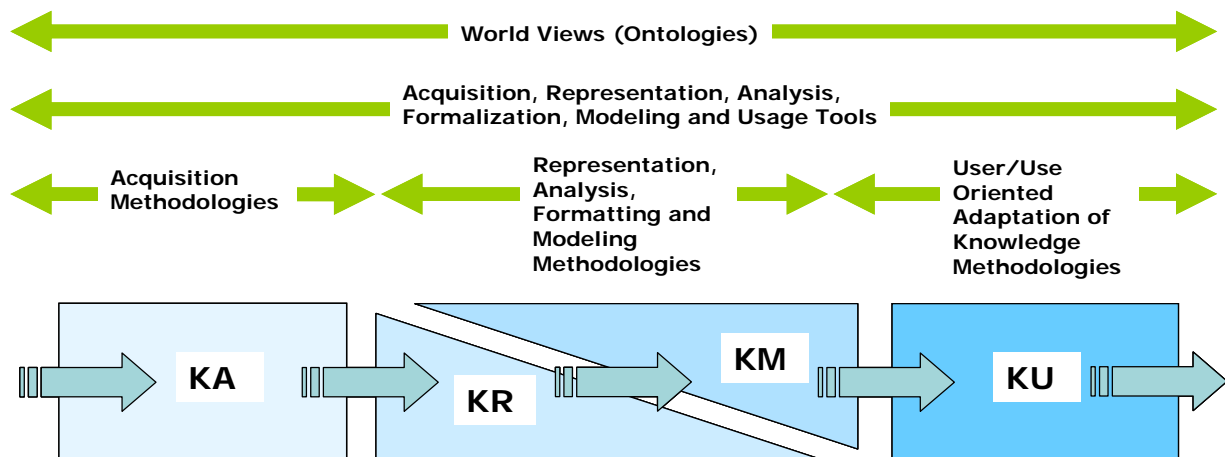
### 3 DCMF Components

In the previous chapter the DCMF process (DCMF-P) was discussed and a number of tools and methods were mentioned. Some of these tools and methods will be further discussed in the following chapters. Here the discussion will be about what the relationship is between the tools, methods and techniques, where they come into the knowledge development process, their function and how they contribute to the DCMF concept.

#### 3.1 Overview of Tools and Methods for Knowledge Development

In the knowledge development process that was discussed in the previous chapter the different phases were described together with some necessary tools, methods and techniques that provide certain functionality, see Figure 3-1. The kind of tools, methods and techniques that were discussed are:

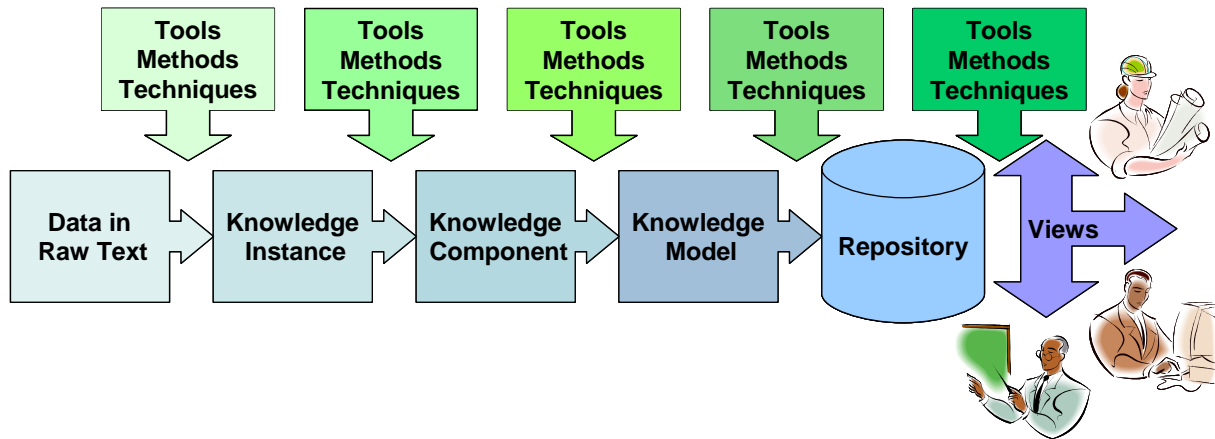
- *Methodologies for acquisition of data, information and knowledge:* to be used in the Knowledge Acquisition phase.
- *Methodologies for representation, analysis, formatting and modelling:* for use in the Knowledge Representation and Modelling phases.
- *Methodologies for user or use oriented adaptation of knowledge:* to be used in the Knowledge Use phase.
- *Tools for Knowledge Acquisition, Representation, Analysis, Formalisation, Modelling and Use:* these are the tools that are required by the methodologies that are used in the different phases of the DCMF process.
- *Ontologies:* Finally, spanning over them all are the ontologies that provide different world views and thus influence the knowledge development process from the acquisition, analysis, formatting, modelling to the use.



**Figure 3-1:** Overview of tools and methods in the different DCMF-P phases.

### 3.2 Data Flow in the DCMF-P

The flow of data, information and knowledge in the DCMF-P is affected by the application of different types of tools, methods and techniques at different steps. After each step there are one or more resulting products that are used as input to the next step or stored in a data base as is. These products may then be viewed by different users for different purposes.



**Figure 3-2:** The data flow in the DCMF process. From data in raw text several products are the result of applying different methods, techniques and tools to the resulting products of each step. The products then end up in the DCMF repository to be useable for different users and purposes.

If we view the DCMF-P from a data flow perspective, see Figure 3-2; data and information is first gathered by knowledge acquisition methodologies. These techniques and methodologies help ensure that the right and necessary data is gathered and that it is done so from the correct sources. When data in raw text is obtained it must be processed in some way so that it may be of use for the end user further on. The data is therefore processed by some knowledge analysis and formalisation methodologies that employ the appropriate tools. By using these tools the data is structured and focused according to some world view. Smaller sections of the structured data are called *knowledge instances* and the world view comes from the used ontology structure. The knowledge instances are useful for some purposes, but they are not reusable since they are specific to the used *data in raw text*. In order to get reusable components, modelling tools and methodologies must be applied, read for example chapter 6. To make sure that no information is misinterpreted the ontologies are necessary.

Modelling tools are applied to the knowledge instances in order to get abstract and reusable *knowledge components*. These components can then, with the aid of more methodologies and tools, be combined to form one or more *knowledge models*, such as MSMs. All of these products (Scenario Data, Knowledge Instances, Knowledge Components, Knowledge Models) can and should be stored in some DCMF repository for future use together with metadata that specifies how they have been produced, i.e. when, where, by whom, from what, using what tool, and so on. This metadata is necessary to ensure traceability.

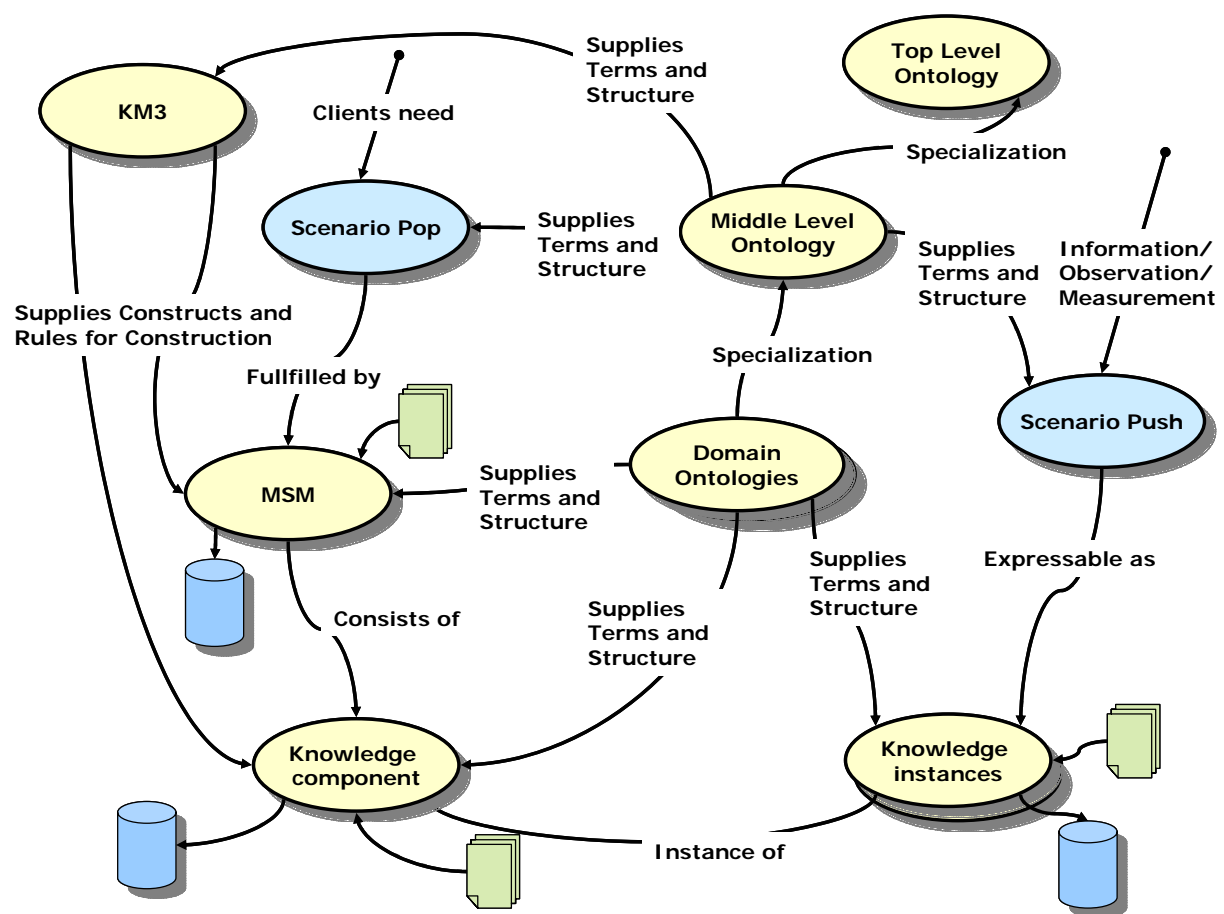
When the models, components and instances are to be used, some methodologies for composing and adapting the knowledge to the user's needs and purposes are also necessary. There should also be additional supporting tools for storage, modification, deletion and search of products in the repository and also for visualisation of knowledge that is adapted to the

user's needs. These tools and methodologies for visualising knowledge in a user adapted way are topics for future research.

### 3.3 An example of the data flow in the DCMF

To see how the data flow in the DCMF Process looks like and how the different tools and methods in the DCMF work together and come in, we will give an example of this. Three things can be seen in Figure 3-3:

- How the flow of data to information and then to knowledge is accomplished.
- How and where tools and methods come in and how they work together.
- How extraction of knowledge is supported.



**Figure 3-3:** Example of how the flow of data and information goes and where the tools, methods and techniques of the DCMF come in and interact.

We start our example from a scenario that will result in information being added to the knowledge base. Let us call it a Scenario Push, see the right most, blue circle in Figure 3-3. We are now supposing that the knowledge acquisition phase has resulted in one or more scenarios with data in “raw” text i.e. unstructured data. For the information in the scenario to be useful, it must be extracted and broken down into smaller more focused instances, i.e., we want to express the raw text as structured knowledge instances. The tools that are useful for

this are SPO, 5Ws or other similar tools (like KM3) for analysis and structuring. Domain ontologies are used to supply terms and structure to the scenario so that it may be formatted with the tools. When there are a set of knowledge instances, they should be stored in the repository and they should also have an accompanying description of when, where and by whom they were created, i.e. meta data about the instance should be created.

The information in the knowledge instances can then be modelled according to some purpose. To make them reusable we may also want to abstract the information so that it is not scenario specific. For example if an instance states that “the Swedish contingent contacts the French Commanding Officer, because of the find of hidden Kalashnikovs, in the Albanian Area of Interest”. It could be rewritten to “a contingent contacts the Commanding Officer in the circumstance of finding hidden weapons”. This change results in a knowledge component that is no longer scenario specific and also results in Rules of Engagement. These knowledge components may be the result of several knowledge instances and should also be accompanied by metadata to be traceable. They are also, like the knowledge instances, interesting to store in a repository. In the making of the knowledge components again the Ontologies play an important part. The ontologies help to ensure that the abstraction and generalisation from the knowledge instances hasn’t resulted in something erroneous. The constructs and rules for construction of the knowledge components are given by some appropriate modelling tool. An example of such a tool is our own Knowledge Meta Meta Model, KM3, see Chapter 6.

As can be seen in Figure 3-3 there is a connection between the KM3 and Ontologies. This connection is on two main levels. The top most level is how the concepts in the KM3 itself relate to an upper level ontology and what the concepts mean and how they are related to each other. The other level relates to the models that are modelled with the aid of the KM3. The information in those models can also be mapped to some ontology depending on what the domain of that information is.

If we look at the data flow from a user perspective where we wish to extract data from the knowledge base, it is important to formulate some kind of scenario or purpose that is derived by some need. This scenario has been called the “Scenario Pop” in the Figure 3-3, see the left most, blue circle. The scenario may then be fulfilled by one or more Mission Space Models. By using the ontologies to narrow down the field of interest it is easier to find the appropriate MSMs.

The Mission Space Models are the result of combining one or more knowledge components. The combining of the knowledge components is done with the aid of both the ontologies and the modelling tool. The meta-data that has been stored together with the MSMs, knowledge components, and knowledge instances is a good way to trace changes or background for a certain component. With this meta-data it is also easier to find the components that can be mapped to the same ontology and thus belong to the same world view.

### **3.4 Components of the DCMF**

Given the process that was described in the previous chapter, the overview of the kind of tools, methods and techniques that are necessary, given in Section 3.1, and the description of the data flow in the process, given in Section 3.2, it is time to introduce the components of the Defence Conceptual Modelling Framework. Some of these components (related to the earlier phases) have already been described in some of our previous reports [Moj03, Moj04], some of



them (related to the middle phases) are described in more detail in the following Chapters, and some of them (related to the usage phase) are the subject of future research. Keep in mind that the DCMF is still at a research level and that we may not be aware of all of its components yet.

- KA methodologies
  - KADS
  - Generic KADS
  - Desiree
- Ontologies
  - Military Specific Ontology Methodology (MiSO)
  - DCMF-ontology Framework (DCMF-O)
    - Upper Level Ontologies
    - Middle Level Ontologies
    - Lower (Domain) Level Ontologies
- Analysis and formatting tools
  - 5Ws
  - SPO
  - KM3
- Modelling tools
  - KM3
  - UML
  - IDEF1X
  - BPMN
- KU methodologies
  - Tools for composition of components
  - Tools for visualisation of knowledge
- The DCMF Repository
  - Management of the repository (include creation, modification, deletion etc of products)
  - Search for products in the repository
  - Enable semantic search
- Additional tools
  - Translation tools
  - User view
  - User access tools
  - etc

Methodologies for knowledge acquisition are used primarily in interview situations where a Subject Matter Expert is queried about something. These methodologies aim to get information without biasing the interviewee, without misinterpreting the given information and to structure an interview to get good results. Examples of methods that we have studied are KADS, Generic KADS and Desiree. We have written about these methods and our experiences with them in the report [Moj03]. There we also give recommendations that are the result of our studies and case studies.

Ontologies are used to give a conceptualisation of a particular domain including the terms, their semantics and pragmatics. They help us by supplying the terms and structure for the scenarios, knowledge instances, knowledge components and models. We have chosen a hierarchical ontology suite that follow from IEEE's view with a three level division of the

ontologies. An upper level ontology is limited to supply concepts that are meta, generic, abstract or philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas. The middle level Ontology describes concepts on a general domain level such as military, engineering, medicine, etc, where the domain ontology is a specialisation of the middle level and can for example be about weapons, sutures, transactions, etc. Our view on ontologies is discussed in the next chapter. Our ontology framework (DCMF-O) and our methodology to develop ontologies called the *Military Specific Ontology Methodology* (MiSO) are described in Chapter 5.

Analysis and formatting tools are used primarily in the knowledge representation phase of the DCMF process. They are used to structure raw data and give us the knowledge instances. At FOI we have primarily looked at three different methods and tools; SPO, 5Ws and KM3. SPO stands for Subject, Predicate and Object. This is a generic method that can be applied to almost any domain. In the semantic web framework, SPO is used to structure information. In essence, the written text is analysed and formalised by following the formula of SPO. Another method that we have applied is the 5Ws-format, WHO is doing WHAT, WHERE, WHEN and WHY. 5Ws is also used in writing in journalism and for telling a story in literature. The final tool that we have used is our own, the KM3, Knowledge Meta Meta Model. Read more about these techniques and methods in Chapter 7. In Chapter 9 an example of how we have applied these tools is presented.

The modelling tools are primarily used in the Knowledge Modelling and Use phases to create knowledge components and to visualise knowledge. KM3's primary use is in the KM phase, but classical modelling tools like UML come also very much in handy here. The KM3 is activity centric and a good tool for modelling activities. Other tools and notations that are very useful for modelling are UML, BPMN and IDEF1X. Each tool is appropriate for different purposes. Where UML takes a more object-oriented approach to the modelling of applications, BPMN takes a more process-oriented approach to modelling of systems, see Chapter 6 and [UML05, IDE05, BPM05].

Methodologies and tools for knowledge use include tools for composition of components and tools for visualisation of knowledge. We have not yet begun to look into these issues; this will be done in our future research. Other additional tools are for example translation tools, formatting tools, user rights and access administration tools, tools for description of metadata about the products that are stored in the DCMF Repository. Appropriate tools for description are [XML05, RDF05, OWL05].

The DCMF Repository is a central component in the DCMF. It is the one place where the products are stored for future use and reuse. Important tools are the ones used for the management of the data base, including creation, modification and deletion, etc., of products in the data base. Another important part of the repository is the functionality to search for available products. If possible this search should not only be keyword based but also enable semantic search in order to make it possible to search for concepts and relations between knowledge instances, components and models.

## 4 Ontology

Ontology has its origin in philosophy, and since then it has played a vital role in the realm of AI (Artificial Intelligence). But lately, with the development of the Semantic Web, ontologies have made their way into the fields of conceptual modelling, domain modelling and now even military modelling and simulation. Ontologies have gained popularity in the knowledge engineering field as can be seen from the works of Uschold and Gruninger [Usc96] for enterprise modelling, Gomez-Perez et al [Gom99] in chemical ontology, knowledge representation [Art96], [Gua98], information integration [Ber98]. A succinct review of the history and background of ontology has been carried out by Chira [Chi03]. Before we launch into a discussion on why we chose to use ontology for the Knowledge Modelling (KM) Phase of the DCMF Process, we would like to present a short overview of the different views of current ontology researchers. In the following sections we begin by a review of some prominent definitions of ontology (section 4.1), followed by a brief look at different types of ontologies (section 4.2). Thereafter, we also summarise some of the required features of any ontology as proposed by Gruber in section 4.3. Since the primary objective of the DCMF Project is to produce conceptual models – MSMs, we present a brief discussion on the relationships between Unified Modelling Language (UML), conceptual modelling domain and ontologies in section 4.4.

Capture, Analysis, Modelling and Representation of the domain knowledge from different military operations scenarios, is the objective of our DCMF Process. And this is essentially the purpose and objective of any good ontology design and development methodology. However, the design approach may differ, based on the type of ontology, the application or targeted use of the ontology, and to some extent the domain which is being modelled. Thus, the importance of selecting the appropriate ontology design methodology is paramount. Hence, we summarise the salient features of some of the many contemporary ontology design and building methodologies that we have surveyed in section 4.5. Thereafter, we motivate our choice of approach design with respect to the goals of our project in the next chapter.

### 4.1 What is an ontology?

The term ‘ontology’ has been defined in philosophy as:

“A branch of metaphysics concerned with identifying, in the most general terms, the kinds of things that actually exist. Thus, the ontological commitments of a philosophical position include both its explicit assertions and its implicit presuppositions about the existence of entities, substances or beings of particular kinds.”

Nicola Guarino [Gua98] has defined ontology from both philosophical and AI perspective as follows:

“*From philosophy:* Ontology is a language *independent* system of categories, the Ontology is a conceptualisation. *From AI perspective:* Ontology is a language *dependent* formal artefact.”

But one of the most popular definitions of ontology was given by Thomas Gruber:

“An ontology is an explicit specification of a conceptualisation.” [Gru93].

Studer and Benjamins [Stu98] have combined both Gruber and Borst’s definition as:

“Ontologies are explicit formal specification of a shared conceptualisation.”

Studer has explained the term as follows:

- *Explicit*: The type of concepts used, and the constraints on their use are explicitly defined.
- *Formal*: The ontology should be machine readable, which included natural language.
- *Shared*: Reflects the notion that ontology captures consensual knowledge, that is, it is not private to some individual but accepted by a group.
- *Conceptualisation*: Abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon.

Finally we quote Mike Uschold's [Usc98] definition of an ontology as:

"An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms. An ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between a number of agents. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing."

So we see that while Uschold allows an informal textual representation of the domain as an ontology Gruber restricts to formal conceptualisations.

After reviewing in depth, the different definitions of what an ontology is, we found uphold Uschold's definition as being closest to our purpose of the DCMF project. Thus, we define an ontology as:

"An ontology is an explicit formal conceptualisation of a shared understanding of the domain of interest including the vocabulary of terms, semantics as well as their pragmatics."

Ontologies have been described to be *functionally* similar to databases. A database schema represents the structure and integrity of the data elements in a single specific domain of application. An ontology is conceptualisations of a domain of interest and as such has common functions as do data models. The difference lies in the domain they cover, database schema are task specific and implementation oriented [Spy02] while ontologies are generic and *task independent*. Also the languages for defining ontologies are semantically and syntactically richer than those for database schema description. At the same time, ontologies are formed by consensus terminology because most ontology researchers adhere to the common principles of ontological commitment, coherence, and clarity as proposed by Gruber [Gru91]. We discuss more on these aspects in the following sections.

## 4.2 Types of ontologies

Guarino proposes a classification of ontologies under three headings, as follows [Gua97]:

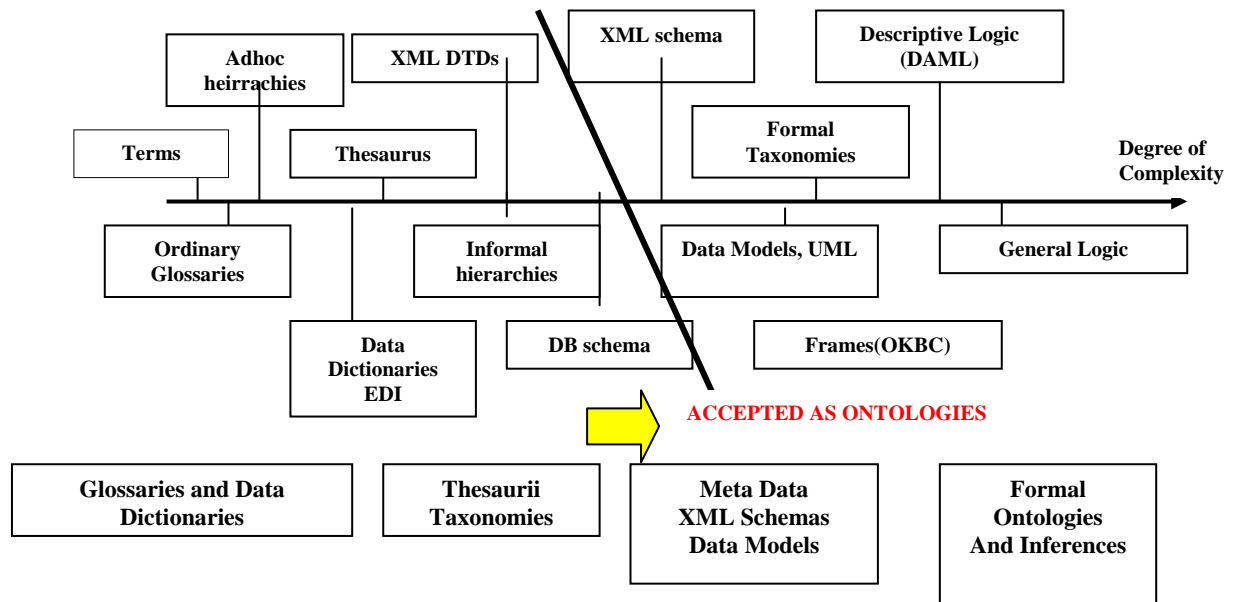
1. By the level of detail
  - a. Reference (off-line) ontologies
  - b. Shareable (on-line) ontologies
2. By the level of dependence of a particular task or point of view
  - a. Top-level ontologies

- b. Domain ontologies
  - c. Task ontologies
  - d. Application ontologies
3. Representation ontologies

We do not go into detailed discussion on the above types of details as they are not highly relevant to our current discussion. We shall see more about the top level, domain and task ontologies in section 4.3.1, when we discuss the design philosophy as proposed by Guarino.

Uschold and Gruninger in [Usc96] have discussed in detail the principles, methods and characteristics of ontologies. They have classified ontologies depending upon their *formality* and *complexity* as a continuum as belonging to the following major categories, illustrated in figure below:

- *Highly Informal*: Which is expressed loosely in natural language
- *Semi Informal*: Expressed in a restricted and structure form of natural language.
- *Semi Formal*: Expressed in artificially formally defined language.
- *Rigorously Formal*: Clearly defined terms with semantics, theorems and proofs.



**Figure 4-1:** Types of Ontologies [Gru02]

In figure 4-1 above, the degree of complexity and formality increases in the direction of the arrow. All those categories to the right of the arrow are those which are widely accepted as being an ontology. Though there are discussions ongoing to determine if glossary, taxonomies and thesaurus can be accepted as loosely formulated knowledge bases. In the DCMF Project, we aim to keep track of the ongoing discussion.

The different types of knowledge representation formalisms represented in figure 4-1 may be explained as:

- *Glossaries and Data Dictionaries*: In this category we have simple look up classifications, including WordNet.

- *Thesaurii and Taxonomies:* Thesaurii are a step more evolved, where they try to relate similar words together like synonyms, homonyms, hyponyms, etc. Taxonomies provide linear classifications of language or type.
- *Meta Data and Data models:* Since up to taxonomies, the knowledge modelling deals mainly with only the syntactic or meaning of only the particular term, most researchers do not accept anything under the level of meta data analysis as being a form of ontology. Meta data, traditional data models, UML models, actually fall in the boundary condition, where they are accepted as semi-formal or semi-informal ontologies. This is because these are not enriched in the pragmatics, or semantics to a high degree. Axiomatic relationships to support inference are also missing.
- *Formal Ontologies and Inferences:* these are the rigidly formal and fully machine readable ontology specification like those specified using KIF<sup>4</sup>, OWL<sup>5</sup>, DL<sup>6</sup> and other logic languages.

### 4.3 Ontology Design Criteria

Gruber [Gru93] has formulated some criteria for design of formal ontologies mostly for artificial intelligence purposes as discussed below:

- *Clarity:* Ontology should be able to effectively communicate its intended meaning to its users.
- *Coherence:* Ontology should support inferences that are consistent with its definitions.
- *Extendibility:* Ontology should be designed to anticipate the uses of shared vocabulary. One should be able to define new terms based on the existing definitions.
- *Minimal encoding bias:* According to Gruber, the conceptualisation should be specified at the knowledge level without depending upon any symbol or language encoding.
- *Minimal ontological commitment:* Finally, Gruber recommends that ontology should not restrict the domain being modelled, allowing the users the freedom to specialise and instantiate the ontology as required.

#### 4.3.1 Design Approach

Methodology for defining ontology includes identifying the scope and use for the ontology. Then the domain knowledge is to be classified and concepts identified. Ontology is defined as a set of classes arranged in a hierarchy or taxonomy, where real world concepts are modelled as classes, their characteristics as attributes and inter-object relationships as relationships, properties or axioms.

Guarino [Gua97, Gua98] has defined ontologies as logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualisation of the world. He has further classified ontologies, depending upon their generality as (figure 4.2):

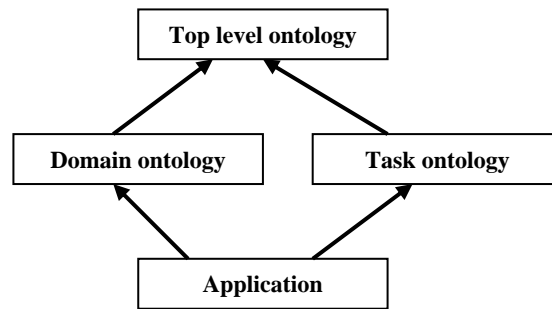
- *Top Level Ontologies:* Top-level ontologies to describe general concepts like time, space, matter, and event that are independent of domain or a particular problem.
- *Domain Ontologies and Task Ontologies:* Are described to be ontologies pertaining to a specific domain or task.
- *Application Ontologies:* Describe concepts that depend upon both a domain and a particular task, usually being specialisations of both ontologies.

---

<sup>4</sup> KIF - Knowledge Interchange Format

<sup>5</sup> OWL - Web Ontology Language, proposed by W3C

<sup>6</sup> DL - Description Logic



**Figure 4-2:** Types of Ontology proposed by Guarino

Guarino proposes a bottom up approach to designing that is he suggests that we identify the most specialised concepts, needed in the application ontology, then the domain ontology and task ontology. Finally, the generic concepts are to be abstracted in to the top level ontology. In essence, his suggested approach is valid in those cases when the ontology is to be designed from scratch. This does not take in to consideration other previously existing ontologies or knowledge bases.

#### **4.4 Conceptual Modelling, UML and Ontologies**

So far we have reviewed several contemporary ontology design methodologies, and types of ontologies, possible levels of formalisation etc. But, the DCMF project is primarily about conceptual modelling of military operations to allow an effective, reusable means of capturing real world domain knowledge in some machine understandable or process-able format. We give a recap of some basic definitions from [Joh97] as:

“A *natural language* is an ordinary hereditary language, spoken by a group of individuals as their native tongue. An example of a natural language is English.”

“Then we have a set of *artificial languages* which are not contrived by humans. A subclass of the artificial languages is that of *machine languages* which as the name implies, are used by a machine to code letters, numbers, instructions, and storage locations in such a way that a computer does not require any translation in order to function according to its coded instructions. Pascal, C++, and Prolog are languages that are accompanied by rules for concatenating the symbols into sequences. In computer science the ability of various formal languages to reflect the subtleties in descriptions of natural language is carefully studied, especially in the field of conceptual modelling. ”

“*Syntactics*, or syntax, is concerned with the way sentences are constructed from smaller parts, such as words and phrases.”

“*Semantics* is the study of meaning in language, i.e., the study of the relationship between linguistic expressions and reality. This should be compared to Syntactics, which is only concerned with the form of expressions in a language.”

“*Pragmatics* is the third part of semiotics, and it concerns the actual use of a language by its speakers and listeners. In order to understand sentences, a syntactical and semantically analysis is often insufficient. An example is the

question “Do you have the time, please?” A “Yes” answer would be correct from a syntactical as well as a semantically viewpoint, but not pragmatically correct, since we know that what the person wants as a reply is an indication of the current time, e.g., “11:56” or “Around noon”. Pragmatics concerns the purpose and effects of uttering sentences. When someone utters a sentence that person may have the intention to convey some piece of information, e.g., to tell the time.”

Thus, we can summarise a definition of conceptual models in the context of computer information systems as:

*Conceptual models are abstractions of a real world domain of discourse. They are intended to capture the semantics, pragmatics and to an extent the syntactic of the domain being modelled.*

It is to be noted that the term ‘conceptual model’ itself has different semantics in different domains and context. For example, within the information system modelling domain, we refer to conceptual models for a high level abstraction and the representation could very well be a simple UML class diagram. It is common to refer to ‘UML Conceptual Models’ to imply a conceptual model of a domain represented using UML class diagram notations. Whereas, the term ‘conceptual models’ specifically in context of the CMMS process refers to re-usable Mission Space Models to be represented in a defined representation format. In this chapter, when we refer to ‘conceptual models’, we mean conceptual models in its broader perspective, i.e. within the information systems domain.

As such, conceptual models and ontologies are closely related. One of the accepted ways of expressing conceptual models is UML (Unified Modelling Language). UML is being used for far more than simply conceptual modelling. UML class diagrams (conceptual models) can be categorised as *semi-formal* ontologies themselves. (Semi-formal because they are not directly machine-readable) However, tools are being developed that enable automatic transformation from UML class diagrams to ontology formalisms like DAML<sup>7</sup>, OWL etc.

Advantages of UML as an ontology modelling language has been proposed by Cranefield [Cra99], Baklawski and Hart in [Bac01] as:

- It has a growing user audience in the software domain for object modelling languages and other information system design. In our case, those attempting to integrate business contracts with existing business management applications are more likely to be familiar with UML than other knowledge representation languages like KIF.
- The graphical notation for UML is easy to comprehend and use and is suitable for human-to-human knowledge transfer.
- UML can be extended to suit the need of ontology definitions.
- Object Constraint Language allows expression of rules and constraints.

Moreover, UML models can be translated into other ontology languages like RDFS<sup>8</sup> or DAML or even in to object oriented database systems. Cranefield in [Cra01] has presented

---

<sup>7</sup> DARPA - Darpa Agent Markup Language. More information can be accessed at [www.daml.org](http://www.daml.org) last accessed on 12<sup>th</sup> Oct 2005

<sup>8</sup> RDFS - Resource Description Framework Schema



methodologies to transform UML ontology models in to RDF<sup>9</sup> and to generate Java classes from UML using XSLT<sup>10</sup>.

Backlawski [Bac01] has presented some mappings for translating in between DAML and UML concepts and from UML to DAML, as illustrated in the table 4.1 below,

DAML Concept	Similar UML Concepts
Ontology	Package
Class	Class
As Sets (disjoint, union)	Difficult to represent
Hierarchy	Class Generalisation Relations
Property	Aspects of Attributes, Associations and Classes
Restriction	Constrain Association ends, including multiplicity and roles. Implicitly as class containing the attribute
Data Types	Data Types
Instances and Values	Object Instances and Attribute Values

**Table 4-1:** Mapping between DAML and UML

Gasevic [Gas04] have presented an ongoing work for UMLtoOWL converter tool, which is now available for testing [Gas05]. One of the future works for the DCMF project can be to test and evaluate the usability of such tools for aiding for analysis process. In a similar approach, within the OMG<sup>11</sup> standardisation group itself, there is an ongoing process for adopting an Ontology Definition Meta-model (ODM) [ODM05] that shall propose the mapping between UML 2.0 and OWL (Web Ontology Language). The ODM proposes a set of profiles using UML 2.0, MOF, and the MDA for expressing conceptual models which can then be directly transformed to OWL and vice versa. We propose to adopt and implement the ODM specification, once it has been reviewed and standardised. For the moment, we have taken inspiration from the proposal and based our current case study on similar lines.

## 4.5 Ontology Design Methodologies

In this section we survey some of the prominent ontology design and development methodologies. As this process is vital to the overall DCMF Process, we have focused in-depth on this analysis.

We shall first present short summaries and salient features of the different ontology design approaches reviewed. Thereafter, we shall discuss the criteria or requirements based on which we analysed the suitability or applicability of each of the surveyed methodologies for the purpose of the DCMF project in the succeeding chapter 5.

### 4.5.1 Method I: Proposed by Uschold and King

Uschold and King [Usc95] provide guidelines for ontology designing based on their experiences in designing the Enterprise Ontology<sup>12</sup> and the TOVE<sup>13</sup> project, which may be

<sup>9</sup> RDF - Resource Description Framework

<sup>10</sup> XSLT - eXtended StyLesheet Transformation

<sup>11</sup> OMG - Object Management Group

<sup>12</sup> The Enterprise Ontology ,developed by the University of Edinburgh and its partners. More information at <http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html> (last accessed on 14th Oct 2005)

<sup>13</sup> TOVE - Toronto Virtual Enterprise Ontology project .

summarised in the following five phases (we include comments relevant to our DCMF Project within parentheses for the rest of this chapter) [TOV05]:

1. *Identify Purpose*: Why are the ontology being built and what its intended use is, and who the targeted users are. (In our case, it should be 'for establishing a generic and re-usable knowledge base for military operations'. The targeted users would be the modelling and simulation system designers, their users, etc.)
2. *Building the ontology*: In this phase the actual design of the ontology is suggested following the given steps below:
  - a. *Ontology Capture*: A middle-out approach for identifying the most important concepts rather than the most general concepts or the most particular ones, followed by generalisation and specialisation to obtain the remainder of the hierarchy. (This approach has been adopted by us in the current project, for several reasons as shall be discussed later.)
  - b. *Scoping*: Identification of the key concepts and relationships in the domain of interest
  - c. *Production*: Precise unambiguous text definitions for such concepts and definitions.
  - d. *Identification*: Terms to refer to such concepts and relations as described above.
  - e. *Consensus*: Agreeing on all of the above. At least a consensus between the developers is required primarily and later on consensus is obtained on a broader scope by publishing the ontology and getting reviews from other users and ontology developers.
3. *Coding*: Explicitly representing the knowledge/conceptualisation captured in the previous phase using the chosen formal ontology representation language. In this case, it would encompass the design in UML conceptual models, and then the implementation in OWL (Web Ontology Language) format.
4. *Integrating existing ontologies*: Merge or map to other existing ontologies either during the ontology capture phase or at the coding phase. Here we have mapping to the generic SUMO<sup>14</sup> ontology, as well as to the specific domain/application ontologies we have developed like the Swedish Defence Organisation Ontology, Weapons Ontology, etc.
5. *Documentation*: Guidelines are to be established for proper design and documentation of the ontology design and development the same way any traditional database system is documented via Functional specification, Database design guide, Test plan etc.

#### 4.5.2 Method II: Proposed by Gruninger and Fox

Gruninger and Fox [Gru95] propose a more prosaic design approach as compared to Uschold and King. They have designed a more formal and extensive ontology, the TOVE ontology [TOV05] (The TOVE is a set of formal ontologies for different aspects of the business enterprise like the Resource Ontology, Time ontology, etc.), using the following outlined method:

1. *Capture of motivating scenarios*: These are the use case scenarios or problems that arise from a given situation. In the case of ontologies, these scenarios motivate and guide the design of the ontology. Any ontology design should start by describing one or more such motivating scenarios and the intended set of solutions for the problems

---

<sup>14</sup> SUMO -Suggested Upper Merged Ontology.

presented in the scenarios. (This step suits us more, since our basic input for the domain knowledge is from various sources like domain experts, recorded media, reported journals, log records etc.)

2. *Formulation of informal competency questions:* The competency questions are based on the motivating scenarios and are visualised as expressive requirements that are represented as questions. The ontology must be able to represent these questions using its terminology and characterise its answers using the axioms and definitions. These competency questions are also a means to evaluate the ontological commitment of the designed ontology. Detailed examples and explanations of such competency questions may be referred to in [Gru94]. (However, we have not focused much in this step at the current juncture, as we do not know the exact parameters we desire for our proposed ontology. But it's possible that we return to this phase at a later stage, once we have explored and established the precise uses, purpose and requirements of our proposed ontology.)
3. *Specification of the terminology of the ontology within a formal language:*
  - Getting the informal ontology extract set of terms from the competency questions and the scenario which act as a basis for specifying the terminology in a formal language.
  - Specification of formal ontology: Specifying the terms in chosen formal ontology representation language.
4. *Formulation of formal competency questions:* Using the terminology now defined in the ontology
5. *Specification of axiom:* Definition and specification of axioms for the terms in the ontology within the formal ontology language. Gruninger and Fox state that simply defining a set of objects alone or proposing a set of ground terms does not constitute ontology. Axioms must be provided to define the semantics of these terms.
6. *Completeness:* Establish conditions for characterising the completeness of the ontology

Thus, we find that Gruninger and Fox also propose a strict and inflexible method.

#### 4.5.3 Method III: Proposed by Noy and McGuinness

A practical handbook like design approach has been provided by Noy and McGuinness [Noy01]. However this approach is more like a user manual for a ontology to be designed specifically using the Protégé<sup>15</sup> ontology editor than a generic ontology design methodology. In simple steps they illustrate the process of capturing the concepts, the slots and the role restrictions. But, on analysis, we see that their basic design methodology is similar to that proposed by Gruninger or Uschold.

Noy and McGuinness have proposed a knowledge engineering method for building ontologies as discussed earlier. They advocate the iterative and refinement process and have proposed three fundamental rules for the ontology developer to help him in design decision process:

- “There is no one correct way to model a domain. There are always viable alternatives”
- “Ontology development is an iterative process”

---

<sup>15</sup> Protege 3.0 ontology editor tool (Open source), Stanford University. (<http://protege.stanford.edu/>). Last accessed 14<sup>th</sup> Oct 2005

- “Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. They are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain”

They provide a step by step instruction for the user to design the ontology in the Protégé editor. They directly implement the ontology in to classes (concepts), facets (constraints) and properties (relationships). For obvious reasons, we do not reproduce the steps illustrated here in this report. But the interested reader may refer to Noy and McGuinness. (This method is highly informal, in the sense that we cannot trace the design of the ontology from its initial knowledge acquisition phase.)

#### 4.5.4 Method IV: METHONTOLOGY

Another design methodology that we would like to review here is that of METHONTOLOGY [Fer97, Gom98, Fer99] that is used for building ontologies from scratch or from other existing ontologies or by a process of re-engineering. Till now, the ones we have reviewed basically deal with designing ontologies from the scratch, that is no previous versions or knowledge base or data model exists. Also, they are mostly in a linear single design that is, once designed; they are not specific on how the ontology can be further modified or evolved.

Several of the steps proposed by Gomez-Perez et al are similar to those of Uschold or to Gruninger. But notable difference is their stress of the *evaluation* and *documentation* steps. METHONTOLOGY supports an *evolving ontology life cycle* unlike the others that support top down, bottom up, or middle out approaches. (That means that the ontology can be modified and can grow constantly as befits our requirement too.)

Figure 4-3 and the list below summarises the different phases of the ontology life cycle, which we discuss in detail below.

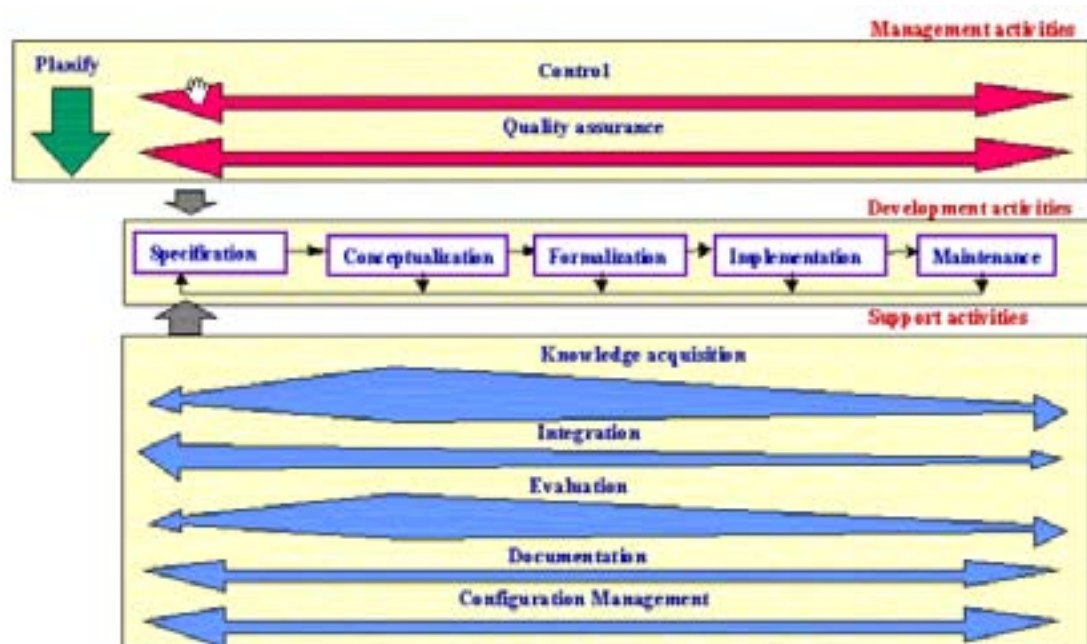


Figure 4-3: METHONTOLOGY ontology development life cycle

Gomez-Perez et al advocate that:

1. *Planify*: one should plan the entire development process like the tasks, time and resource allocation etc.
2. *Specification*: Just as one never starts a trip without knowing the destination and purpose for the travel, one should never start the ontology design and development process without establishing the *purpose* and *scope* of the ontology. Gomez Perez propose writing down formal or informal questions and answers to establish the purpose and scope which is similar to the competency questions phase as recommended by Gruninger and Uschold.
3. *Acquire Existing Knowledge*: They also advocate the use of existing knowledge bases and knowledge acquisition using techniques as proposed by Gruninger and Uschold [Usc96]. This phase is vital if one has to acquire ample knowledge about a domain. This also ensures that we get a degree of consensus on the domain being captured.
4. *Conceptualise*: Following Knowledge acquisition, we need to *conceptualise* the knowledge using some conceptual knowledge modelling technique as also proposed by Gomez-Perez in [Gom96].
5. *Formalise*: The next step recommended can be quoted as “To transform the conceptual model into a formal or semi-compatible model, you need to formalise it using frame-oriented or description logic representation systems.”
6. *Integration*: Ontologies are intended to be reused therefore Gomez-Perez suggest the *integration* of relevant ontologies as possible. Farquhar and colleagues [Far95] have identified four kinds of relationships between ontologies that have been integrated: inclusion, polymorphic refinement, circular dependencies and restrictions. Other ontology mapping or integration methodologies have been investigated by Bergmaschi and Castano.
7. *Machine Readable*: To make the ontology ‘machine-readable’ we need to select the formal machine process able implementation language. (we chose XML based and current WWW recommended language for ontologies, OWL)
8. *Evaluation*: Gomez-Perez now stress the need to *evaluate* the ontology designed, so that to rule out any erroneous definitions and discrepancies in the ontology. They have proposed techniques for evaluating ontologies. [Gom95]
9. *Documentation*: Thereafter they recommend that proper *documentation* is vital as in any software development project, not only for easy reusability, modification, but also for configuration management, change traceability etc.
10. *Maintenance*: Finally, they recommend that ontology once designed and developed cannot be forgotten, it needs to be constantly *maintained*.

We have looked at how the METHONTOLOGY process can aid the design of ontology building from scratch or through re-engineering from existing Knowledge Bases.

#### 4.5.5 Method V: UPON Methodology

The final ontology design methodology which we review here is the UPON (Unified Process for ONtology building) proposed by A. Nicola and M. Missikoff [Nic05]. Their process builds on the accepted Unified Process and uses UML. Thus this design methodology is not unfamiliar to designers already familiar with the unified process for designing. In figure 4.4, below we see a summarised overview of the different phases in the ontology building process. UPON makes use of domain experts and knowledge experts similar to the way DCMF uses the SME and ontology expert in the CMMS process.

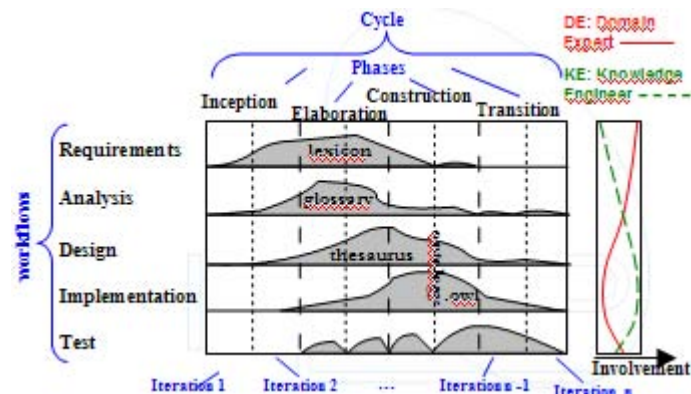


Figure 4-4: UPON ontology building methodology

Some of the other salient features of UPON are:

- *Use Case Driven:* That is the first input is the setting up of scenarios and use cases from the domain of discourse.
- *Iterative:* The different phase of the design methodology is followed iteratively, starting from crude details and refining successively to get specific aspects of the domain.
- *Incremental:* The ontology can grow in steps and is flexible to accommodate new information gathered from new scenarios.

The design methodology closely follows the unified process and has the following phases:

1. *Inception Phase.* Requirement capturing and modelling the use cases.
2. *Elaboration Phase.* Analysis of requirements and fundamental concepts are identified and loosely captured.
3. *Construction Phase.* Based on the loosely identified concepts a skeleton for the ontology may be designed. Successive iterations of the first three phases, will lead to refinement and a more stable version of the ontology ultimately reached.
4. *Transition Phase.* The ontology is subjected to rigorous testing, documentation and finally released for public use.

They have also proposed detailed strategies for requirement workflow, analysis workflow, capturing and modelling phase, etc.

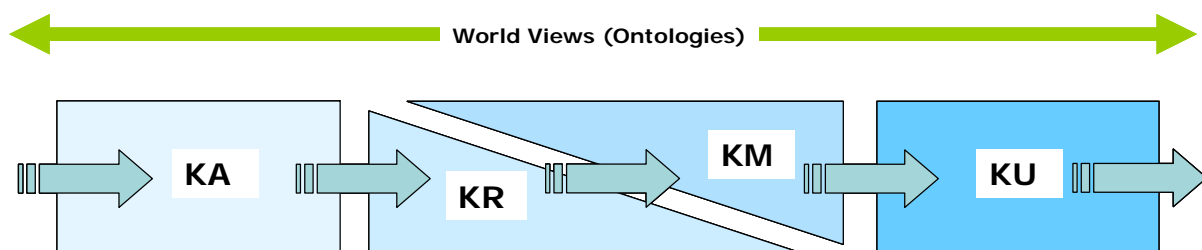
## 5 Military Specific Ontology (MiSO) - Methodology and DCMF-O Framework

In the previous chapter, we reviewed some of the contemporary ontology building methodologies. We also looked at the relationship between conceptual models, UML class diagrams and Ontologies. After this extensive related research study and analysis, we present the proposed design methodology and philosophy for the DCMF project.

We begin by positioning the role of the proposed DCMF-O (Defence Conceptual Modelling Framework-Ontology), in the context of the overall DCMF Process in section 5.1. Thereafter, we focus only on the ontology development methodology for the DCMF project, which we propose to name as Military Specific Ontology Development methodology, here with referred to as MiSO methodology. We begin by listing the requirements and criteria to be fulfilled by the ontology in the DCMF Process in section 5.2. Based on these criteria, in section 5.3, we propose our MiSO methodology as a combination as well as enhancements of the ontology design methodologies reviewed in chapter 4. After the discussion on *how* the ontology is to be designed, we move on to the *architecture* or type of ontology structure to be designed, in section 5.4. Thereafter, we give overviews of the major constituents of the DCMF-O, namely SUMO, JC3IEDM,<sup>16</sup> and the set of domain ontologies.

### 5.1 MiSO in context of the overall DCMF Process

Here we discuss on the uses, the roles, and positioning of the ontology design and development process in context of the DCMF Process



**Figure 5-1:** The role of ontologies in the overall process

There are two *different* processes which flow through the current DCMF project:

1. The overall DCMF Process, which has several phases, individual sub processes etc as has been discussed in chapter 3. Also, we shall see a detailed illustration of this process in chapter 9 where we present a case scenario analysis through the proposed DCMF Process.
2. The process and design methodology proposed for the building of ontological knowledge base- MiSO.

<sup>16</sup> JC3IEDM - Joint Consultation Command & Control Information Exchange Data Model , which is developed by the MIP DMWG in association with NATO Data Administration Group (NDAG)



In this chapter, we focus on the *second* of these processes. It is to be noted that these two are deeply intertwined as well. That is, they are not independent processes, but more a co-existing processes. The ontology building process goes on parallel along with the DCMF Process. Again, the ontology built using the MiSO, has a vital role to play in the overall DCMF process. Some of the phases or the design philosophy overlap with or are similar to phases described in the DCMF process and in the KM3 process.

## 5.2 Requirement Criteria for DCMF-O

Here we discuss the different factors which influenced our choice for the appropriate design methodology most suited to our project goals. Based on the overall objectives and goals for the DCMF project, we can list some of the main functional requirements and design criteria for the proposed ontological knowledge base as follows:

- *Handle unstructured information as input sources:* The DCMF aims to capture and model military operations related domain knowledge from a variety of sources. Sometimes, it could be Subject Matter Experts and deep interviews. It could be based on field reports sent in by military organisations. It could be other intelligence systems or information systems; it could be a video clipping or news coverage. Thus, the first level of domain interaction is basically a set of *scenarios*. These *scenarios* need to be examined and we treat them as the *raw, unstructured data input* to our DCMF Process.
- *Flexibility and Adaptability:* We need a flexible design approach, since we would proceed by examining each available scenario. A given scenario may provide us with certain information, and another scenario could provide some more completing set of information. Thus, the ontology needs to provide input to the scenario analysis in one hand, and on the other hand, also needs to grow (take in) new information.
- *Reuse existing knowledge base and information:* A lot of existing information pertaining to military operations or related fields has already been captured and modelled in different formats. Existing knowledge bases and ontologies should be reused and integrated. Several of the military operations related knowledge have also been standardised, such as the MIP's JC3IEDM. Therefore, we should adopt a strategy that we begin from the existing standards and documented knowledge, and thereafter try and discover the additional or surrounding knowledge. Similarly, several domain ontologies like the MILO, Weapons of Mass Destruction, and Terrorist ontology have been developed by different groups of researchers. These could be readily adapted to be a part of our proposed DCMF-O.
- *Traceability:* We should be able to trace the original source of our information.
- *Machine-Readable:* We should be able to have a machine readable version as well as a human understandable format with little or no additional effort.
- *Interoperability:* We should be able to interoperate/integrate with our M&S applications. The Simulation Models build on the conceptual models, which in turn are based on the proposed ontological knowledge base. Therefore, the simulation models and the DCMF-O should be able to interoperate with each other not only at simple data exchange level, but also at higher conceptual, strategic and tactical level.
- *Rapid and Easy Development:* It should be a fast and easy methodology, since most of our designers and analysts would not be ontology experts.
- *Reusability for Multiple Applications:* We should be able to use the ontology for a variety of applications, in the initial phases as a look up dictionary or lexicon to help us in analysis raw data (in the Natural Language scenario analysis). Thereafter, we



should be able to use the established knowledge base for checking compliance to specified military operating procedures. We could even use it for a case based reasoning tool to suggest probabilistic course of action in event of similar military situations. Details are discussed in the section of Uses on DCMF Ontology.

- *Formal as well as Informal Representation:* The proposed DCMF-O is aimed to be used by different target users. On the one hand, it should be meaningful and give valuable input to high level decision makers and commanders in their strategic planning of operations. On the other hand, the Military Simulations designer and developer should get enough detailed information, to enable him to implement the same MSM directly in to a simulation model. Thus, there is a need for the DCMF-O to be represented as *informal easy to understand UML class diagrams* as well as *formal*, machine readable, OWL representations.
- *Credibility, Authenticity to be verifiable, validated:* Given the nature of this domain, the requirement for the DCMF-O to be credible, authentic is self-evidentiary. Thus, the DCMF-O concepts should also model the degree of credibility, authenticity of source etc. In short, the ontology should be readily verifiable and validated.
- *Action Centric Model:* Unlike other domains for ontology building, which are ‘object-oriented’ or centred around the concept of ‘objects’ and things that revolve around them, the military operations domain is *action-centric*. That is, the central concept of interest to us is the *action or activity* that is occurring. The supporting concepts answer questions like- *why is this action happening? Who is doing it? What is needed to counteract this?* etc
- *Static and Dynamic Aspects:* This requirement is related to the one above, action centric. Any military operation scenario involves objects which have some constant attributes, for example, a jeep has always four wheels; and there are some dynamic variables, for example, the rifle could be currently loaded with only two cartridges. While this temporal distinction of characteristics is not that vital for object centric modelling approaches, the same assumes important dimensions in this case where action- centric modelling is required.
- *Consistency:* Each military scenario could be analysed by different subject matter experts as well as different ontology experts. However, the end result should be the same no matter who performs this transformation process. In other words, the ontology should be so designed along with required mapping rules and guidelines that the analysed output, MSM derived should always be consistent as well as repeatable.
- *Composability:* Each scenario can be analysed and saved as individual MSMs. The idea is that several such MSMs could be put together in a logical sequence to compose a bigger, broader picture of an over arching MSM.
- *Produceability:* As said earlier, the DCMF-O should be transformable into MSMs. This requirement is related to the machine readability requirement.

### 5.3 MiSO Development Methodology

On analysis of the above requirements and the guidelines for designing ontologies, we propose:

- A multi-layered architecture for designing the ontology is better than a single monolithic structure. This allows faster development. Easier to maintain and modify parts of the ontology. Also it encourages re-usability by others for different applications. Again this structure offers maximum flexibility in design. We may extend the current architecture both horizontally as well as vertically (by horizontal,

we mean that we could add more concepts and relationships to the same layer, and by vertical we mean that we may choose to add a fourth or fifth layer as we may need in the future).

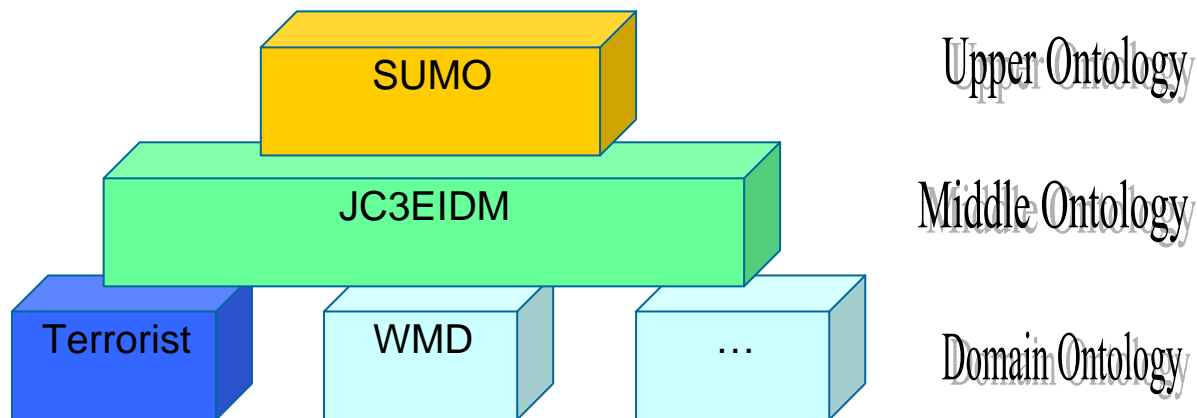
- A middle-out design approach is best suited in our case, as shall be discussed in section 5.4, where we present the different layers of our multi-layered architecture. This approach also supports rapid, easy development.
- We propose to adopt and enhance the JC3IEDM as the middle layer. Since the JC3IEDM itself is aimed at providing easy interoperability between different information systems, the proposed DCMF-O should also promote interoperability to the same if not greater degree.
- We propose to model the domain of interest in two steps.
  - In the first step we capture the domain knowledge as a series of UML conceptual models, thereby conforming to the *semi-formal* ontology type. These are useful for human understand ability, promote rapid re-use amongst users, and can be readily transformed to *formal ontologies*. However, we do not focus on the UML conceptual model representation in the rest of this chapter. The SUMO ontology is available online as UML models, and the JC3IEDM exists as IDEF0 (see Fig. 5-4) graphical models, which is another representation language for conceptual modelling. From this point onwards, we refer to the ontology knowledge as the formal knowledge modelling approach. It is understood that the semi formal UML/conceptual modelling phase is inbuilt.
  - We propose to model the captured domain knowledge using Web Ontology Language (OWL), with DL (Description Logic) axioms. This supports full inferencing and is machine readable.
- We adopt the informal knowledge capture phases as suggested by Uschold and King for formulating the basic knowledge capture activities. From Uschold we also adopt the idea of investigating and capturing the domain first in natural language free text as scenarios. This step is also motivated by the UPON methodology.
- We also choose the competency questions phase as suggested by Gruninger and Fox, but we would use these for determining the minimum level of required information in a scenario to be modelled in the ontology. These competency questions, based on a revised version of the 5Ws analysis process, are a useful aid in extracting the explicit and implicit knowledge from a scenario description. Details are discussed in the Chapter 10 where we present some of our experiences in the actual design of the prototype implementation of the JC3IEDM in to our DCMF Ontology.
- We propose to follow parts of METHONTOLOGY for its evolving life cycle, like its more structured goals of planning, modelling, development, documentation and maintenance.
- Given the easy familiarity with the unified process as proposed by the UPON methodology, we adopt parts of their design philosophy as well. Their identification of the roles and responsibilities of the knowledge engineer and the domain expert in the different phases of ontology building has provided us with useful insights. In fact, UPON comes very close to the requirement of our DCMF Process. They also follow a similar process to our Knowledge Acquisition, Knowledge Representation, Knowledge Modelling and Knowledge use phases.

Thus, we decided to take the best from each reviewed ontology design methodology and blend it to derive our own MiSO methodology. It is a combination of practical design guidelines, approaches and also includes phases from software project management

In the next section, we present the proposed architecture for our ontology. We have opted to model the DCMF-O as a multi-tiered ontology based on the suggestions made by Guarino. However, we choose to develop the ontology suite moving *middle-out* as suggested by METHONTOLOGY. The reason for this choice being more practical oriented. Since we chose to adopt the SUMO as an upper generic ontology, it already exists as a well established and standardised ontology. Therefore we need not develop that further. Moreover, we have a sound starting point in the JC3IEDM data model (as shall be discussed in section 5.4.3) for our middle layer of the domain ontology.

## 5.4 DCMF-O Architecture

We adopt Guarino's strategy of differentiating generic top level ontology definitions from specific domain or task related concepts (Fig 4-2).



**Figure 5-2:** Proposed DCMF-O Architecture

The proposed architecture comprises of three vertically aligned layers of ontologies:

- *Upper Ontology Layer:* Suggested Upper Merged Ontology (SUMO) from IEEE has been used as the top level generic conceptual layer. This layer has been used to tie down the domain oriented concepts into more abstract real world concepts like *entity*, *time*, *space*, etc. A short summary of this SUMO is discussed in section 5.4.2
- *Middle Ontology Layer:* The middle layer is intended to provide little more specialised concepts than the top, generic SUMO level. But still it encompasses a major domain area. In our case, this level should cover the entire range of topics included under the military operations and modelling domain. However, we have chosen to adopt the established standard JC3IEDM as a starting point for this middle level. We discuss the JC3IEDM in section 5.4.3.
- *Domain Ontology Layer:* Finally, we have a bottom layer of specific and focused domain ontologies like the Swedish Defence Organisation structure ontology, weapons of mass destruction ontology, terrorist ontology, vessels, etc. Note that here we refer to a *collection* of ontologies rather than a consolidated single ontology. These ontologies may be application oriented or task oriented. We will discuss more about these ontologies in section 5.4.4.

### 5.4.1 Salient Features of DCMF-O

Some of the advantages of the DCMF-O Architecture are:

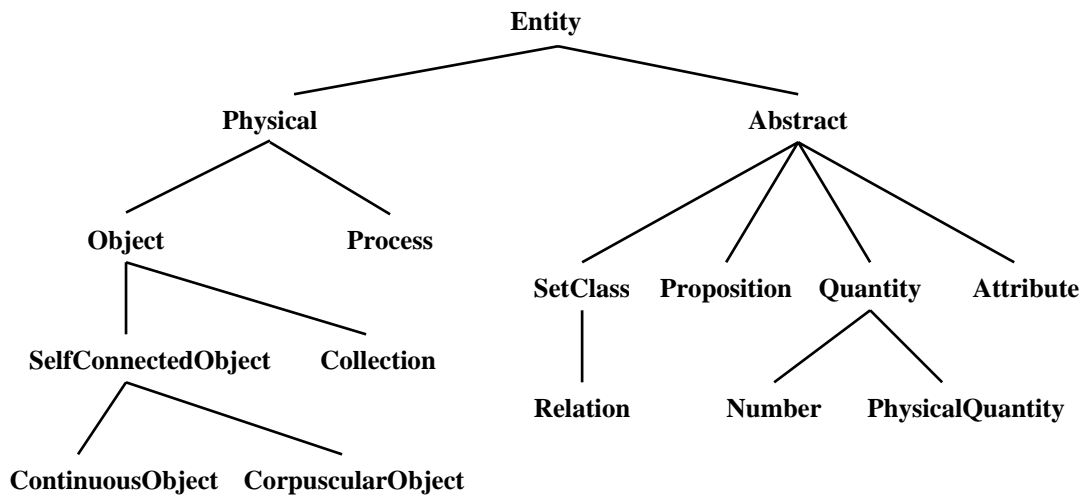
- This stratification allows us to improve reusability of the proposed ontology by other users and applications.
- At the same time, it allows us to plug-and-play other existing ontologies, like the upper level, SUMO ontology for generic concepts. Or the domain ontology for Weapons of Mass Destruction domain ontology (for details see section 5.4.4).
- It also provides us with the desired level of flexibility, so that we may modify or extend parts of the ontology structure, without needing to modify or change the entire structure.
- Another advantage is that we can have different team of experts working on designing and developing different parts of the ontology. Thus, we could focus on different areas of the domain in-depth.
- On comparison to the requirements put forward in section 5.2, we see that most of the requirements are met like flexibility, adaptability, extensibility, reusability. The choice of using UML class diagrams as well as in machine readable OWL format also satisfies the requirements for different formats for different end users.
- Some of the other requirements like authenticity, credibility, produceability have not been the focus for the current research, But as these are more related to the precise methodology of knowledge extraction and analysis rather than knowledge modelling, we believe that these issues shall be met through the series of methodologies proposed as well as the over all DCMF Process methodology.

### 5.4.2 SUMO: Suggested Upper Merged Ontology

SUMO is an upper ontology that has been proposed as a starter document for the SUO WG (Standard Upper Ontology Working Group), an IEEE-sanctioned working group of people from the fields of engineering, philosophy, and information science. The Standard Upper Ontology Working Group (SUO WG) is developing a standard that will specify an upper ontology to support computer applications in areas such as data interoperability, information search and retrieval, automated inference, and natural language processing [SUO05]. According to [Nil01], it is estimated that SUO will eventually contain between 1000 and 2500 terms and roughly ten definitional statements for each term.

As mentioned above, SUMO has been a starter document and it is still in the development phase. The most recent proposed version is 1.72, proposed in December 2004. SUMO is designed to be relatively small so that assertions and concepts will be easy to understand and apply. Currently, the ontology consists of approximately 4,000 assertions (including over 800 rules) and 1,000 concepts. The knowledge representation language for the SUMO is the SUO-KIF (Knowledge Interchange Format) which is a predicate logic. A specification of the current version of SUO-KIF can be found at [SUK05]. The ontology can be browsed online, and source files for all of the versions of the ontology can be downloaded from [SUM05].

The structure of SUMO's top level concepts is illustrated in figure 5-3. The root node of the SUMO is 'Entity', and this concept subsumes 'Physical' and 'Abstract'. The former category includes everything that has a position in space and time, and the latter category includes everything else. Physical entities are further divided into 'Object' and 'Process'.



**Figure 5-3:** SUMO Top Level Structure [Nil01]

Other general topics covered in the SUMO include: structural concepts such as instance and subclass; general types of objects and processes; abstractions including set theory, attributes, and relations; numbers and measures; temporal concepts, such as duration; parts and wholes; basic semiotic relations; agency and intentionality [Nic03].

In addition to the SUMO core upper ontology, SUMO is also associated to a Mid-level Ontology (MILO) and a set of domain ontologies. These domain ontologies include [SUM05]:

- Communications
- Countries and Regions
- Distributed computing
- Economy
- Finance
- Engineering components
- Geography
- Government
- Military
- Transportation
- World Airports.

We look at relevant parts of MILO in the next phase of the DCMF Project, so that we may extract parts like that of Military, Transportation, and Geography into our proposed DCMF-O. Whether the extracted MILO concepts will become a part of the JC3IEDM or become another horizontal extension is still an open question.

### 5.4.3 Overview of JC3IEDM

The main source of information and the basis for the ontology design and development is the MIP (Multilateral Interoperability Programme) [MIP05] proposed standard JC3IEDM (Joint Command Control Communication Information Exchange Data Model) [JC305]. The MIP aims to provide an assured capability for interoperability of information to support joint military operations. Interoperability is not envisioned merely at a data level but also at strategic, operational and tactical level to allow proper planning and functioning of joint

operations. As this objective of the MIP is well in line with the goals of the DCMF project, the proposed JC3IEDM was also found to be appropriate standard to base our ontological knowledge base representation.

The purpose of the JC3IEDM, listed as an extract from the specification states:

- A description of the common data in the JC3IEDM that contains the relevant data, abstracted in a well structured normalised way that unambiguously reflects their semantic meaning.
- A base document that can be used as a reference for future amendments to the model.
- A core upon which nations can base their own modelling efforts of chosen areas and on to which specialised area models can be attached or “hung”.
- A basic document that nations can use to present and validate functional data model views with their own specialist organisations.
- A specification of the physical schema required for database implementation.

The specification document along with its several annexes is well over 1000 pages long. The specification is technically a brilliant document, with detailed explanations, models and notes on each component of the data model. It would be tedious to list all of its salient features here. Therefore, we list a few of the main concepts and refer the interested reader to the main JC3IEDM specification document. [JC305].

The JC3IEDM is a revised version of the previous LC2IEDM which is a data model covering solely Land Military Operations. The current JC3IEDM is intended as a comprehensive data model covering all joint military operations, be it air, land, sea, etc. The JC3IEDM or its predecessor LC2IEDM is also known by a short name: Generic Hub. The name comes from the basic data exchange functionality provided by the JC3IEDM. One of the primary goals of the model is to establish the minimum required data parameters in an easy understandable format, so that interoperability of data between the different systems, military governments/units can be carried out. Using the Generic Hub as a common data model implies that each national military organisation needs to comply their internal information systems with the specified JC3IEDM model.

The JC3IEDM is described from three different perspectives:

- *Conceptual Data Model:* A top, high level, Conceptual Data Model of generalised concepts such as Actions, Organisations, Materiel, Personnel, Features, Facilities, Locations, intended for top officers, senior commanders etc, who do not need to know the specific technical details of the model, but is sufficient to be aware of the different concepts and their relationships.
- *Logical Data Model:* Middle, Logical Data Model which is more detailed, is based upon breaking down the high level concepts into specific information that is regularly used. For example, a tank is an armoured fighting vehicle that is a piece of equipment that is a piece of materiel. It also makes implicit knowledge explicit, like following the human reasoning patterns that a tank is a piece of armoured fighting equipment and allows command and control systems to generalise by recognising, for instance, that tanks are equipment. A logical data model specifies the way data is structured with an entity-attribute-relationship diagram and supporting documentation. At this level, technical implementation specific details are still obscured from view. This level is useful for middle level system analysts and domain experts.
- *Physical Data Model:* The third and lower most, Physical Data Model provides the detailed specifications that are necessary to generate a physical schema that defines the structure of a database. Mainly intended for the information system developer. The

physical data model can be seen as a traditional database schema model, which illustrates the different logical concepts (tables), their attributes (fields) and the relationships.

We have adopted a combination of the top two levels, the conceptual and logical model for our proof of concept research work

#### 5.4.3.1 Overview of Main Concepts in JC3IEDM

In this section we present a summary of some of the main concepts from the JC3IEDM. The description is not a technical documentation but more an explanation for facilitating easy understanding for the reader and to introduce the reader to some of the principal concepts and terms, adopted in this proof of concept, which is also used and illustrated in our results section.

The JC3IEDM has been setup to fulfil the following requirements:

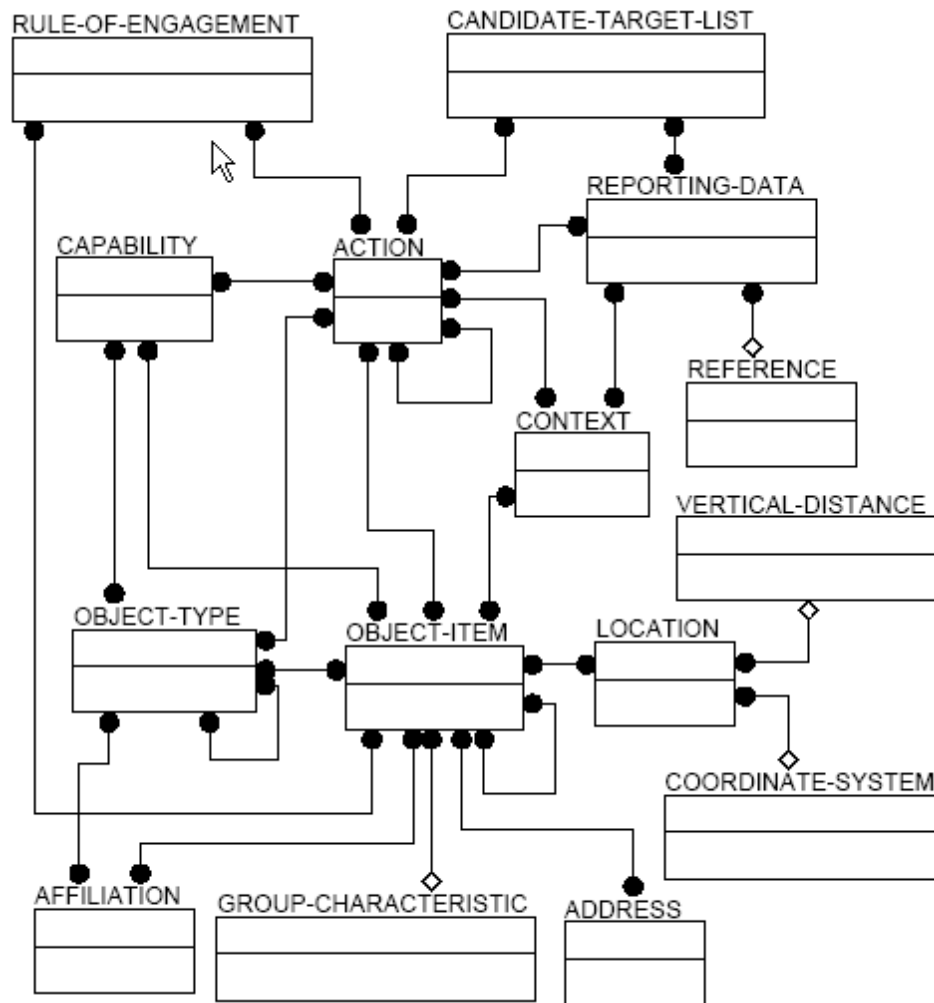
1. There is a need to describe objects of military significance. *Objects of interest* refer to physical things like units, equipment, materiel, personnel. Also non-physical concepts like location, co-ordination points.
2. Individual objects need to be differentiated from a particular type of objects. Also there should be a way to identify groups of individual objects.
3. Objects should be identified by sufficient characteristics as is required for command and control tasks.
4. There should be means to display operational status and *situations* as represented by facts of the objects.
5. There should be means to record and maintain historical log records of objects and other dynamic information as *information packages*.
6. There should be means to record *activities* of the objects.

From the perspective of the DCMF project:

- To describe a MSM, one needs the same concepts of Objects or Entities of interests around which the operation is focused.
- Each entity has both static characteristics as well as dynamic properties, as is represented by the *situation* concepts in the JC3IEDM.
- And the main concept in an MSM is that of activities, or Actions as proposed in the JC3IEDM as well.
- And to co relate pieces of information, to provide the context and other vital information, a group of information packages is required as well.

Thus, this point to point matching of the requirements for the DCMF Project with the main fundamental grouping of concepts of the JC3IEDM was yet another compelling reason to choose the JC3IEDM as a centre figure.

The central concepts in the JC3IEDM are extracted in the figure 5-4 given below:



**Figure 5-4:** Conceptual Data Model of the JC3IEDM

We explain the principal concepts given in figure 5-4 above by explaining their intended *use* or the *purpose* for defining them. We leave the interested reader to refer to the JC3IEDM specification for a precise technical explanation.

1. The main central nucleus of the model is the *action* which is used to describe military actions on the lowest granular level or even operations on an aggregated higher plane. Since military actions are always associated with a particular purpose, the *action* is related to a *context*. All technical data and informational aspects are modelled by *reporting-data*.
2. The model differentiates between planned actions and unplanned occurrences as *action-tasks* and *action-events*. (These are not depicted on the top level of the conceptual model in figure 5-4).
3. As expected in any military operation, actions are associated with certain *rule-of-engagement* and are applicable to prescribed *candidate-target-list*.
4. Objects have been defined as belonging to a particular type 'Object-Type' or as an individual 'Object-Item'. Object Types are generally *static* and *persistent*, whereas individual Object-Items are *dynamic* and most likely to change over time. For example, the characteristics of gun, main track width, load class, etc., are attributes of Object-Type whereas, actual fuel contained, ammunition left, current operational status of a tank are characteristics of Object Item. Both Object-Type and Object-Item are further classified in to extensive hierarchies. (We chose to implement all the



second level of classification. Further on, we chose to model only relevant categories and sub categories. The rest are left for future work.).

5. An *object* must have the capability to perform a function or to achieve an end. Thus, a description of *capability* is needed to give meaning to the value of objects in the sphere of operations. At the same time, each *action* needs a certain minimum specified *capability* from the object resource which can carry out the specified action.
6. It should be possible to assign a *location* to any item in the sphere of operations. In addition, various geometric shapes need to be represented in order to allow commanders to plan, direct, and monitor operations. Therefore *location* is related to *object-item* concept. Examples include boundaries, corridors, restricted areas, minefields, and any other control measures needed by commanders and their staffs.
7. All *objects* have *affiliation* to either some political nation, ethnic group, religious group or any other radical grouping which do not fall under the normal object-type grouping of *organisation*.
8. Furthermore, several aspects of *status* of items need to be maintained. So we have *object-item-status*, *action-task-status*, *action-event-status*, etc.
9. As described in the requirements for the JC3IEDM, the model permits a description of the composition of a type object in terms of other *type objects*. Such concepts include tables of organisations, equipment, or personnel.
10. In addition to the obvious concepts modelled as classes in the figure 5-4 above, we also have a number of relationships between the concepts which reflect information about what is held, owned or possessed in terms of types by a specific object item. (*Situations* - establishments, holding, association etc) There is a need to record relationships between pairs of items. Key among these is the specification of unit task organisations and orders of battle.
11. The model also supports the specification of current, past, and future role of objects as part of plans, orders, and events.
12. The same data structure could be used to record information for all objects, regardless of their hostility status.
13. Provision has been made for the identification of sources of information, the effective and reporting times, and an indication of the validity of the data.

So far, the description of the basic concepts matches with the expected conceptual model for military operations domain. And as previously stated the JC3IEDM has been developed through a consensus of several nations, part of the MIP. Thereby, the validity of these primary concepts is well established.

We found that the existing JC3IEDM data model cannot be used *as is* for a variety of reasons as explained in the section 10.2. So, we propose a design philosophy to adapt the JC3IEDM to suit the needs of our DCMF-O as shall be presented in section 10.2. But before we delve deep in to that, we present an overview of the application/scenario oriented specific domain ontologies.

#### 5.4.4 Domain Ontologies

The same ontology design and development methodology (MiSO) has been followed for the design and development of each individual domain ontologies in the cases where the domain ontology has been built from scratch. In cases, where pre existing ontologies have been used, they have been carefully studied and re-engineered. Here to, we found help from the UPON methodology as well as METHONTOLOGY design methodology, as these were the two

approaches which supported ontology re-engineering or modifying from other existing sources. In some cases, additional concepts were added to enrich the existing ontologies.

There are today five domain ontologies which were defined mostly on the basis of the scenario in the case study (see chapter 8). The scenario contains descriptions of weapons, vehicles, terrorist activities and different layers in the Swedish Defence and therefore it was natural to build domain ontologies on those subjects. We also put in a fifth domain ontology, Weapons of Mass Destruction, since it was already available and formalised as an extension of SUMO. Input from SUMO and the JC3IEDM has been considered in the conceptualisation and storyboard design phase of the domain ontologies themselves. The domain ontologies are either merged or mapped to extend the JC3IEDM. Section 10.3 describes the merging and mapping methods and tools in more details.

None of the domain ontologies are evaluated and there are inconsistencies. Those ontologies need to be more elaborated to be used in a sharp context. Meanwhile they function as an example for support concepts in our scenario and to show how domain ontologies can be mapped to the JC3IEDM.

#### **5.4.4.1 Weapons Ontology**

The Weapons Ontology describes weapons and ammunition used in the Swedish Defence for example Missiles, Small arms and Machine guns. This ontology is developed by us with concepts taken from a publication available on the Internet with information about weapons in the Swedish Defence. [W&V05] The concepts in this source were stated in Swedish but also translated into English and the English concepts were used. The Weapons Ontology is mapped to the JC3IEDM under Material Type.

#### **5.4.4.2 Vehicle Ontology**

The Vehicle Ontology holds information about different vehicles used in the Swedish Defence. Under the root concept Vehicle we find the concepts Aircrafts, Motor Vehicles and Ships which all have subclasses in at least two layers. The Vehicle Ontology was also developed by the DCMF project group and the source for this ontology is the same as the Weapon Ontology above. [W&V05] This Ontology was just as the Vehicle Ontology mapped under Material Type in JC3IEDM.

#### **5.4.4.3 Weapons of Mass Destruction**

This Ontology includes chemical and biological weapons, as well as radiological weapons. It was already implemented in OWL as an extension of SUMO and we could just reuse it. The Ontology is available on the Teknowledge Corporation homepage [WMD05] and there is also a description about the model, point of contacts and submitter of the Ontology. Examples of classes are: classifications of different bacteria and diseases that those bacteria can cause.

#### **5.4.4.4 Terrorist**

The Terrorist Ontology was also reused from Teknowledge Corporation since it is already formalised in OWL and extends SUMO. This Ontology describes terrorist organisations and

has concepts like ForeignTerroristOrganisation and TerroristOrganisation. The details about the Ontology can be found on [Ter05].

#### **5.4.4.5 Organisation Swedish Defence**

This Ontology was developed by us and the source for the concept and structure is the Swedish Defence homepage [Org05]. The Ontology is already not up to date since it was developed before the new Organisation structure in the Swedish Defence was available. Another problem is that this Ontology is written in Swedish which makes it difficult for tools to map it to other Ontologies with English concepts (see section 10.3). This ontology was merged with the JC3IEDM manually because the concepts were in Swedish, the mapping tool called PROMPT could not sense the relations to the Swedish concepts.



## 6 The Knowledge Meta Meta Model - KM3

### 6.1 Introduction

The Knowledge Meta Meta Model (KM3), developed by the Swedish Defence Research Institute (FOI), has its place as a tool and a language to construct well-formed conceptual models that are to be used successfully in simulations. To fully understand its role, it is instructive to learn about one information flow in the DCMF process and see where it is used in this flow. A general flow of information when producing the models is envisioned to be roughly like in the following list, where the main use of the KM3 is in the third step:

1. Partially structured information is gathered from some sources. For example, it could be interviews from witnesses of an event or some other sensor information. In any case, it is a statement about the world containing facts or fiction.
2. The information that is gathered is interpreted and structured according to a world view (ontology). That is, if the event is a combat scene or a terrorist attack, the event is interpreted in terms of the ontology. A number of ontologies may be applicable when interpreting the same event depending on who the interpreter is.
3. The interpreted information is subsequently transformed into a common format, generalised and stored as a reusable model. This common format is described by a Knowledge Meta Meta Model (KM3).
4. The reusable conceptual model, now called a Mission Space Model (MSM), can be instantiated with real world data and serve as a basic structure when performing simulations.

The intention when producing the KM3 was not to construct a grand “unified model description language”. It rather represents one possibility to capture system structures and behaviour in an object-oriented and rule based way. The purpose of this section of the report is to overview the main features of the KM3 and present some design choices when constructing it.

### 6.2 The KM3

The KM3 is all of the following (in no particular order):

- The KM3 is a specification. It is a specification consisting of object-oriented concepts, primarily aimed at capturing different dependencies in and between activities. In this setting this means that the KM3 is a specification for the creation of generic and reusable conceptual models of objects and processes of (military) interest.
- The KM3 is a tool. It is a tool for structuring knowledge about objects and processes as conceptual models. The main objective of KM3 is to produce generic templates of knowledge (MSMs, in the above list).
- The KM3 is a language. It is a common language to for different stakeholders involved in the modelling process, to enable them to construct conceptual models.

Thus, the KM3 is mainly used as a specification for construction of generic models, which in turn are used to model knowledge at an instance level. KM3 is, in this respect, a model for how to make models. A model produced using the KM3 is a well-formed, well understood conceptual model which in turn can be instantiated to be used as a simulation model.

KM3 is used as a means for capturing knowledge. Within the KM3, knowledge is defined to be the possibility to capture, in an object-oriented and rule based way, all relevant information pertaining to a situation as a conceptual model. Furthermore, the basic elements of situations are actions. This means that knowledge of a situation is based on the knowledge of the actions in the situation. The rest of the knowledge captured in the model follows from this.

To understand all the effects and consequences of the performance of an action is considered to have deep knowledge about a situation. To see several applicable actions in a situation, without being able to fully grasp their consequences are considered to having broad knowledge about the situation. Ideally, a model should be defined to capture as broad and as deep knowledge as possible. This means that the model should be able to incorporate as many relevant actions and other objects as is necessary to accommodate for as full understanding of the situation.

The implication of this knowledge view is that it should be possible to define an object with its relations to other types of objects to the extent and level of detail that is needed for the modelling purpose. It should also be possible to, in a controlled fashion, add new information to the model as it becomes available, change available information as it is reassessed, and delete information as it becomes obsolete. How this maintenance of models is done is part of future research on the KM3.

### **6.2.1 Demands**

A number of demands were put on the KM3 that influenced its design. It should be a) activity centric, b) it was necessary to be able to capture the static and the dynamics aspects of objects in the same model, c) the models should be reusable, and it should be d) possible to model uncertainty of activity execution. Items a), b) and c) are further explained below, whereas item d) can be commented on directly. Uncertainty in execution of activities is in the current version of the KM3, is modelled as an attribute of the rules which act as the start and stop conditions of an activity. For instance, it is possible to express that the chance of an activity to start is 0.85 (on a scale from 0 to 1) and, once started, the chance of it stopping is 0.34. A part of the future research concerning the KM3 is devoted to exploring and further enhance this system.

### **6.2.2 Activity Centric**

A common notion is that a process is a partially ordered set of activities with a well-defined goal. The activities are indivisible units of work that are combined to form processes. KM3 is primarily focused on the description of activities (which in the KM3 terminology is called actions). The reason for taking this route is to make the abstraction level more generic and thereby increase the reusability of the models. A process, where the ordering of actions is significant, is considered less reusable than the modelled actions by themselves.

Also, a characteristic of military operations is that they normally can be seen as combinations of actions rather than monolithic processes. The generic actions are combined to form descriptions of processes. For example, the process of driving a car from point A to point B under different weather conditions would result in a large number of process descriptions, whereas an action centric description of the same process would result in fewer, but more generic, models where some minor part would be focusing on the differences between the processes.

Actions in KM3 are modelled as objects. To model the execution of an action, all other objects that are affected by the action execution should be modelled as well. This also includes other action objects that may be seen as parts of a main action (sub-actions of the main action). Every action is modelled as a named set of state and rule descriptions. States are attributes of the action object and rules are conditions for changing the attribute values. When a condition becomes (is) true, values of the action attributes changes. This change also includes attributes of other objects that may be affected by the execution of the action, and this may, in turn, trigger further actions. The effects propagate through the model. In this way, the dynamics of an entire model is captured through rules of execution of activities.

### **6.2.3 Static and Dynamic Descriptions**

A static description of an object is an enumeration of its attributes and the domains from which the attributes can take their values. For example, a Person object can be described by giving it a name and filling the values for the attributes Length and Age. The permissible values are constrained by the domain definitions from which the attributes can take their values. This snapshot description is often sufficient when reasoning about the object. However, to be able to express how an object evolves, it is necessary to include dynamical descriptions of the object. In KM3 this is done by the inclusion of rules into the object descriptions. The rules determine how the states of objects changes as the objects evolve. Dynamic rules do not say anything about, e.g. the range of an attribute's values. They say which attribute value can legally precede or succeed another attribute value. Often the rules are formulated as being obligations, recommendations, or prohibitions. An obligation means that an object state must follow another; a recommendation means that a state may follow another, and a prohibition means that a state must not follow another.

The static and dynamic views of an object are complementary. A Person object has the attributes Name, Length, and Age taking values from defined domains. This is the static view. It also has a rule, which says that by each passing of a year, its Age attribute should be incremented with 1. This is the dynamic view. Any modelling tool, including the KM3, should have the possibility to model both static and dynamic views of an object. In KM3 this is done through the modelling constructs attributes and rules.

### **6.2.4 Encapsulation**

Inspired by ideas from classic object orientation, the KM3 uses encapsulation to package and protect different model elements. Being the basic elements, action objects encapsulate the entity objects that are associated with the activities. The only part of the entities that are visible to the actions is the current state of the entity. For instance, an entity object Person may publish the values of its attributes Age and Length for the action object to read. No part of an action is visible to an entity. The state of an encapsulated entity becomes automatically part of the state of the action.

One demand that was put on the KM3 was that of a high level of reusability and maintainability of the models produced using it. The main benefit of adopting this view on encapsulation of model elements, together with focusing on activities, is that it results in modularisation of the models, which increases their reusability and maintainability.

### 6.3 A High Level View of KM3

The KM3 consists of four main components; 1) Model element, 2) Attribute, 3) State, and 4) Rules. Model element is analogous to the class concept in object orientation, i.e., it can be thought of as a grouping of similar object with similar properties. Attribute is analogous to multivalued attributes. State is analogous to a single valued attribute, and finally, rules are analogous to operations and are used to control activities and state changes. Following is a presentation of the components and their relations. Figure 1 is the KM3 with the components indicated through encirclements of classes. Another view of the model can be found in Appendix A.

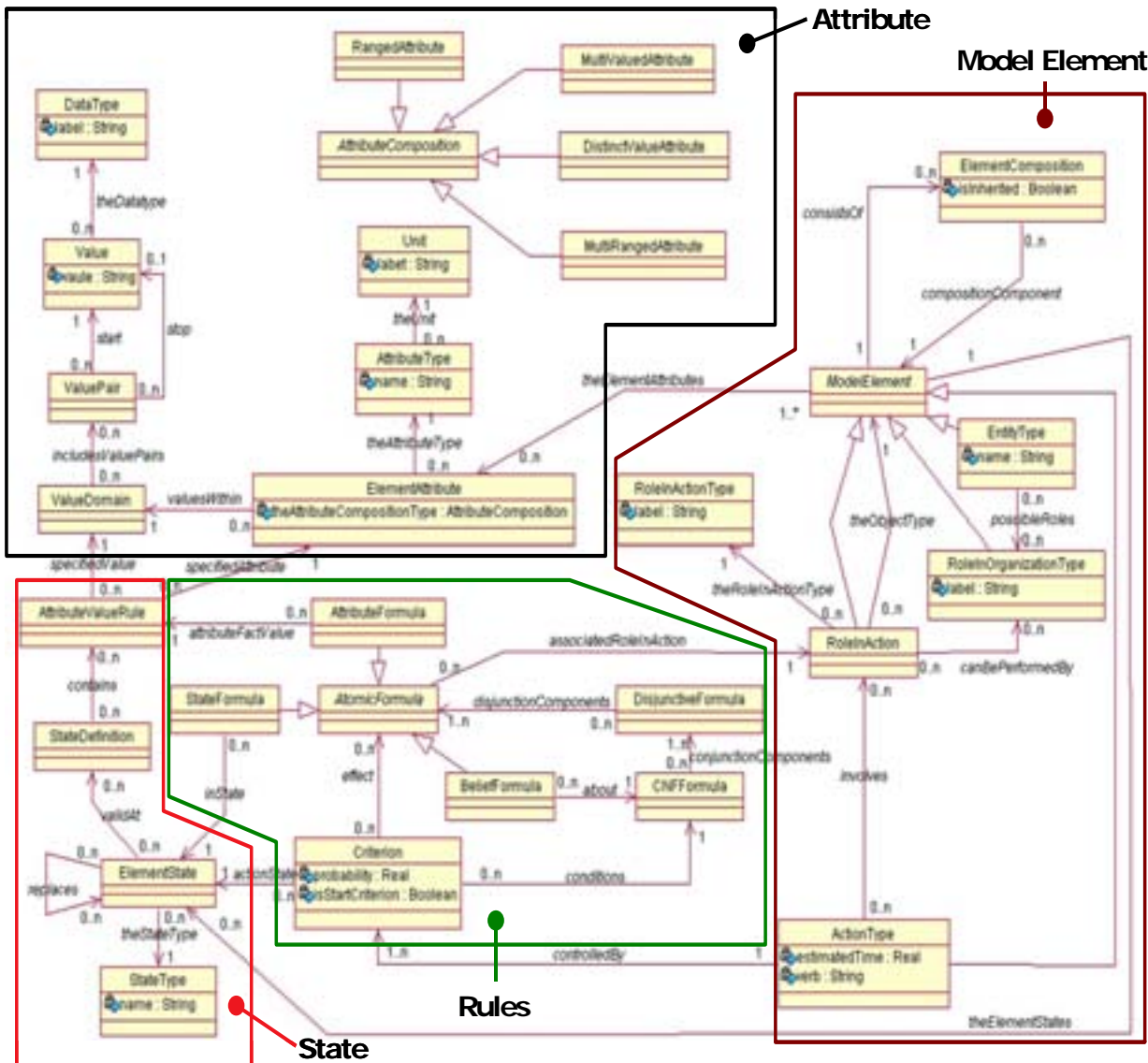


Figure 6-1: The KM3

#### 6.3.1 Model Element

Model element is the fundamental construct of the KM3. Through this construct, the objects that are part of activities as well as the activity object themselves are modelled. Every model element is recursively decomposable in different ways. Common for all model elements is



that they can be associated with attributes and states. Model elements can be of four different types: `EntityType`, `RoleInAction`, `RoleInOrganisationType`, and `ActionType`.

Figure 6-1 is the KM3 model in UML Class diagram notation. It is best understood by starting from the `ActionType` construct (bottom right in the figure). This corresponds somewhat to first finding a verb in a sentence. After this the other entities related to the activity are determined. This corresponds to finding the nouns and other word classes. If a required element is missing, information about it must be gathered, described and integrated into the model before finishing the modelling of the action.

Deciding on which entities should be associated to the verb is governed by the `RoleInActionType`. Action role types can be grammatical constructs such as subject or object, or named roles e.g., “the vehicle” or “the wounded”. Every action role is associated with an entity type that in effect classifies the role as to be played by an entity type described by a noun.

The difference between `RoleInAction` and `RoleInOrganisationType` is that different Types can play different roles in different contexts. A `RoleInOrganisationType` is characterised by rights (obligations, recommendations and prohibitions) of a position in a military organisation. A `RoleInAction` defines the actors within a specific action. In other words, it is possible to define similar names for concepts but their meaning is never the same. The rights associated with an organisational role constitute a role in an action.

It is important to realise that there is a difference in the level of abstraction between action roles and the other model elements. A role in an action denotes which type of entity it is played by. Actions associated with action roles are not explicitly expressed in the model but are derivable from the association.

When the action roles has been fully determined, the pre- and post-conditions as well as states of the activity is specified. The definition of entity types are done independently of definitions of actions and each entity type can be associated with one or several different actions.

### 6.3.2 Attribute

An attribute describes an optional, measurable characteristic of a model element. Through a measurement the model element is classified according to some unit of measure. An attribute is always associated with at least one model element but there is a possibility that an attribute can be shared among several model elements. An attribute is always associated with one of the following classification names: `DistinctValueAttribute`, `MultiValueAttribute`, `RangedAttribute`, and `MultiRangeAttribute`. `DistinctValueAttribute` means that only one point value is valid for the attribute. `MultiValueAttribute` means that many point values are valid for the attribute simultaneously. `RangedAttribute` means that only an interval is a valid value for the attribute. `MultiRangedAttribute` means that several intervals simultaneously are valid values for the attribute.

Value domains establish the extension of the attribute given its classification. A value domain is defined by values giving the upper and lower bounds of its applicability. If the upper value is 'null', then the value domain is a distinct value. Otherwise it is an interval.

Every value in a value domain is associated with a data type. An application of this design can be to enable the automatic conversions of units of measure. For example, the attribute 'Length' measured in the unit centimetre is associated with the model element 'Person'. The attribute should be classified as a DistictValueAttribute, since we only want a person to be of one length at a time. Our value domain will thus be defined through a pair of values where the lower end is, say, 0 and the upper end is 250. All values are of data type Integer. In this example we have created an attribute that can be described by a number (at least 0, at most 250) and a unit (centimetre). The attribute can take exactly one value at a time.

### **6.3.3 State**

A state is a set of attributes with values associated with a model element. A valid state is defined by a rule (StateDefinition) formulation and includes restrictions on the values it can take. All states have unique names. Changes of state occur as model elements participating in actions evolve and this includes the states of the actions. A change of state of a model element means that some value of some of its attributes has changed. Why this has come about depends on the formulation of the rules governing the state changes (the Criteria).

As values of states changes, one state replaces another. Permissible state changes will take place according to the rules that constrain all possible state changes. A valid state of a model element is a state in which all values are in compliance with a state definition. That is, all optional and non-optional attributes of a model element must have values that are permissible in the state definition.

### **6.3.4 Rules**

The dynamics of model elements are captured by rules in KM3. Through the specification of rules, all changes of model elements are described. A rule specifies the conditions under which an action starts and ends. The shape of a rule is a pair, where the first element represents an activity role, and the second element represents an atomic formula. An atomic formula is a statement about the state or attributes of a role. Every rule can be classified as being a state-formula, an attribute-formula, or a belief-formula. Beliefs do not necessarily need to be objectively true to trigger an action, but rather reflect the subjective views of a person.

From atomic formulas more complex formulas can be constructed by connecting them conjunctively or disjunctively. A disjunctive formula is created by connecting conjunctive atomic formulas by means of OR-connections (a.k.a. disjunctive normal form, DNF). A conjunctive formula is created by connecting atomic disjunctive formulas by means of AND-connections (a.k.a. conjunctive normal form, CNF).

Criteria are created from rules. A criterion determines the start and stop conditions for actions, which in turn results in state changes of the action, and possibly also in other model elements. In other words, the state changes are the effects from some action criterion becoming true.

## **6.4 Discussion and Summary**

In the CMMS process there was identified a need for a common structure for describing conceptual models. One such structure, called Knowledge Meta Meta model (KM3), was designed to take this role. A number of demands were put on the KM3; the models produced

using it should be activity centric, be able to capture the static and the dynamics aspects of objects in the same model, be reusable, and it should be possible to model uncertainty of activity execution. These demands are part of the answer as to why invent the KM3 and not adopt any model designed according to the same set of demands. At the time of design no such model was known to us.

In sections 6.2 and 6.3 we have presented the main features of the KM3, and discussed how the demands influenced its design. In Appendix A additional, more detailed, information about the model is presented. A distinguishing feature of the KM3 when compared to other models is the rules section. It is common to model dynamic rules separately using a rule language. In KM3 rules are stated declaratively, as pre- and post conditions of actions resulting in changes of states, thereby avoiding imperative formulations of the dynamics of actions.

The main benefits of the KM3 with respect to the overall goals of the original CMMS (see chapter 1), and also the DCMF, can be stated as in the following list. A recap: “The main purpose of the CMMS is to facilitate and support *development*, *reuse* and *interoperability* between simulation models.”

- Concerning the facilitation of *development*, the KM3 supports this by providing a common description for all stakeholders of what is to be simulated, and thus serves as a bridge between the military experts and the developers. The military experts own the mission processes and are an authoritative source when validating the content of the conceptual models.
- Concerning *reuse*, the use of modelling and simulation in the Swedish Defence is increasing and, as a consequence, so are the concerns on how to secure, validate and maintain knowledge and how to keep efforts in doing so at a minimum. By creating libraries of validated conceptual models with certified qualities those models can be reused as components. This result in lower development times as new models need not be developed from scratches every time and better quality of the final product.
- Concerning *interoperability*, a trend within the automatic development of software today is known as the model driven architecture (MDA). The intention is to produce software cheaper and faster by letting the developers produce (sometimes graphical) models of a system. The models are then automatically transformed into executable machine code. Within this trend there is a recognised need for a common notation that unifies different models and one role for the KM3 is to fulfil the need for this common notation.

The current status of the KM3 model is that it is a work in progress. Future research involving the KM3 includes a validation of the DCMF process starting from an only partially structured information input, and ending in an information structure ready for use in the construction of a simulation model. The validation is expected to improve the overall DCMF process, and also the KM3.

Future research also includes an investigation of different ways of modelling uncertainty and risk. Currently this is done by including an attribute (accepting a value between 0 and 1) in the Criterion class. It is necessary, for instance, to define the exact semantics of the probability attribute. What does it mean to say that the probability is, say, .85 for a stop criterion to be true? This must be determined. Is this an appropriate way of modelling uncertainty, or are there other, more suitable alternatives? This should also be determined.

There is also a need to clearly decide on the exact relations between the KM3 and the other parts of the DCMF process. This includes relations between methods and techniques, as well as different artefacts. When, in the process, should the KM3 be used, and how should it be used at that point? We know that it is to be used when constructing the end result of the DCMF (the MSMs), but are there other opportunities to use it as well? An example of an alternative area of use is given in chapter 7 where it is used as an analysis tool. We also need to determine how to express one modelling standard, e.g., the JC3IEDM in the terms of the KM3. The relations between the different artefacts need to be as explicit and clear as possible.

Finally, another area for future research concerns the storing and maintaining of simulation models. This also includes work on languages for querying the model store.

## 7 Analysis Method (From KA to KR)

In order to analyse unstructured knowledge and transform it into represented knowledge we needed a parsing method. Some methods were studied and exemplified. This chapter describe those parsing methods and which method we chose and the reason for that choice.

### 7.1 Parsing methods

The scenario used as input to the prototype development was originally in free text. This next needed to be analysed and parsed in order to be structured into the models to become a Mission Space Model (MSM).

For this purpose we have studied three parsing methods. One of those the Subejct-Predicate-Obejct (SPO), has been studied, exemplified and documented [Moj04]. This year we continued with looking closer at the 5W:s for parsing text and express operations on this format. Finally we also used our own developed Knowledge Meta Meta Model (KM3) as a parsing method. Initially the KM3 was developed to be used in the later phases of the DCMF but we considered it meaningful to test if the KM3 was useful as an analysis tool.

In this section all three methods will be described as analysis methods for parsing unstructured scenario text. The original text is about Swedish Forces participating in a Peace Support Operation in Cosovo where various activities are taking place. For a more detailed description of the scenario, see chapter 8.

#### 7.1.1 Five W:s

The Five W:s (5W:s) are used as an analysing tool in several disciplines but also in the military domain. The method answers the questions: *Who* is doing *What*, *Where*, *When* and *Why* in a scenario. [Kat05, Tri05, Dem05]

More specifically, a sentence is parsed according to the 5W:s by asking the questions:

- *Who* is doing something?
- *What* is going on?
- *Where* is it being done?
- *When* is it being done?
- *Why* is it done?

Even more specific for the military domain the 5W:s provide a structure for describing the commanders intent and military tasks on the following basis:

- *Who*: unit that accomplish the task
- *What*: task to be accomplished
- *When*: timing of the task
- *Where*: location for accomplishing the task
- *Why*: reason for accomplishing the task [Per05]

To exemplify we chose a sentence from the chosen scenario (see chapter 8):

“UN Peacekeeping Force will cut off weapon transport at village Y at Date 2005-05-02 in order to diminish powerbase.”

On the basis of the questions above this sentence is parsed into the components:

UN Peacekeeping Force...	WHO
...will cut off weapon transport...	WHAT
...at village Y...	WHERE
...at date 2002-05-02...	WHEN
...in order to diminish powerbase.	WHY

**Table 7-1:** Description of how the sentence is parsed according to 5Ws.

### 7.1.2 Subject-Predicate-Object

Another parsing method is Subject-Predicate-Object (SPO). The SPO terms are clause elements or components in sentences from a grammatical perspective. The subject and object can also be seen as nouns and the predicate as verb. The SPO structure is used in Resource Description Framework (RDF) which is the base for the Semantic Web [W3C05]. This method was studied and tested in last years activities. For a more detailed description, see [Moj04].

According to the SPO sentences is parsed by asking the questions:

What happens in the sentence?                      The answer of the question is the *Predicate*.  
 Who does “the Predicate”?                              The answer is the *Subject*.  
 What “the Predicate” “the subject”?                The answer of this is the *Object*.

Using the same sentence as above gives a different result with SPO compared with 5W:s. The sentence has to be rewritten in this case in more than one SPO triple.

“UN Peacekeeping Force will cut off weapon transport at village Y at Date 2002-05-02 in order to diminish powerbase.”

UN Peacekeeping Force...	Subject
...will cut off...	Predicate
...weapon transport...	Object

**Table 7-2:** Description of how the sentence is parsed according to SPO.

### 7.1.3 KM3

The third parsing method is the Knowledge Meta Meta Model (KM3) which was developed for use in the later phases of the DCMF. But it can also be used as an analysis tool in the earlier phases. The method for doing this analysis is to first find the verbs of the sentences. After this nouns and other word classes are determined. The grammatical constructs are then mapped onto the constructs of the KM3. The KM3 is described in chapter 6 and Appendix A.

To do it more complex and perhaps at the same time more interesting we have selected another longer excerpt from the same scenario:

“An officer at the Intelligence department receives the information automatically due to the pre-defined subscriptions for his role. The officer creates a main information element to be used as a point of reference when the errand is explored further. To find more information regarding the case, the officer searches the GIS-system (part of the NBI) with a special interest for the area around village Y.”

This text is analysed concentrating on the Searching activity and using the KM3 into the following structure. Comments meant to be included in the KM3 structure are put between /\* and \*/ signs. Comments are in separate paragraphs marked Comment.

ET:GIS

ElementComposition :: part\_of : <ET:GIS, ET:NBI>

Comment: the entity type GIS is a part of the NBI system. The syntax here is to prefix the names of entity types with ET:. In KM3 there is a generic way of handling relationships between model elements. This generic way is called ElementComposition. There are many types of relationships, e.g., aggregation, instantiations, or generalisation. In this case we have a relationship called part\_of which is another name for aggregation. Therefore the ElementComposition type is qualified with the part\_of construct. The intention here is to express that the GIS-system is part of the NBI-system

AT:Searching /\* Searching the GIS for additional information \*/

Comment: Here we have an action type called Searching. An action type is prefixed by the AT: construct. As all model elements, an action type has a name.

Time:May 2002, AFTER AT:Creation /\* of ET:Main\_information\_element \*/

Comment: All action types has a non-optional attribute called Time. This indicates when an instance of an action type takes place. There is also a possibility to relate the occurrence of an action with the occurrence of another action, as is shown in this example. Here the searching is related to the action type Creation through the temporal dependency AFTER, meaning that Searching takes place after the Creation action. This is one way to form combinations of generic actions into generic processes.

RoleInAction:<searcher, subscriber>

RoleInAction:<searched, info-system>

Comment: There may be a number of entities involved in an action. These entities have named roles in the action. Here are the roles searcher and searched, and they are filled by entities themselves being roles (in organisations).

RoleInOrganisationType:<subscriber, ET:Subscribing\_officer>

RoleInOrganisationType:<info-system, ET:GIS>

Comment: In the action there is a role called searcher. This role is filled by an organisational role called subscriber. This in turn is filled by an entity type Subscribing\_officer. Likewise, the searched role is filled by the role info-system which is associated with the entity type GIS.

Criterion:(probability 1, isStartCriterion T,  
[ET:Main\_information\_element AND ET:Relevant\_information])  
ActionState: Searching

Comment: a criterion is a rule stating the conditions for an action to start and when started, a rule for it to stop. Here is a start criterion indicated by the non-optional attribute isStartCriterion having the value T (for True). The criterion is formulated as an existence rule saying that if there exists entity types Main\_information\_element and Relevant\_information then the action should move into the state Searching. There is also an attribute prob that in this case has the value 1. This means that whenever the criterion evaluates to T, the action will always take place. A less than 1 number indicates that there is a chance that the action will not take place despite the start criterion being true.

Criterion: (probability 1, isStartCriterion F,  
[ET:Main\_information\_element updated AND  
ET:relevant\_information] OR

```
[ET:Main_information_element updated AND NOT
  ET:relevant_information])
ActionState : Not_Searching
```

Comment: Here is the stop criterion for the action. The criterion indicates that whenever an entity type is updated whether or not there exists any relevant information, the action moves into the state not searching.

```
ET:Main_information_element
State:Updated /* with additional information */
```

Comment: This is an entity type mentioned in the start criterion of the action. This entity type has an attribute called state which has the value updated. The value has been set as a result of the searching action. It is an effect of the action.

This example is quite simplified for at least three reasons; the first because there is much information prescribed by the KM3 which is not included here. For instance, talking about the `part_of` CompositionType requires a strict definition of the term (as is necessary for all terms). Secondly, this is only a small part of the scenario. The full scenario will render a very long description. Finally, the syntax as given here is simplified and not fully defined. A overview of the entire KM3 is given in chapter 6 and in appendix A. One item on the future research list is to determine and decide on the best syntax for the KM3.

## 7.2 Chosen approach

The result of the parsing will be different depending on which method is used. The decision of which method that should be used depends on the focus and the future employment of the parsed scenario.

Last year we studied the SPO and also carried out examples to test that method [Moj04]. The advantages of the SPO are the strong connection to the semantic web and formats such as RDF and Web Ontology Language (OWL) which support interoperability and machine readability. One disadvantage of the SPO is that the source often needs a lot of rework before it fits into the structure which might lead to information loss during the elaboration of the scenario source and later problems with the traceability.

This year we looked closer on the 5W:s. This method is interesting since it is both a general analysing structure and also used in some standard efforts in the military domain [Kat05, Tri05, Dem05]. One example is the Coalition Battle Management Language effort which propose the use of JC3IEDM together with the 5W:s structure, as one part of the concept, to reduce ambiguity and automatise the commanders intent [Dem05]. For expressing activities and orders in the military domain the 5W:s are considered usable. We are not really aware of the disadvantages with the 5W:s yet but it seems to be suitable for a certain level of abstraction and perhaps not really able to capture lower (more detailed) or upper (more abstract) levels of abstractions. However, this year we chosed the 5W:s as the parsing method for our scenario.

Furthermore we wanted to test our own KM3 as an analysis tool which turned out to be successful. The KM3 facilitated the interpretation and the structuring of data in a more detailed way than the other parsing methods. A scenario structured according to the KM3 will be machine processable and therefore possible to interoperate with other programs and systems. Due to the lack of methodology for using KM3 as an analysis tool the parsing activity was rather complicated when using the KM3.



An important conclusion is that all the parsing methods give different results with different focus and perspectives. Probably, there exists no optimal method, at least not among those mentioned here. The choice of the most appropriate method depends on the requested focus and the aim of the parsing activity.



## 8 Case Study

To be able to carry out a case study we needed some kind of input data. This kind of input data are normally coming either from Subject Matter Experts (SMEs) other military knowledge sources or military scenarios. We have already in previous work, [Moj04], tried a simple case study with knowledge acquired from an SME. This year we decided to choose a scenario as input to a case study.

In this chapter we will describe why a scenario was needed, the demands on such a scenario and the different scenario candidates that were studied. One of the demands on the scenario was the connection to the Networked Based Intelligence (NBI) project. Therefore this chapter also provides a brief description of the NBI project. Finally the chosen scenario is presented together with the justification of the choice.

### 8.1 Objectives and aims with the scenario

We needed input data to be able to carry out a case study from the Knowledge Acquisition phase via the Knowledge Representation phase to the Knowledge Modelling phase. (see figure 2-1) Through these phases a scenario with unstructured text would evolve into a conceptual model and in theory be storable in a repository for the final Knowledge Use phase. To find a scenario that best fit our purposes and was relevant for the Swedish Defence, we identified some criteria that should hold for such a scenario.

First the scenario should be based on the military domain. The second criterion was that the scenario should comprise a Peace Support Operation (PSO). Furthermore the scenario should have both a Control and Command (C2) perspective and an interoperability perspective. Finally the scenario should have some connection with Network Based Intelligence (NBI) because it would lead to increased possibility for cooperation between the NBI project and the DCMF project and better use of resources in both of these two FOI sibling projects.

#### 8.1.1 The NBI project

The vision of a future Network-based Intelligence (NBI) service, considered for the Swedish Armed Forces (SWAF), comprises the ability to safely and easily provide and be provided with knowledge relevant for a certain context. NBI should deliver services that will ease cooperation in distributed and networked organisations and enable automation of intelligence-related processes. In the beginning of 2004, the Swedish Defence Research Agency (FOI) was assigned the task to conduct research related to knowledge management within NBI. The long-term purpose of the research is to provide the Swedish intelligence community with knowledge required for the realisation of and shift towards NBI [Ekl04].

### 8.2 Scenario candidates

Three scenarios were studied. All of them has its origin in the military domain and were constructed for different reasons by actors in the military domain. Below are the three scenarios described.

### **8.2.1 Scenario 1**

The first scenario is taken place in Congo 2012. The situation in the world is more or less the same as 10-15 years ago with the same kind of conflicts and problems. Terrorism against western countries has escalated in general but in particular against USA.

The specific scenario describes a Swedish patrol that is a part of a PSO. This patrol has currently the mission to cut off a weapon transport in order to diminish powerbase for the local guerrilla leader. On their way from the camp the patrol gets information via the information system that a children soldier troop is moving towards them. The patrol decides to change route in order to avoid the children troop and gets to the transport as planned.

### **8.2.2 Scenario 2**

The second scenario is taken place in Cosovo in May 2002. A Swedish patrol from the Swedish peace keeping force discovers a looted weapons depot and report this into the information system of the Swedish Intelligence. An intelligence officer in Sweden receives the report and starts a further investigation. Information from different sources leads to the estimate that the missing weapons might be smuggled to Sweden by organised criminals. Cooperation between different military and civil organisations to acquire information leads to the confiscation of the weapons in the harbour of Gothenburg in Sweden. Section 8.4 below contains the scenario in detail.

### **8.2.3 Scenario 3**

The source for the third scenario is documentation of a course in training pilots in flight simulators and the situation that the scenario offers is the following.

NATO forces are conducting a Peace Support Operation (PSO) in order to regain stability and security in Cosovo in May 2002. The specific mission of this operation is to establish an international co-allied air wing at air station Vicenza, Italy. Another part of the mission is to conduct air operations in the assigned Operation Area in order to assist the UN in restoring international peace and security in the Cosovo region. Such air operations may include reconnaissance, Air-to-Ground and defensive as well as offensive counter air missions.

The overall scenario described above is the background for the exercises which then are specified in a number of smaller scenarios. Those smaller scenarios are training situations where the pilots, in the simulator, have different roles and missions. For example Air-to-air refuelling or Status check.

## **8.3 Evaluation of the scenario candidates**

The second scenario was chosen because a) it met all criteria, b) it was considered to have almost sufficiently information and therefore it did not require so much further extensions or assumptions to create domain ontologies and MSMs

The first scenario met all the necessary criteria. The disadvantages were that it did not give so many details as needed and therefore required quite a lot of work up to be useful. There was a risk that the expansion could be time consuming and give focus on aspects other than the actual development of the prototype.

The third scenario only fulfilled two of the criteria namely the PSO and C2 perspectives. Furthermore the documentation of this scenario demanded knowledge about the domain of the air force and there terminology. Since the scenario was quite extensive it would have been necessary to work it up thoroughly in order to make it understandable. All these factors led to the conclusion that the third scenario was not suitable for the purpose.

## **8.4 The chosen scenario**

The scenario is taking place in Cosovo and its surroundings. NATO forces are conducting a Peace Support Operation (PSO) in order to regain stability and security in Cosovo.

### **8.4.1 Background**

The ongoing conflict is an effect of the following course of events:

- War in former Yugoslavia during last decade.
- Diplomatic efforts in 1999 to find a long-term peaceful solution.
- Peace breaks down again
- Interethnic riots in March 2002 on the people of Cosovo
- On 24 March 1999 NATO forces began air operations over the area to prevent a humanitarian catastrophe.

The return of refugees and displaced persons to their original homestead in the Cosovo region during Sep 2001-Mar 2002 went unsuccessfully:

- Returning refugees had no homes
- Riots
- Black market
- Guerrilla operations

Serb military and police forces answered with blockades of humanitarian aids.

A number of violations of human rights of ethnic Albanians in Cosovo where reported.

The conflict culminated on May 2<sup>nd</sup> 2002 when Serb forces killed 250 guerrilla men and civilian in an airborne attack.

### **8.4.2 Planned end-state**

The situation of Cosovo, Federal Republic of Yugoslavia is regained to a state of stability and security. Displaced persons and refugees are significantly restored to their original homestead.

### **8.4.3 The task of KS05**

The task of the Swedish contingent is to ensure safety for the inhabitants of the AOR (Area of Responsibility). Albanians inhabit the area mainly, but several thousands of Serbian's live in the area as well. The duties of the contingent involve escorting people of a particular group through potentially dangerous areas and manning roadblocks in search for weapons and explosives. The battalion also performs patrol and surveillance of the area to mark their presence.

#### 8.4.4 Course of events

Cosovo May 2002

1. A Swedish patrol from the contingent in Kosovo finds weapons in the forest near a village called Janjevo.
2. The observation is reported in the battalion's local Network-based Intelligence system (NBI-system), which triggers encrypted transfer of the observation to subscribers of the concerned information type.
3. The information about the found weapons is made available for Intelligence Division at the Joint Military and Security Directorate (MUST).
4. An officer at MUST receives the information automatically due to the predefined subscriptions for his particular role. The officer creates a main element to be used as point of reference when the errand is explored further.
5. To find more information regarding the case, the officer:
  - a. Searches the area near the site where the weapons were discovered using the GIS (Geographic Information System) facilities of the NBI-system. He finds a military weapons depot just north west of Janjevo.
  - b. The weapons depot feature in the GIS is interlinked with some supplementary information. This information makes some statements concerning a particular individual, Major Curtan Strangovic, having some relations to the weapons depot. According to the knowledge base of NBI:
    - i. Major Strangovic has connections to organised criminal groups in Sweden.
    - ii. He has lived in Sweden, but moved back to Kosovo in 1995 to support the Serbian struggle.
    - iii. He is the leader of a military unit that is responsible for the weapons depot in Janjevo.
6. The officer issues an information request, destined for the Swedish contingent in Kosovo, in the NBI system. He wants to know: has the weapons depot been looted? Who or which group could have been involved? The information request is coupled with the main element of the task in the NBI-system.
7. The battalion in Kosovo acquires the requested information and enters the observation in the NBI-system where it is inter-linked with the main element of the task. The supplied information states:
  - a. There has been a burglary in the depot and weapons are missing (they know that weapons are missing since they can compare the current state of the depot with an inventory made recently).
  - b. People living nearby can tell that there have been some nightly activities in the surroundings of the depot last week.
8. When the supplied information has been disseminated to the officer at MUST, he/she compares weapons missing from the depot with the weapons found in the forest area and establishes that a number of units are still missing. The missing weapons are relatively light and are commonly used by terrorists and other criminals. Heavier weapons like antitank weapons have been left in the forest hiding place.
9. Since Major Strangovic has a connection with Swedish organised criminals the officer estimates that there is a risk that the weapons are destined for transportation to Sweden.
10. All reports and estimates are inter-linked with the main element of the errand and pushed for dissemination at the Security Division.

11. An officer at Security Division reads through the background material from Intelligence Division and also comes to the conclusion that it is reasonable to believe that the weapons might be smuggled to Sweden.
12. Smuggling is not the responsibility of the Swedish Defence. Therefore the errand is pushed for dissemination by the Swedish Customs, having access to parts the NBI-system
13. The customs finds an interesting ship in the port of Budva, situated relatively close to Janjevo, in their knowledge base. The ship, named Pioneer, has earlier been involved in smuggling activities.
14. The officer at the customs estimates that the disappeared weapons could be transported to Sweden with this ship. Since land transportation can not be excluded, the customs raise the readiness at all borders, both at sea and land.
15. The customs requests support from the Swedish Military Defence.
16. The officer at Security Division estimates Pioneer's time of arrival to Swedish waters and the ship is being tracked on its way to Sweden, using various external intelligence systems
17. To be able to track the ship in greater detail when it approaches Swedish waters, the Security Division makes an information request through the NBI-system, destined for *Sjöbevakningscentralen*<sup>17</sup>. *Sjöbevakningscentralen* is subordinated OPIL<sup>18</sup>.
18. A handling officer at OPIL receives the request from Security Division and forwards it to the *Marinens Taktiska Förband* (MTK)<sup>19</sup> which sends a request to *Sjöbevakningscentralen* in Gothenburg.
19. *Sjöbevakningscentralen* identifies Pioneer and registers her in the GIS facility of the NBI-system. Then all organisations (OPIL, MTK, Swedish Customs and the Security Division) can track the ship live on her way into the port of Gothenburg.
20. Upon arrival the customs searches the ship and find the disappeared weapons.

---

<sup>17</sup> *Sjöbevakningscentralen* is a unit responsible for sea surveillance and monitoring

<sup>18</sup> OPIL – Operative Command of Swedish Armed Forces

<sup>19</sup> MTK – Tactical Command of Naval Forces





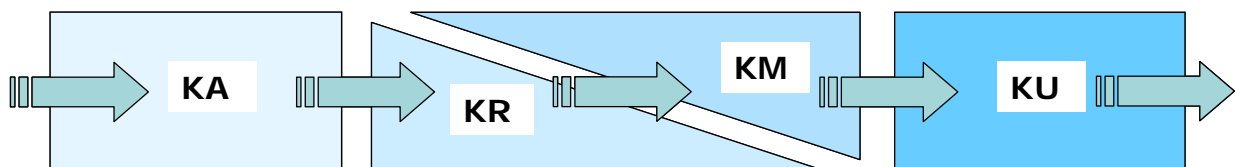
## 9 Case Study Analysis

So far in the preceding chapters, we have discussed the overall objectives, objectives for the current phase, some literature and state of the art review. In chapter 2 and 3 we have discussed in detail the overall DCMF process and its goals, expected results and strategies. Thereafter in chapter 4 and 5 we have moved further deeply in to the design approach for our proposed ontology architecture. Then we have seen the details of our MiSO ontology design. In chapter 6, we have reviewed our indigenously developed KM3, for knowledge analysis as well as knowledge representation. In Chapter 7 we summarised knowledge analysis tools 5Ws, parsing methods, SPO. Now in this chapter we illustrate how all these different theoretical parts of our project function coherently in practice by using a case study.

This year, we chose to check the feasibility, and also evaluate our vision for the DCMF process by means of a practical case scenario analysis. The chosen scenario, which has been explained in chapter 8, was documented as a movie in Swedish. Therefore the first step was to transcribe it into English. Then it was divided into smaller sub scenarios. During seminars the scenarios was parsed into the 5 W's structure. In some cases the scenarios missed information on one or more W's. For example could a scenario miss information about WHY. In those cases the scenario was arbitrarily expanded with this information to make a whole 5 W's structure. For a description of the parsing method see section 7.1.1.

### 9.1 Overview of Scenario Analysis Process/ DCMF Process recap

In chapter 2 and 3 we have discussed the four main phases of the DCMF Process, as shows in the figure 9-1 below:



**Figure 9-1:** The Different Phases of the DCMF Process

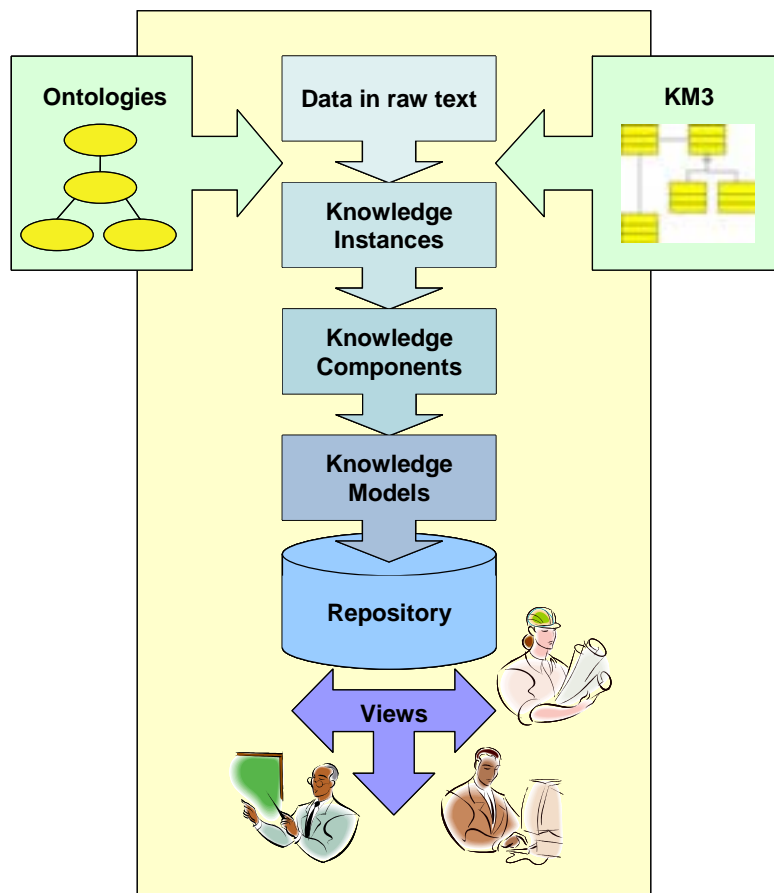
In the DCMF-P the transformation of any scenario goes through a series of iterative steps. In Knowledge Acquisition, all relevant information from the domain and the given scenario is accumulated or *acquired*. In the Knowledge Representation, the acquired knowledge is represented in a more structured representation format. In Knowledge Modelling, the semi structured knowledge is now formalised in to the proposed ontology, DCMF-O and the KM3. Finally, in Knowledge Use phase, the knowledge acquired and modelled so far is put to practical uses.

We may recapitulate these basic ideas as a step-wise summary as:

1. Assimilate information from various sources, subject matter experts, and literature. Information may be available from heterogeneous sources.
2. The assimilated natural language raw unstructured information is analysed, tagged and represented as semi-structured annotated text.

3. The analysed information is first formalised in to UML class diagrams, and finally in to ontology, before it is structured according to KM3. In fact, step 2 and step 3 go hand-in-hands iteratively till a clear and high level MSM is obtained.
4. In the final steps we make use of the ontology knowledge base set up as well as the KM3 models for MSMs for different end uses.

In the current cycle of proof-of-concept analysis of the chosen scenario, we have not been able to focus on all the aspects of the above mentioned steps. This year, we have concentrated more on the initial steps but mostly on knowledge representation and modelling (methodologies for knowledge acquisition through interviews has been extensively studied by us in earlier year, see chapter 2). There are several open issues left which shall be discussed in chapter 10. We have identified some of the major components of the DCMF Process as illustrated in figure 9-2 below. Now, we shall see how or what is going on in most of these blocks with respect to the chosen scenario.



**Figure 9-2: DCMF Process Overview**

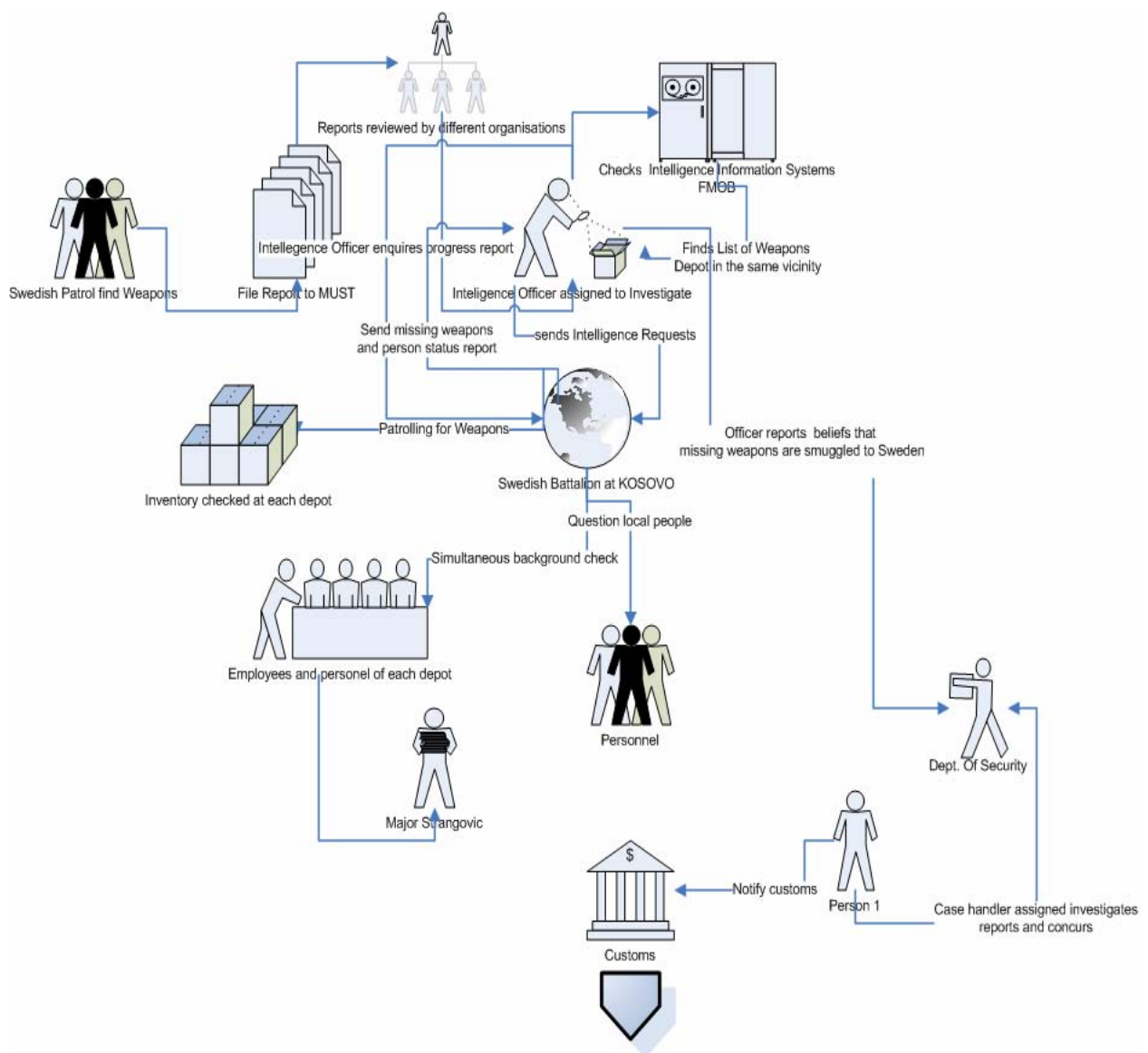
## 9.2 Scenario Analysis Overview

A use case scenario has been chosen, where a potential terrorist activity is noted and follow up actions carried out as part of a peace keeping operation. The place is Cosovo, where the Swedish contingent is part of the PSO – Peace Support Operation. For a detailed description of the scenario in natural language text see chapter 8. In this chapter, an overall analysis of moving from raw unstructured data and information towards a structured, re-usable Mission Space Model (MSM) is provided.

*Phase 1- Knowledge Acquisition:* The objective is to capture every nuance and aspect of the scenario as possible. In this phase the scenario exists as raw, unstructured data. (Observe that normally there are even some steps, about identifying the authorised knowledge sources, acquiring the knowledge from these sources etc, before these activities.)

*Inputs:* We represent the scenario using different medium of representations including natural language human representation, use case diagrams, Story board write ups of the scenario. Other inputs may be recorded messages, video clippings, other information system data etc. Thus in this phase of knowledge acquisition, we try to capture all relevant information and then try to represent them in as simple and human understandable form as possible.

*Outputs:* The output at this stage would be word documents, use case diagrams etc. A graphical use case scenario diagram has been developed as included in appendix D in enlarged format. We include an extracted and reduced version in figure 9-3 below.



**Figure 9-3: Part of the Use Case Diagram for Scenario**

*Analysis Process:* The scenario information is acquired through in-depth interviews of subject matter experts, review of different input source documents, videos, recordings and all other diverse forms of information.

*Phase 2- Knowledge Representation:* In this phase we analyse the knowledge contained in the text, by lifting up implicit, sometimes hidden or pragmatics of information as well as explicitly available data. In the current iteration, this transformation has been carried out by our team of experts. However, we foresee that a formalised and at least semi-automated transformation methodology based on the chosen approaches, should be developed in the next iteration. We have chosen to use the 5Ws approach and the KM3 as two independent approaches for knowledge extraction.

*Inputs:* The outputs from the previous phase that is the *raw, unstructured* information are the inputs.

*Outputs:* At the end of this phase the scenario exists as *semi- structured and partially grouped set of data*.

*Analysis Process:* We use the ontology to provide us initial concepts and terms to look for. As suggested by the UPON (section 4.5.5) we use the ontology as a lexicon (or a thesaurus) to help us in identifying the relevant concepts. Thereafter, we also propose a set of additional ‘transformation questions’ (based on the competency questions theory of Gruninger and Fox). These transformation questions are derived from the ontology itself and these are expected set of characteristics to be discovered from the scenario. An iterative process of discovery and matching with the ontology is to be done, till the minimum set of required explicit and implicit information has been extracted from the scenario. We give some sample list of transformation questions in Chapter 10.

*Phase 3- Knowledge Modelling:* We may choose to store just this structured information separately in an independent Knowledge Base. While the final form or methodology for this transformation is yet to be investigated, we foresee that as a first step, we would re use the mapping between KM3 syntax and the Ontology to extract the particular MSM structure for each scenario from the ontology and then represent it as a KM3 based MSM.

*Inputs:* We take the outputs from the 5Ws analysis and the KM3 analysis as a knowledge extraction tool.

*Outputs:* We map the semi structured data on to the proposed ontology, DCMF-O and finally create instances of the information to create the knowledge base. The outputs are the ontological representations of the instances themselves, in OWL, and some extensions to the existing ontology structure.

*Analysis Process:* In this knowledge representation phase, we transform the gathered data in to formal structured data, using UML class diagrams, and finally implemented into the ontology as ontology instance information. Once the knowledge has been created in to the MiSO ontology knowledge base, the ontological structure (a mirror image of the ontology instances), for example, the actual ontology class or relation definition of each instance, can be linked to form a unique structure, which is in fact an instance - MSM for this particular scenario instance. The results can be generalised to produce a real MSM that can then be stored in the DCMF-R (DCMF – Repository).

#### *Phase 4- Knowledge Use and Dissemination:*

*Inputs:* The generalised and re-usable MSM structure, as well as the ontology knowledge base structure (schema) and the set of scenario instances in the ontology.

*Outputs:* We then also process further to deduce a generic procedural structure – the MSMs that can be represented using different notations, like UML activity diagrams, BPMN<sup>20</sup>, Petri Nets and finally stored using KM3 in the MSM database.

*Analysis Process:* In the knowledge use and dissemination phase, we have investigated several possible uses for the processed scenario information. Some of them being:

- The MSM structure for each scenario group analysed, either in the ontology or in the KM3 Based, MSM Repository, is machine-readable, but not human understandable. So, we have visualised two different means of representing the same either in abstract generic English text or in graphical form using UML activity diagrams or other notations like BPMN. The formalism for generating the generic English text specification is something to be developed in the future. The primary target group of users for the MSMs are the simulation model developers for whom the proposed MSMs as both a graphical notation as well as machine-readable XML or OWL versions, is definitely an advantage.
- Another use may be for determining the compliance or error checking. That is one the MSM for the scenario is deduced, then the same can be used as a *process model* which is the standard procedure to be followed in similar scenarios. Thus, when the actual procedure deviates from the standard procedure, a tool which raises automatic warning or other signals may be designed.
- Another use is that each of the individual scenario instances may be used as composable units to compose a bigger military operation (MSM).
- The existing MSM may even be referred as a guide for those looking for directions to plan their next course of action.
- Case-based reasoning may be developed in the future, once an extensive collection of scenario analysis has been performed. Then based on this ontology knowledge base, decision support system like tools may be developed, which can match and propose possible MSM (operating procedure) for a given scenario situation.
- Some other possible targeted users and uses have been discussed in earlier reports [Moj03, Moj04]

### **9.3 Walk Through Example of Scenario**

After the detailed description of each step in the previous section, we now exemplify the above mentioned steps by providing a ‘walk-through’ analysis of the chosen scenario.

#### *Phase 1- Knowledge Acquisition:*

*Input:* We had some video clippings and in-depth interviews carried out with subject matter experts for further clarification and enrichment of the scenario description.

---

<sup>20</sup> BPMN - Business Process Modelling Notation. Details available at [www.BPMN.org](http://www.BPMN.org), last accessed on 14th October 2005

*Output:* The following is an extract from the scenario description in natural language is:

“A Swedish patrol from a battalion in Cosovo finds weapons in the forest near a village called Janjevo.  
The finding is reported in the battalion’s intelligence report and this is transferred in code to Stockholm. The information about the finding is received by the Intelligence Division at MUST and the report is registered in the System. The information about the found weapons is made available for the department of international intelligence (MUST IntUnd).”

**Table 9-1:** Scenario Text- Paragraph 1

*Phase 2- Knowledge Representation:* Explicit/implicit knowledge from natural language representation:

*Input:* The scenario description document in natural language (English).

*Output:* Some of the implicit knowledge extracted from the same text as discussed in phase 1 maybe tabulated as in table 9-2 below:

- There is a PSO in COSOVO sometime in MAY 2002, of which the Swedish contingent is a part of. (*Inferred from background context material*).
- Janjevo is a geographical location in COSOVO.
- There is a forest near Janjevo.
- Swedish troops go on regular patrol missions
- There is a procedure (military) to be followed by *any* military PATROL if they are on a *patrolling* mission. It also implies that there would be standard operating procedures and regulations governing this process of patrolling.
- ‘Swedish’ implies that Sweden is a sovereign nation, and that it has military capability, and is part of the UN Peace Support Operations.
- Weapons are hidden, that is, they are obscured from normal sight and they are not left for public viewing.

**Table 9-2:** Sample of Implicit Knowledge Inferred

The explicit knowledge can be extracted using 5Ws or KM 3. This year we carried out both these options separately to check if same knowledge can be extracted by these different methods. In table 9-3 we list the 5Ws extraction of the scenario text excerpted in Table 9-1:

Text		
Id	1	
Who	Patrol from Swedish contingent KS05	Patrol: Military Type Organisation (under govt type object-type: Unit-Type : has Affiliation object type relation to Swedish Contingent: Object-Item-Group. Swedish: affiliation
What	Patrolling	Patrolling: action-task purpose WHY AOI : Location
When	From 1 <sup>st</sup> may to 31 <sup>st</sup> May 2002, [1, 1]	
Why	To secure the AOI (Area of	
Where	AOI somewhere in Kosovo	AOI : specification detail AOI is both a Location as well as a CONTROL FEATURE: may even be a sub type of control –feature like ROUTE etc

Table 9-3: Explicit Knowledge extracted using 5Ws methodology

Table 9-4 presents an extract from the knowledge components identified via the KM3 methodology:

Entity Type :: Swedish patrol  
Entity Type :: Contingent in Cosovo  
ElementComposition : <Swedish patrol , Contingent in Cosovo>  
Entity Type :: Weapons

Entity Type :: Forest  
Attribute : Close to ET:Janjevo, in kilometers  
CompositionType : RangedAttribute  
Domain : Distance  
StartValue : 0  
StopValue : 10

Entity Type :: Janjevo  
Attribute : Village in Cosovo  
CompositionType : RangedAttribute  
Domain : Inhabitants  
StartValue : 100  
StopValue : 1000

Action Type :: Finds  
Time : May 2002  
RoleInAction : <finder, patrol>  
RoleInOrganisationType : <patrol, ET: Swedish patrol>

Criterion :: SF: (prob 1, isStartCriterion t,  
[Swedish patrol : onPatrol AND Forest AND Weapons])  
State: found weapons

ActivityState : Finding weapons (has occurred) /\* activity Finds has occurred \*/

<i>Entity Type</i> : Swedish patrol <i>State</i> : alerted AND onPatrol <i>Entity Type</i> :: Weapons <i>State</i> : found
-------------------------------------------------------------------------------------------------------------------------------------

**Table 9-4:** Explicit Knowledge extracted using KM3 Methodology

Detailed explanations of the different components like ‘Entity Type’, ‘Action Type’, etc., can be found in chapter 6 and Appendix A.

*Explanation:* A Swedish patrol finds hidden weapons in a forest near Janjevo in Cosovo. The explicit knowledge can be identified and extracted by using different parsing methods and tools like the 5Ws, SPO and KM3. In case of 5Ws we get who, what, why, when, where concepts. We do have still some open issue regarding the exact methodology to be adopted and the extent to which we should endeavour to capture implicit knowledge. But we hope to resolve these issues in the ongoing project.

*Phase 3: Knowledge Modelling:* Semantic mapping of the semi structured information in to the DCMF-O. We merge the implicit knowledge with the explicit knowledge in a machine readable format.

*Input:* The explicit and implicit knowledge tables available as output from the previous phase. (Tables 9-2, 9-3 and 9-4)

*Output:* A set of instances of the particular scenario created in the DCMF-O by following the transformation rules and guidelines, discussed in detail in section 10.2.2.1.

In table 9-5 below, we give a partial list of all the concept (class) types whose instances were created for the sample scenario text (table 9-2) discussed above.

<b>Instances created for the following classes:</b>
Action-Task
Action-Task-Status
Action-Objective-Type (securing AOI)
Action-Required-Capability (for patrolling mission) related to
Action-Event-Status: (through this it is associated to action-event. Finding weapons in a particular action-task: patrolling.)
Reporting-data
Action-Event
Object-Item-Group-Account: (the composition or relation of object types involved in the patrolling action.)
Capability: sub type: Mission-Capability: (specifies required parameters for carrying out a patrol.)
Affiliation
Context
Location
Control Feature
Action-Temporal-Association: time events, sequences and info for placing the action tasks and events in temporal sequence
Object-Type:Equipment-Type:Non-Consummable-Equipment-Type:Weapons



**Table 9-5:** Instances of Ontology Concepts

A sample extract of the actual OWL format code looks like table 9-6:

```
<Action-Required-Capability rdf:ID="patrolreqdqty">
  <quantifies>
    <Mission-capability rdf:ID="patrolmission">
      <capability-id
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >cmmscapability2</capability-id>
      <capability-unit-of-measure-code rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
      >square-metres-per-hour</capability-unit-of-
measure-code>
      <is-quantified-in rdf:resource="#patrolreqdqty"/>
      <capability-subcategory-code rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
      >maximum-Range</capability-subcategory-code>
      <capability-category-code
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >military-load-capability</capability-category-code>
      <capability-day-night-code
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >day-and-night</capability-day-night-code>
    </Mission-capability>
  </quantifies>
  <capability-id
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >cmmscapabilityid1</capability-id>
  <is-minimum-required-for
rdf:resource="#actioneventstatus1"/>
    <action-required-capability-quantity rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int">15</action-required-
capability-quantity>
    <action-id
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >patrol_reqd_qty</action-id>
  </Action-Required-Capability>
</requires-as-a-minimum>
<is-based-on>

. . . . .
```

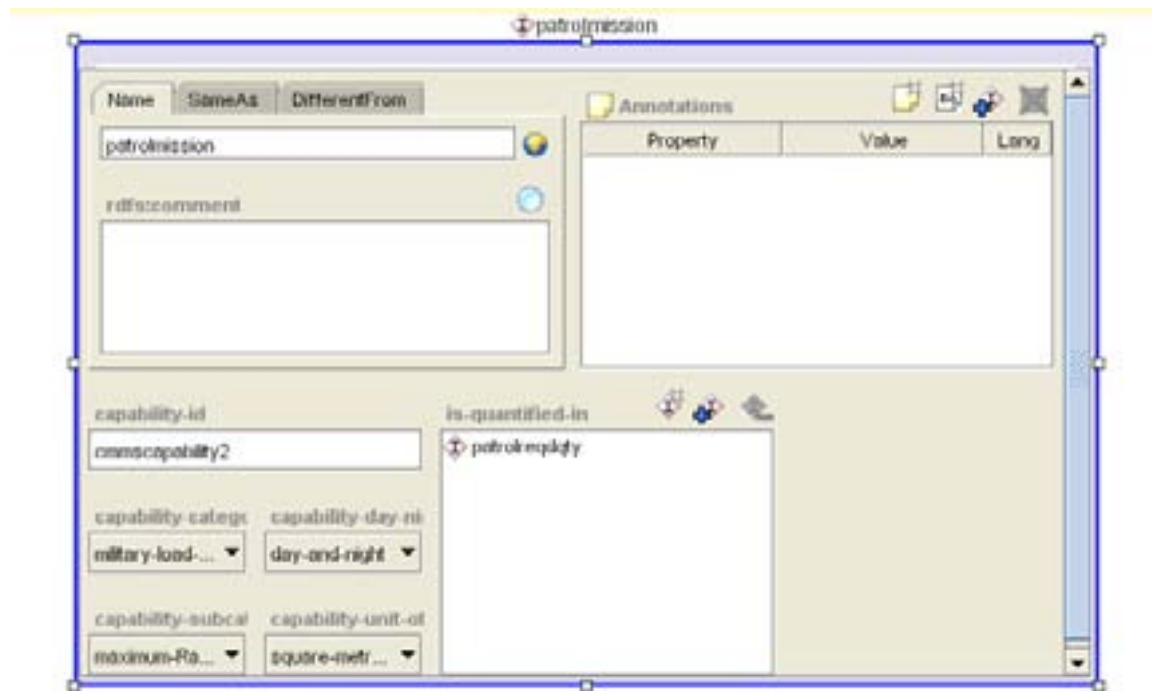
**Table 9-6:** Sample OWL format code for the Scenario paragraph 1

*Explanation:* By following the guidelines for mapping between the 5Ws concepts and the DCMF-O (more specifically the JC3IEDM adaptation for the middle level ontology) we get the first mapping results.

We need a more formalised methodology, which is currently under research. We identify the matching top level concepts in the ontology and then progress deep within the vertical tree branches till the highest match concept is determined. Several matching algorithm and techniques for doing the same automatically are available and these need to be reviewed and studied in the ongoing research.

Now we have captured the scenario as an ontology instance represented using OWL format, which is a fully *structured and machine-readable format*. Figure 9-4 below illustrates how the instance objects for the Swedish patrol looks like in the ontology editor tool Protégé. We see

that the Patrol unit object is identified by its ID, and we can select its capability category from the pre defined set of values. We can link it to the capability object created separately. We can choose the day-night capability settings etc. All these 'attributes' are defined in the JC3IEDM. Most of the default values listed in these attributes have been extracted from the suggested or recommended set of values as described in the JC3IEDM itself. We propose to extend these set of attributes in the next phase to suit our targeted scenario groups.



**Figure 9-4:** Swedish Patrol instance creation in Protégé Ontology Editor Tool

*Phase 4- Knowledge Use and Dissemination:* In this phase, we have visualisations and ideas of possible outputs and their potential usage. However, in this year we have not been able to focus on the actual methodologies or approaches required to make the transition to these visualised outputs.

*Visualised Use 1:* Abstraction on the structure obtained from the above ontology mapping implied in natural text: based on the instances identification, we can generalise on the conceptual level as:

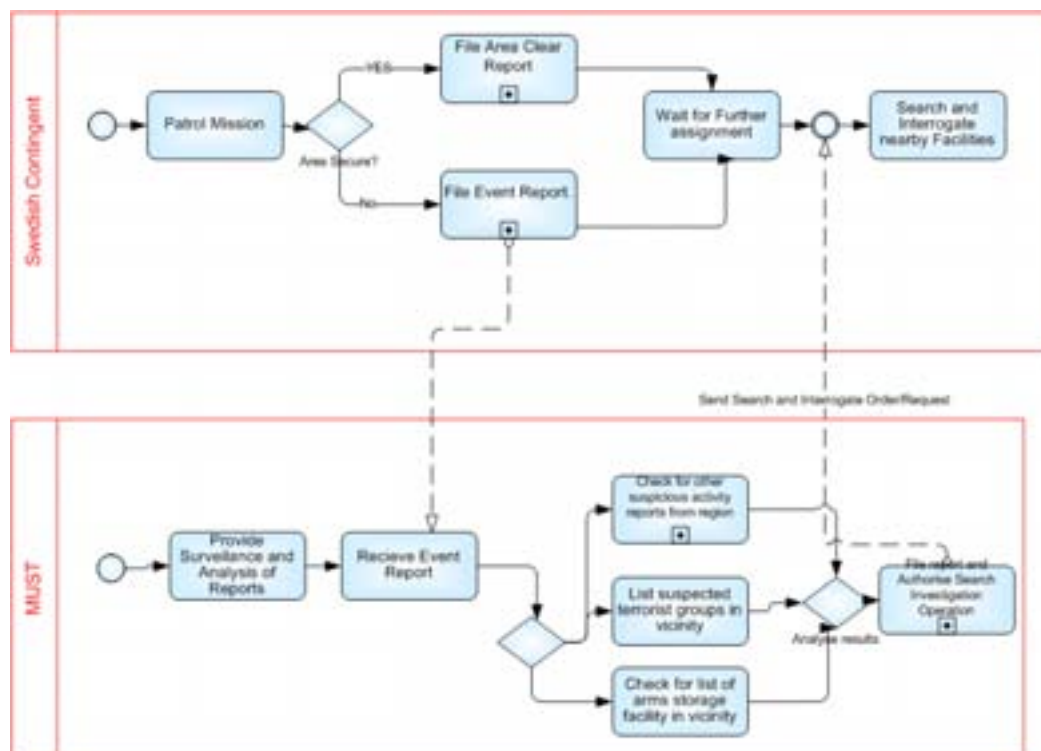
*A military-type organisation unit with affiliations to a particular political group while performing an action-task of patrolling a specified area of interest (a location) which is the current control-feature for the overall military operation of peace keeping in Cosovo finds hidden weapons.*

*Visualised Use 2:* The natural language text may not be of much use to the simulation modeller or designer. Or to the mission commander who needs to visualise and plan his military operations. Therefore a graphical form of the same may be visualised just as a process flow or procedural structure. The choice of actual representation language or tool could vary. We may use:

- UML activity diagrams for indicating the relationships between the different 'action' components.

- UML sequence or choreography diagrams to model the communication between the different military or non-military organisations involved in the operation.
- Other formal notations like Petri Nets or Event Process Chain diagrams may also be used.
- BPMN (Business Process Modelling Notation) can also be used alternatively. BPMN is graphically more expressive than UML activity diagrams and also can embed formal conditions, logic constraints within its 'attributes'. (We have used BPMN to illustrate some of the visualised uses).

The Scenario has been 'instantiated' as a series of small MSMs, for each group of analysed sub-scenarios. In the overall scenario analysis, we see that there are several organisations interacting with each other in the operation to track and recover the missing weapons. First, after the Swedish patrol has come upon some hidden weapons, they report back to their battalion unit in Cosovo. The information is then forwarded to the MUST handling officer who processes the information and sends back the next 'action' to be carried out by the Swedish contingent stationed at Cosovo. (We do not go in to more extensive details here). The above illustrates clearly that there is not only a certain action-centric chain of events occurring but also a clear communication and exchange of protocols, orders, information, etc. Figure 9-5, below depicts part of this interaction between different organisations. We have used BPMN notation and conventions [BPM05].



**Figure 9-5 :** Viewing interoperation between different military organisations in the scenario

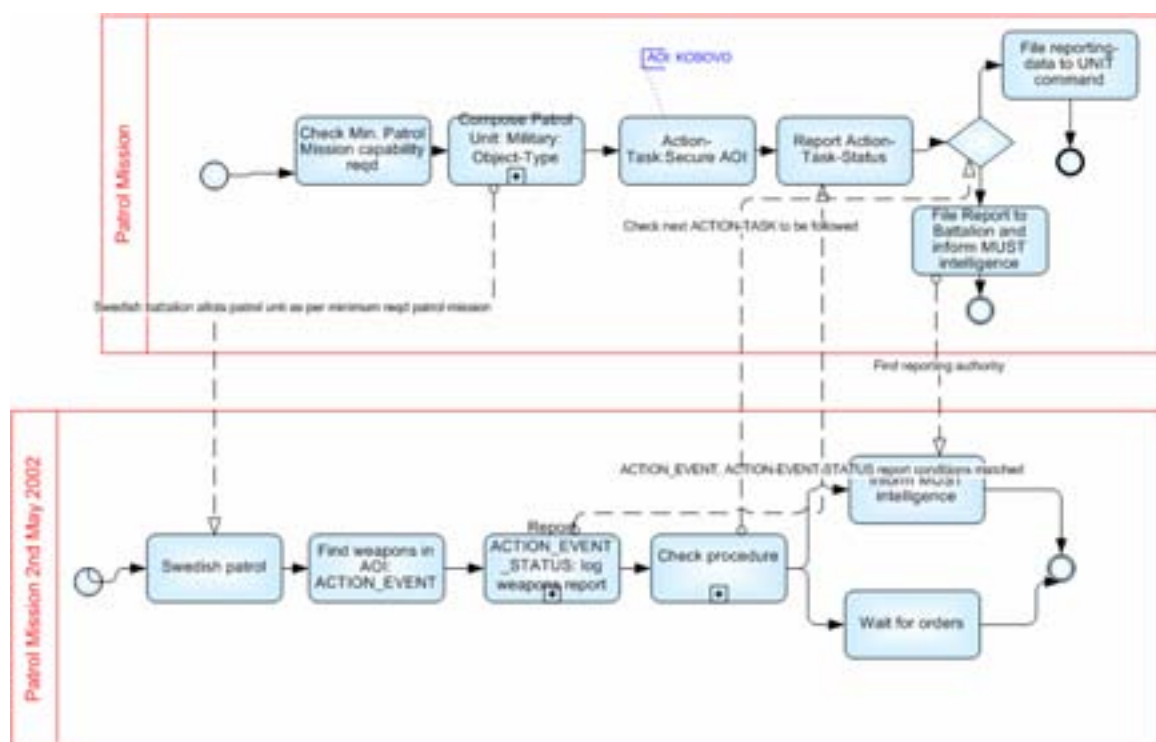
The different organisations are depicted as *swim lanes* (swim lanes are the horizontal division of the entire process diagram). In the figure 9-5, we see the Swedish patrol and the MUST shown as two swim lanes. The rectangular boxes represent actions or may even represent a sub-procedure or group of actions (plan for a sub operation), etc. The horizontal arrows depict the sequence flow and we may add temporal constraints by using

clock like notations for time. The dashed vertical arrows between different action components in the two swim lanes indicate the point of interaction and the kind of communication being exchanged. The diamond shaped forks are used for 'gate ways' or decision making control points.

This was an example where we have modelled and used the scenario (or its sub-scenario to be more precise) to see how the different organisations interoperate. Thus, this is a *case-specific view*.

*Visualised Use 3:* In the previous usage visualisation, we analysed the scenario from within its own level of abstraction. In another visualisation, we propose that we may generalise the sub-scenarios to a higher conceptual level to derive a generic outline for procedures. Several of these may be combined together to provide an operational plan for a certain type of scenario. Thus, in this case we move from the *specific* to the *generic* MSM.

Again, we use BPMN as the representation notation for the current visualisation. In this case, we use the different swim lanes to model different levels of abstraction. In figure 9-6 below, we see a generic pattern for *patrol mission* on the top swim lane and the particular case of the scenario is depicted in the bottom swim lane.



**Figure 9-6:** Comparing Scenario procedure to recommended Procedure

This kind of abstraction may be used for a multitude of uses:

- It provides the procedural plan to be followed. We could model the rules of engagement as one such model. The generic procedure plan can then be referred to as an indicative, informative planning guide for a particular instance. That is, the Swedish contingent commander may look at the generic patterns (MSMs) to decide what the suggested next course of action is.

- In another advanced usage, we may use these as a conformance checking tool. That is we may compare the generic patterns to the represented actual execution of a particular scenario. Then we can compare to check for consistency, deviation from the standard generic protocol, etc. Other kinds of decision support and analysis may be based on these basic interpretations.

In this chapter we have taken a closer look at how a military operation may be captured, represented, modelled, used and finally reused. We have illustrated through a case study the different phases of our proposed DCMF Process. We have seen how the different components discussed in this report, KM3, 5Ws, SPO and the DCMF-O all co-relate to each other and interoperate. In the next chapter, we summarise some of our research experiences, contributions for the current report and open issues.



## 10 Experiences gained

In this report, we began with a discussion on the objectives for the DCMF Project and related it to its origins in the CMMS concept. We continued by discussing the overall goals and components of the DCMF Process. Thereafter, we moved to a detailed discussion on some of the identified components, specifically the KM3 and the DCMF-O. We have presented summaries of relevant theories and models used (e.g., 5Ws, SPO techniques, SUMO, JC3IEDM, etc.). Finally, we have illustrated our proposed DCMF Process through a case study evaluation.

During this process, we have gained valuable experiences of various aspects of the DCMF Process. These lessons learnt have provided us with insights and motivations for future work. In some cases our assumptions and hypotheses have been proved correct in other cases they have been incorrect. We have found areas where further research is needed and areas where we have got a better idea of in which research directions to proceed.

In this chapter, we present a series of discussions on the experiences gained in different phases of the DCMF Process. In section 10.1, we discuss the results and open issues in the KM3 component that was gained as a result of relating the KM3 to an ontology. In section 10.2, we discuss the issues we faced in the process of modelling the JC3IEDM as an ontology. Finally, in section 10.3, we discuss the process of domain ontology mapping and/or merging.

### 10.1 KM3 Results Discussion

This section concerns the mapping of an ontology to the KM3 and the results from doing it. To map means, in this context, that constructs of the ontology (classes, relations, etc.) are to be mapped onto the constructs provided by the KM3. Another way of putting it is that the ontology should be expressed in terms of the KM3. Such mappings are never easy tasks and need almost always (unless the languages are trivial) manual work from people working in an iterative fashion. This iterative way of working can be called, when discussing this particular task, semantic calibration. The main idea behind it is that one mapping is produced and subsequently tested. Depending on the results of the tests the mapping is adjusted, and further tested until a satisfactory result is obtained.

In this section we are looking at an ontology called Joint Control Command Consultation Information Exchange Data Model (JC3IEDM). Why this data model is referred to as an ontology is explained in section 5.4.3. The JC3IEDM is presented more in detail in section 5.4.3. The JC3IEDM is to be mapped onto the Knowledge Meta Meta Model (KM3) (see section 6). One of the main ideas of the KM3 is to be able to express information interpreted and encoded in one way (i.e., by some ontology) and render it in another way (the KM3 way), thereby making the information accessible for use by tools operating on KM3 structures.

The JC3IEDM is stratified into three layers; the conceptual model, the logical data model, and the physical data model. In this section we are interested in mappings between constructs of the conceptual model layer and how they are mapped onto the KM3. We are not concerned with, for instance, issues of data type mappings which are properly dealt with in the physical layer.

The KM3 has four major parts; Model element, Attribute, State, and Rules (see section 6.3). Through these parts we are able to identify things and actions as objects (using model

element) and the relations between them, describe the general characteristics of those objects through by stating their properties (using attribute), describe the current values of attributes (using state), and formulate the dynamics of the objects (using rules).

The conceptual model layer of the JC3IEDM consists of 15 constructs (see section 5.4.3), connected by a two kinds of relations (association and aggregation). Additional kinds of relations are given in the textual definitions of the model. The full KM3 consists of 32 classes with two kinds of relations (generalisation and association), with a possibility to define additional ones. (for detail see appendix A.)

In the text below, classes from the JC3IEDM are written using upper-case letters (as they are in the JC3IEDM diagram). The classes of KM3 are written using a mixture of upper and lower case letters.

We start the mapping by the central concept 'action'. It is called ACTION in JC3IEDM and ActionType in KM3. It is unproblematic to map ACTION of JC3IEDM onto the ActionType of the KM3. An action can be decomposed into subactions in both models, and the associations present in the JC3IEDM (e.g., links to REPORTING-DATA, CONTEXT, etc.) can be described in KM3 through the use of the ElementAttribute class. One important function of this class is to be able to define missing classes as model element attributes.

It is unclear whether the ACTION construct is at a 'type' or an 'instance' level. Does it capture a particular action or an action in general? Most likely, both. It is associated with the OBJECT-ITEM class, indicating that it is on an instance level. It is also associated with the OBJECT-TYPE class, suggesting the type level.

The OBJECT-ITEM class is mapped onto the ElementState class of the KM3. The intention of the OBJECT-ITEM is to describe an object as having for instance an ADDRESS. This is also what the ElementState is for; to describe an instance of a model element through the values of its attributes.

The OBJECT-TYPE class is mapped onto the EntityType class of the KM3. This mapping is unproblematic considering not only the choice of names but also the associations that they both share. An OBJECT-TYPE is associated with an ACTION through an intermediate class CAPABILITY. Likewise, the EntityType is associated with a ActionType but in a different way. The association involves two different classes to be complete.

1. First, the EntityType is linked with the RoleInOrganisationType whose intention is precisely that of CAPABILITY. It exists to state what actions an entity is capable of performing (in the case of KM3, the recommended way of doing this is by defining a role in terms of rights and prohibitions).
2. Second, the RoleInOrganisationType is associated with the RoleInAction class. The intention is to be able to state whether an EntityType role is, for instance, the subject or the object in an action.
3. Finally, the RoleInAction is associated with the ActionType class and thereby closing the link between an EntityType and an ActionType.



The link between action(-type) and the object type is there both in JC3IDM and KM3. In KM3 an additional segment is added to the linkage, though.

An ACTION and an OBJECT-ITEM is not only connected through the classes and links described above, but also through the class RULE-OF-ENGAGEMENT.

The connection has a correspondence in the KM3 where the ActionType class is connected with an ElementState class through the Criterion class. The intention with this class is to be able to formulate rules that govern actions. The formulations of the rules are more detailed in the KM3 than in the JC3IEDM. The KM3 prescribes that rules concerning the dynamics of actions are to be formulated in a particular way, resulting in a cluster of classes to capture this. In the JC3IEDM no such attempt is present at the conceptual level. The classes Criterion and RULE-OF-ENGAGEMENT are mapped onto each other. Table 10-1 summarises the main mappings of the JC3IEDM onto the KM3.

JC3IEDM	KM3
ACTION	ActionType
OBJECT-ITEM	ElementState
OBJECT-TYPE	EntityType
RULE-OF-ENGAGEMENT	Criterion
CAPABILITY	RoleInOrganisationType
REPORTING-DATA	ElementAttribute
CONTEXT	ElementAttribute
CANDIDATE-TARGET-LIST	ElementAttribute
REFERENCE	ElementAttribute
LOCATION	ElementAttribute
VERTICAL-DISTANCE	ElementAttribute
COORDINATE-SYSTEM	ElementAttribute
ADDRESS	ElementAttribute
GROUP-CHARACTERISTICS	ElementAttribute
AFFILIATION	ElementAttribute

**Table 10-1:** Mapping of JC3IEDM and KM3

This is a high level mapping of the concepts. One feature instantly recognisable in table 1 is that there is a lot of classes in the JC3IEDM that are mapped onto one class in the KM3. This is a decision of the constructor of the table. There is often a possibility to model concepts as being either attributes of classes, or as being proper classes in themselves. Being proper classes they may be associated with other classes. In this case, for instance, the JC3IEDM class CANDIDATE-TARGET-LIST has been mapped onto the ElementAttribute class. It may very well be equally correct (at least on the surface) to map the CANDIDATE-TARGET-LIST onto the EntityType class. Subtle formulations in the model specifications requiring a deep analysis, or simply pragmatics often decide the final result.

Other examples of problems when mapping from one model to another, are when one concept maps to a combination of several concept. Or, only in part maps onto a concept. Or, do not

have any correspondence. If these problems occur, as they often do, the iterative way of working (the 'semantic calibration' as it was called above) may be the only way to solve them.

### **10.1.1 The relation KM3 and MSM (Mission Space Model)**

A major research field within computer science, called knowledge representation, has been devoted to questions concerning what representation is appropriate when solving different classes of problems. The hypothesis is that an appropriate choice of representation simplifies the solution of a problem. Representing the knowledge in one way may make the solution simple, while an unfortunate choice of representation may make the solution difficult or obscure.

The KM3 is presented in figure 6-1 as a UML Class diagram notation where each of the boxes represents classes and the lines between them represents different kinds of relationships (aggregation and generalisation). All aggregation relationships have cardinality constraints associated with their endpoints to be able to express that, for instance, a member of one class is related to many members of another, or that a member of one class is always a member of another.

The UML class diagram (which has the form of a graph), can be transformed into many other representation forms. For instance, all information put in there can be changed into a free text format while still retaining all the original information. Another form of particular interest is to use XML. The benefit of this representation form is that there exist many supporting tools for manipulating information in XML. For interoperability in and between systems the storage of information in a structured text format is beneficial as well. Most systems can open and operate on information encoded in a structured text format.

There exist currently at least one tool that supports automatic transformation between UML Class diagrams and XML. The tool is called Protégé (and the appropriate plug-in for the tool is called UML Backend). It can import UML models constructed in other modelling tools, e.g., Rational Rose or Poseidon, and save the models in an XML format. This is possible since all mentioned tools support the XML Meta-data Interchange (XMI) specification [XMI05]. The relevant consequence of this is that the KM3 model represented as a UML Class diagram can be transformed into a XML structure using these tools.

Generalised knowledge modelled in accordance with the rules of KM3 is called a Mission Space Model (MSM). This model is populated with real or hypothetical data to become the base of simulation. A simulation has a number of results; a) it increases the understanding of what is being simulated, b) it may be used for predicting outcomes, c) it may be used for optimisations, and d) it may be used for validation and verification of the simulation model. The relation of the KM3 with respect to the MSM is to provide rules for constructing well-formed MSMs and thereby provide the means for achieving those simulation results.

## **10.2 DCMF-O Design and Development Experiences**

As discussed in Chapter 5, we found the JC3IEDM to be a suitable candidate to be used as a middle layer ontology (part of the DCMF-O) in the DCMF. However, the data model of the JC3IEDM is not readily adoptable as an ontology. We discuss the limitations of it in the following section.

### 10.2.1 Limitations of the JC3IEDM Data Model

In Chapter 5, we have reviewed the background, and basic concepts described in the JC3IEDM data model. Its intended purpose and detailed level of analysis renders it as an excellent starting point for our DCMF-O knowledge base development. It precludes the necessity of extensive research to gather and extract domain knowledge from dispersed resources. However, there are some limitations or weaknesses inherent in the JC3IEDM data model, as listed below:

- We found that several of the operational or procedural information has been recorded in the specification but has not been amply represented as logic constraints in the JC3IEDM data model. One of the fundamental reasons for this could be that the original intention for this specification has been to support *data interoperability* between different information systems. This is a point of divergence from our project goals, which are on a different perspective. We intend to use our ontology base to support several features including inferencing, operational conformance checking, suggesting suitable course of actions etc. We may use our proposed ontology to support interoperability of information systems, but that is not the *primary* objective for the DCMF-O.
- The JC3IEDM is a *data model* and does not follow the norms of object oriented modelling. Also several conceptual definitions are unclear and fuzzy, like a type level concept ‘Object-Type’ and again an individual identification of the same type, as ‘Object-Item’. But again, we found some domain specific modelling constraints. In the military operations domain where the whole analysis is *action centric*, the need to distinguish between a *type* of an object example: a tank is a type of equipment) and the particular named object *item* (example: the tank with a given ID) is well established. This demarcation is however, not precise for typical object oriented information system modelling, where a class is generally a definition of a type and an instance of a class refers to a specific individual. A direct one-to-one transformation from the JC3IEDM to the DCMF-O (Chapter 5) implies that we model both Object-Type and Object-Item as separate classes. In the current version, we have maintained the separate entities. But it has led to several technical implementations (ontology instantiation) problems, when we tried to capture actual scenario information. To ‘instantiate’ the individual scenario information in to the ontology, we need to create an instance of *both* the Object-Type and the Object-Item. It led to duplicate information being stored and quite redundant. For example: when we want to model the Swedish Patrol KS05, then we have to create a object-type of the identity of Patrol of Swedish affiliation, and we have to create a specific Swedish patrol with the call sign KS05 as well.
- Also, a careful examination of the deeper levels of the JC3IEDM model reveals that not all the concepts modelled may have been accurately represented. That is, there could be alternative expressions, differing from Subject Matter Expert (SME) to SME. Thus, a case by case review of certain concepts would be best advised.
- The JC3IEDM has evolved from the LC2IEDM data model specification, and is as such still a specification under revision/development. For instance, the land operations concepts have been extensively modelled, the same cannot be said of the other proposed *joint operations* branches, like air operations, intelligence, naval etc.
- Another limitation is that the existing model is not evenly balanced. That is certain concepts have been explored and modelled in depth like for example, an extended classification of ‘Facility’ under ‘Object-Item’ has been broken down to further sub – classes to the level of Harbour, Quay, Jetty. Whereas, on the other hand several important concepts have not been explored enough.

- Another major limitation for a direct use as an ontology is that the JC3IEDM data model does not hold all the ‘business rules’ inherent in its data model. For a relational database, this holds no problems, as these would be readily implemented as ‘triggers’ and business logic in the front end application. However, we would expect a fully formal ontology (as suggested by the ontology definition we have adopted) to hold all relevant logic constraints within itself. The business rules have been stated in the JC3IEDM data specification and for the moment, we choose to directly adopt them and represent them as *axioms* in our Ontology using Description Logic statements. But on the other hand, it also allowed the main structure to be very flexible and generic, that is to say, that not all possible military group objects had to be pre-defined in order for us to capture the required information about a Swedish Patrol Unit named KS05. That is we did not need to have ‘patrol’ as a pre defined type entity in the structure. Thus, the knowledge base is generic and flexible and is constantly evolving. And once the ‘patrol’ object-type has been created, it can be reused to create other object-items, say a Norwegian patrol, or a Swedish contingent with a different call sign, etc.

## 10.2.2 Our Design Philosophy to Adapt the JC3IEDM

We have discussed some of the limitations of the JC3IEDM data model. And we have explained the need for modifications required to express the JC3IEDM as ontology. We were faced with two tasks:

1. To form a set of transformation guidelines for adapting the existing JC3IEDM into an Ontology. For this we have proposed some guidelines for mapping and transformation.
2. To render the adapted JC3IEDM to suit our specific targeted purpose of our project. That is to suit the needs of the DCMF goals.

### 10.2.2.1 Transformation Guidelines from JC3IEDM into DCMF-O

We designed simple rules for transforming between the IDEF0 representation of the JC3IEDM and our OWL representation. IDEF0 notation is the notation used in figure 5-4, and OWL is a form of XML proposed by the W3C as a way to describe ontologies.

In the first step, we chose to draw UML class diagrams, to enable us to understand the concepts and facilitate our further development. Then, we followed these rules of thumb:

- As far as possible a direct one to one transformation is to be carried out. That is, a class in the original model is to be modelled as a class in the UML and thereafter in OWL formulation of the ontology (the target).
- All attributes of the JC3IEDM are to be modelled as Data Type property in the target ontology.
- Associations in the JC3IEDM have to be analysed carefully and then modelled as Object Property in the target ontology.
- Redundant connecting tables, required for normalised database schema design are to be dropped. (However, for the current proof of concept, this has not been completely carried out).
- Foreign Fields and internal index, attributes, required for relational data bases, need not be transferred to the target ontology. All ontology concepts have their own internal identifiers and are linked essentially through their object relationships.
- Sub category codes have been introduced in the JC3IEDM to split Data base tables in to Parent-child (one to many or many to many) relationships. These may be removed

or modelled differently in ontologies. A sub class inherits all properties of its parent, and then we may modify its behaviour.

An example for the last mentioned case is given below:

4.8.1 The category codes in the OBJECT-TYPE hierarchy are not permitted to be changed. New instances of OBJECT-TYPE must be created with the appropriate category codes within the OBJECT-TYPE hierarchy.

4.8.2 If the value of the organisation-type-command-function-indicator-code is “Yes”, then the organisation-type-command-and-control-category-code cannot be NULL.

**Table 10-2:** Business Rules Extracted from JC3IEDM

In the above case we propose that sub categories of Object-Type be modelled as a sub-class under Object-Type. That is, we use the Parent-Child inheritance property of the ontology to establish the relationship between the parent ‘Object-Type’ and the different children ‘Organisation-Type’, ‘Materiel-Type’, ‘Feature-Type’, ‘Person-Type’ and so on. Thus, the child class automatically inherits all properties as specified in the parent class. Additional properties may be added for each child. Or we may use ‘Restriction’ axioms on the inherited properties to modify or condition certain features for the child class. Thus, we have made use of some of the stipulated Business Rules as Description Logic axioms or restrictions to *modify* the JC3IEDM structure into our DCMF-O.

### 10.2.2.2 Enrichment /Adaptation of JC3IEDM for MiSO

The Military Specific Ontology (MiSO) was presented in Chapter 5 as being the method to make the ontology part of the DCMF. As such, any ontology should be aligned to according to MiSO if it is to be included as a part of the DCMF. JC3IEDM is an ontology that needs to be aligned accordingly. We chose to concentrate only relevant portions of the JC3IEDM. This was due to two reasons. Primarily because we were focusing on describing military operation scenarios which revolves around actions, their resource allocations, and so on. Secondly, this is our first iteration for the design and development phase for our proposed ontology. Therefore, we are exploring all possibilities not taking into consideration their feasibility for design and implementation. Also, at this stage we are interested in establishing a proof-of concept working environment. No doubt, in our next iteration, we shall extend or adopt the entire range of specification as given in the JC3IEDM.

We also need to extend certain parts to enrich the ontology. Wherever a vertical extension (that is, increase in depth. A specialisation) was necessary we have modelled them as separate domain ontologies, which can be plugged in whenever necessary. Where a horizontal extension to the JC3IEDM was required, we have added those classes or in several cases merely properties (data type property and object properties). When we move from a subjective, context and application specific description to an abstract, generic and objective description of the domain, we are in fact moving on a /vertical /plane. So, it is possible for us to add more ontology layers above or below the current 3 layers that constitute the DCMF-O (Fig 5-2). Again, we have considered only JC3IEDM as a candidate for the middle layer, and as mentioned earlier, we may have other middle level ontologies like the IEEE MILO ontology as well. Similarly, we visualise that we may develop more context specific domain ontologies. These ontologies provide additional knowledge within the same level of abstraction, hence we add horizontal extensions.

We have added ample comments and documentation within the ontology itself to record the changes made. Thus, any other potential user of our ontology can be aware of the modifications made.

### 10.2.3 Sample Transformation Questions /Rules for mapping to MiSO

We need to formalise the mapping or transformation rules from the semi structured information output available after the first knowledge analysis and representation phase, to the structured ontology. We use the ontology first to provide us with a list of terms to look for in the analysed information. Once a top level probable term has been matched then we proceed to match with corresponding child classes till we reach the lowest possible class match. There are several automated or semi automated text extraction tools for this kind of syntactic or keyword based search. However, we propose to move towards a semantic approach, where the intended meaning of the term is to be matched rather than a specific keyword. We are aware that much work in this context has to be done by us. But for the current proof of concept case study analysis, we propose a set of guidelines or mapping recommendations which help the ontology designer to analyse and match relevant concepts from the 'semi digested' information from the 5Ws in to possible list of target concepts in the DCMF-O.

The 5Ws analysis provides the answers to: Who? Why? What? Where? When? We provide a possible top level mapping concepts from the JC3IEDM. In some cases, we also provide a set of secondary questions, which the ontology designer may use as a guide to clarify and seek further implicit knowledge. We do not go in to a detailed list of all the recommendations made, but only illustrate a few.

*Who?* The question 'who' from the 5Ws may be matched to:

1. Top level: Object Type or Object Item
2. Specific: Any of the specific objects types or object items. Note, the 'who' could also be a Feature type or materiel type,

*Why?* This gives the reason or context in which the action or operation is occurring or is being planned. In our current case study, the context for the overall scenario is "cosovo peace keeping". On the other hand, each individual 'sub scenario' or 'action-entity' from KM3 or the Action-Task from the DCMF-O can each be mapped to a specific 'CONTEXT' object as well as 'ACTION-OBJECTIVE' object.

*What?* What is being done? What has occurred? What has been the objective for the action? The 'doing' object of the input statement, which naturally leads us to the 'action' being done.

1. Top level: Action: Action Task or Action-Event
2. Secondary questions: What was the intention for the action? Answer can be matched to 'Context', 'Action-Objective' (if the objective is only for the current 'action'). Note that this also gives us the mapping from the *Why* aspect of the 5Ws analysis
3. Informational and data characteristics mapped to action-objective, also to reference-data for capturing the data.

*When?* When is the action taking place? Is in the past? Or is it in the present? Or may be scheduled as a 'plan' for the future? All absolute or relative temporal associations are mapped to Action-Temporal-Association. Also mapped to an associated Reference Data item for capturing related technical aspects.

*Where?* Where is the action or described course of actions taking place?

1. Can map to 'Location', (either absolute or relative positioning, 'vertical-distance', may also use specific co ordinates).
2. If the described where is a specific location, then it could be an 'Object-Type' or an 'Object-Item'. More specifically a Feature-Type or a Feature could describe the concepts from the 'Where'. Ex: a forest, a lake, an ocean, or a feature: the Pacific Ocean, the Alps, the Caspian Sea. In our case study, a 'forest near Janjevo' has two concepts. One a Feature-Type: 'forest', the other a Feature: 'Janjevo' and a relative Location aspect: 'near'.

Based on the above concept identification, we propose a transformation rule for the *Who* as: 'Who' answers the questions of a person, organisation, and role, authority *who* performs the task, activity, action, or affects the events. We summarise as a corollary in table 10-3:

```

May be one or more of:
Object-type OR Object-Item
If Object-Type THEN MAY be
Person-Type
Organisation-Type
CANNOT be one of
Materiel-type
Feature-type
Facility-type
Corollary: IF Object-Item THEN maybe Organisation AND/OR
Person.

```

**Table 10-3:** Example Transformation Rule

To summarise some open issues in the DCMF-O modelling context:

- We need to resolve transformation rules from a data model in to ontology.
- We need to adopt standards like the Ontology Definition Metamodel to facilitate automatic transformation from UML conceptual models in to ontology.
- We need to formalise the process from the knowledge extraction to knowledge representation and on to the ontology knowledge modelling. So far, we have been able to identify the missing links in the DCMF- Process.
- Some of the requirement criteria on the ontology knowledge base discussed earlier have not yet been focused on like credibility, repeatability, verifiability.
- Methodology for the DCMF-O to evolve conceptually with extraction of new concepts from new scenarios need to be developed.
- Finally, once we manage to format and structure the knowledge as a formal OWL ontology, we should be able to search and retrieve information as per our needs. Much research in to 'querying' mechanisms is needed.

### **10.3 Merging the ontologies**

In chapter 5 we presented an overall description of a number of ontologies; SUMO, JC3IEDM and the domain ontologies. In this section we will describe more about how we merged them, the theories and methods behind and the tools that we used for merging and mapping the ontologies.

### 10.3.1 Methods

Different ontologies can be connected to each other by either merging them or mapping them together. There is a difference between merging and mapping. Ontology merging proposes to generate a unique ontology from the original ontologies. Ontology Mapping is the process whereby two or more ontologies are semantically related at conceptual level by establishing different links (relations) between the ontologies. This is done by finding what related concepts of the ontologies there are, and establish the relations between them. [Noy99, Gom04]

What to prefer between merging and mapping depends on the final use of the ontology, but in most cases ontology mapping is suitable because it preserves the original ontologies which facilitate traceability and also interoperability and composability. For merging and mapping the ontologies we used a blend of the ontology methods described in chapter 4.

### 10.3.2 Tools

The merging and mapping can be performed manually but it is time consuming and error-prone and hence an expensive process. To automatise and facilitate those processes there are a number of tools developed, for example PROMPT and GLUE. [WBS04, Doa02]

To merge our ontologies we use the PROMPT plug-in tool available in Protégé editor. The PROMPT tool is an ontology-merging tool which gives guidance for the merging process making suggestions, determining conflicts and also proposes conflict resolution strategies. The tool can also support activities like extract a part of an ontology, move frames from included to including project, compare two versions of the same ontology and create a merged version. This tool allows us to do both merging and mapping. The reasoner in the tool suggests automatically possible classes and relationships to be mapped. [PRO05]

### 10.3.3 Open issues

Although there are methods and tools for mapping and merging ontologies those have to be studied and evaluated more. There is not really any standard but rather development in progress in different constellations. Furthermore, our ontology structure presented in this report, the DCMF-O, has not yet been neither verified nor validated. We are aware of that there exist inconsistencies and problems with the current DCMF-O, but in theory it is aligned with research and results in the Ontology area. It would be necessary in the future to validate all the parts of our ontology with SMEs. This will in particular be important for the domain ontologies that we have developed from sources in publications on military taxonomies. Most likely there could have been misconceptions when structuring the concepts and their relations in those domain ontologies.

As already mentioned mapping is better than merging since we only add relations between the ontologies and keep the original ontologies intact. This process is preferably done automatically with some kind of tool in order to minimise errors due to the human factor and also minimise the time cost. We used the PROMPT for this which worked out quite fine although there were some problems. To map the Swedish Defence Organisation Ontology with the JC3IEDM showed up to be non feasible because those ontologies are stated in to different languages. The Swedish Defence Organisation Ontology is expressed in Swedish and the JC3IEDM in English which led to inability for the reasoner to sense similarity between classes in the two ontologies. Therefore the mapping had to be done manually which



is not the optimal method and it is important to be aware of the consequences of having ontologies stated in different human languages. Sometimes it is of value to use different languages, for example in an interoperability aspect, but it requires reflection of what kind of problems it might arise.



## 11 Discussion, Conclusions and Future studies

The increasing use of modelling and simulation in the military domain puts high demands on how knowledge is used and managed. Major challenges are how to acquire, validate and maintain knowledge and how to achieve this with the minimum effort. To address issues relating to knowledge bases for modelling and simulation, the US DoD in 1995 introduced a concept called Conceptual Models of the Mission Space (CMMS). For unknown reasons the concept was never completed all related activities seemed to end around the turn of the century. However, the Swedish Defence Research Agency (FOI), found the concept interesting and has, since 2001, done research on the concept to explore its potential.

The work at FOI began in 2002 with an extended study of published material about CMMS. It was discovered early on that many of the specifications of the CMMS process were vague and unfinished. This meant that to get a clearer understanding of the concept, a basis for a common methodological framework had to be developed. The main objective of this report is to present this framework, now called the DCMF – Defence Conceptual Modelling Framework.

A plan for the work with a focus on the early phases of the DCMF process was established in 2003, and the following year subsequent phases were in focus. During the work it became obvious that a lot of the necessary components, methodologies and tools were missing to finish the process. Examples of fundamental pieces missing included a structure by which Mission Space Models (MSMs – the final outcome of the process), were describable. A proposal for such a structure was made and it is called the Knowledge Meta Meta Model (KM3). Another fundamental piece missing was an ontology structure. Such a structure was not mentioned in the original CMMS documents.

This year the work has mainly focused on making MSM prototypes and following through with the process to assess its feasibility and to gather experience. During this work we have also been able to further identify necessary tools, methods and techniques. We have applied the DCMF process on a hypothetical scenario and thereby done some validation of developed tools and theories. We have discovered a number of issues and gathered valuable experiences during this work.

We have formulated, but not finalised, a set of requirement criteria for a knowledge base to be used in the DCMF process. Those criteria have had an impact on the selection process for various available methods, tools, ontologies, etc. used in the process. We have studied and analysed several of the contemporary ontology design and modelling methodologies. Based on this research and the requirements put forward for the DCMF, we have created a methodology, called MiSO (Military Specific Ontology development), to develop military specific ontologies.

Using MiSO we proposed a multi layered architecture called Defence Conceptual Modelling Ontology Framework (DCMF-O) for modelling reusable knowledge for the military operations domain. We have also surveyed and compiled a collection of existing ontologies and other knowledge bases which may be included in the DCMF-O. They may be used there as Specific Domain Ontologies either as-is or after modification according to the MiSO framework. Examples include the Weapons of Mass Destruction, Terrorist, Vessels, and Geographical Features ontologies.

The ontology structure presented in this report, the DCMF-O, has not yet been verified or validated. There is no standard way of doing this today. To do it, available methods and tools for mapping and merging ontologies have to be studied and evaluated in more depth.

We are aware of existing inconsistencies and problems with the current DCMF-O, but in theory it is well aligned with current research and results in the Ontology area. We envision it to be necessary in the future to validate all parts of the DCMF-O with Subject Matter Experts (SME). This is particularly important for the domain ontologies that we have developed from sources in publications on military taxonomies. It is very likely that flaws exist due to on misunderstandings on our behalf when structuring the concepts and their relations.

We found that mapping ontologies is a better way to work than merging, since we only add relations between the ontologies and keep the original ontologies intact. Mapping and merging is preferably done with tool support in order to minimise errors due to the human factor and, since the work is labour intense, automating as large part of the work as possible minimises the time needed.

One task that we found problematic was to map the Swedish Defence Domain Ontologies with the NATO JC3IEDM ontology. This became difficult mainly because of language problems. The tools used could not work with ontologies in different languages. For example, there was some inability on behalf of the reasoner to sense the similarity between classes. The result was that mapping had to be done manually. The lesson learned was that it is important to be aware of the consequences of having ontologies stated in different natural languages.

The Knowledge Meta Meta Model (KM3) has been evaluated in several ways with mostly positive results. It has been found useful in a supporting role when interpreting data. This means that when confronted with ambiguous or otherwise unclear data, the KM3 can support an interpreter of the data by supplying concepts and structure. Furthermore, an interpreter may also use the KM3 to discover ambiguities in data. Another result is that the KM3 may be used to further structure semi-structured information. This ability is important when considering that one of the intended uses of the KM3, is to provide a way to structure information into a form that facilitates machine processing.

We have experienced some difficulties, as was expected, when using the KM3. Most of those difficulties are due to methodological support for doing the work. We have found the KM3 useful in the role which it was intended to fill and we see, at this point, no reason to discontinue the development of the model itself and the methods for using it. The KM3 has been used as an analysis tool and its full potential as a foundation for the reusable knowledge library has not been fully explored yet.

We have, for the first time in the project, chosen to illustrate and validate the proposed DCMF process by means of a case study. For this purpose, we chose one of several available candidate scenarios. Evaluating the DCMF process by using the scenario has identified several issues in the different phases of the process. The scenario was given in a free text form which needed to be read (parsed) and interpreted. We found that the most appropriate parsing method depends on the purpose of the activity and who is performing it. The results will also be different depending on the methods used. For instance, if two method experts analyse and formalise a common scenario description they will likely end up with two different formalisations. Future work involves designing stricter guidelines for the analysis and formalisation of information.

Using the KM3 as a structuring tool facilitated the interpretation and the structuring of data in a more detailed way than the other tried methods (mainly the 5Ws method). However, due to the lack of methodology for using KM3 as an analysis tool, the parsing activity was complicated when using it. At this stage, we found that the 5Ws and KM3 methodologies cannot be compared on an as-is basis. The two methods provide us with information which describes different semantic aspects. The 5Ws has the advantage that it groups related information in to the five main 'competency questions'. It gives a more readable output and is easy to comprehend. On the other hand, the KM3 is able to extract detailed knowledge regarding rules, constraints, conditions, etc., which the 5Ws cannot do. Hence, we propose to use a combination of these two methods to complement each other, so that we may arrive at a more complete knowledge representation.

All steps in the proposed DCMF process have been done manually and we are looking into what tools are needed to automate the process to the largest possible extent. To formalise the methods for automatic tagging and extraction of explicit data from raw natural language or from other forms, and the methodology for extracting implicit knowledge from domain knowledge is also suggested as future areas of study.

Some interesting tasks for future work are among others: formalising mapping rules for mapping from the Knowledge Acquisition phase to the Knowledge Representation and thereafter to the Knowledge Modelling phase; formalising the ontology structure through the use of inbuilt axioms and logic constraints; extending the current ontology structure to include more detailed specifications like "Rules of Engagement", "Standard Operating Procedures", etc.

Finally, by performing the DCMF work, we have gained valuable experience which takes the DCMF project a step closer to its goal. We have found the task to be very large, complex and complicated with many challenges. We believe that a great effort is needed to reach the goal of knowledge reuse and interoperability. We believe that if true interoperability between simulation models built on conceptual models is desired, then a real international cooperation in this field would be very valuable and welcome.



## Acronyms

5 W's	Who What When Where Why
AI	Artificial Intelligence
AOI	Area Of Interest
AOR	Area Of Responsibility
BPMN	Business Process Modelling Notation
C2	Command and Control
C2IEDM	Command Control Information Exchange Data Model
CMMS	Conceptual Models of Mission Space
CMMS-TF	CMMS Technical Framework
CNF	Conjunctive Normal Form
DAML	DARPA Agent Markup Language
DCMF	Defence Conceptual Modelling Framework
DCMF-R	Defence Conceptual Modelling Framework - Repository
DCMF-O	Defence Conceptual Modelling Framework - Ontology
DCMF-P	Defence Conceptual Modelling Framework - Process
DL	Description Logic
DMSO	Defence Modelling and Simulation Office
DNF	Disjunctive Normal Form
FDMS	Functional Descriptions of the Mission Space
FOI	Swedish Defence Research Agency
GIS	Geographic Information System
IEEE	Institute of Electrical and Electronics Engineers
JC3IEDM	Joint Command Control Communication Information Exchange Data Model
KA	Knowledge Acquisition
KE	Knowledge Engineering
KIF	Knowledge Interchange Format
KIRC	Knowledge Integration Resource Centre
KM	Knowledge Modelling
KM3	Knowledge Meta Meta Model
KR	Knowledge Representation
KU	Knowledge Use
LC2IEDM	Land Command Control Information Exchange Data Model
M&S	Modelling and Simulation

MDA	Model Driven Architecture
MILO	Middle Level Ontology
MIP	Multilateral Interoperability Programme
MiSO	Military Specific Ontology
MOF	Meta Object Facility
MSM	Mission Space Models
MSMP	Modelling and Simulation Master Plan
MTK	Tactical Command of Naval Forces
MUST	Joint Military and Security Directorate
NATO	North Atlantic Treaty Organisation
NBI	Networked Based Intelligence
ODM	Ontology Definition Meta-Model
OMG	Object Management Group
OPIL	Operative Command of Swedish Armed Forces
OWL	Web Ontology Language
PSO	Peace Support Operation
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SME	Subject Matter Expert
SPO	Subject Predicate Object
SUMO	Suggested Upper Merged Ontology
SUO	Standard Upper Ontology
SUO WG	SUO Working Group
SWAF	Swedish Armed Forces
TOVE	Toronto Virtual Enterprise Ontology
UML	Unified Modelling Language
UPON	Unified Process for Ontology building
US DoD	US Department of Defence
WMD	Weapon of Mass Destruction
VV&A	Verification, Validation & Accreditation
WWW	World Wide Web
XMI	XML Meta-data Interchange
XML	Extensible Markup Language
XSLT	eXtended StyLesheet Transformation



## References

- [And05] Andersson, B., Garcia Lozano, M. Mojtahed, V. Using a Knowledge Meta Meta Model (KM3) Building Conceptual Models of the Mission Space (CMMS). 05F-SIW-040. Fall SIW 2005
- [Art96] Artala, A., E. Franconi, N. Guarino and L. Pazzi. Part-Whole Relations in Object- Cantered Systems: an Overview. Data and Knowledge Engineering 20(3): 347-383. 1996
- [Bac01] Bacławski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Holmes, W., Letkowski, J., and Aronson M., Extending UML to Support Ontology Engineering for the Semantic Web. Proc. of the Fourth International Conference on UML (UML2001), Toronto, October 2001
- [Ber98] Bergamaschi, S., S. Castano, S. D. C. d. Vimercati and M. Vincini (1998). An Intelligent Approach to Information Integration. Formal Ontology in Information System. N. Guarino. Amsterdam, IOS Press.
- [Bor97] Borst, P., H. Akkermans and J. Top (1997). Engineering Ontologies. International Journal of Human-Computer Studies 46(Special Issue on Using Explicit Ontologies in KBS Development): 365-406.
- [Bor03] Borgida, A. & Brachman, R.J. Conceptual Modelling with Description Logics. Description Logic Handbook. 2003: 349-372
- [BPM05] Business Process Modeling Notation , <http://www.bpmn.org/>. Last visited 2005-11-02
- [Bra95] Brazier, F., Treur, J., Wijngaards, N. The Acquisition of a Shared Task Model, Artificial Intelligence Group, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands, 1995
- [Bra04] Brade, D. A generalised Process for the Verification and Validation of Models and simulation Results. Disseratatio, Fakultät für Informatik, Universität der Bundeswehr München. 2004.
- [Bur05] Burge, J. E. Knowledge Elicitation Tool Classification Artificial Intelligence Research Group, Worcester Polytechnic Institute, web page last visited 2005-10-28  
<http://web.cs.wpi.edu/~jburge/thesis/kematrix.html>
- [Cha93] B. Chandrasekaran, et al., Generic tasks and task structures: History, critique and new directions, Second Generation Expert Systems (pp. 232-272). Berlin: Springer-Verlag, 1993.
- [Chi03] Chira, O. Ontologies. IDIMS (The Intelligent Agent Based Collaborative Design Information Management and Support Tools (IDIMS) project) Report. February 2003. Available online at [http://www-lih.univ-lehavre.fr/~serin/ACS/encadrement/theseNSassi/bibliographie1/Report\\_on\\_Ontologies.pdf](http://www-lih.univ-lehavre.fr/~serin/ACS/encadrement/theseNSassi/bibliographie1/Report_on_Ontologies.pdf)
- [CMM97] DMSO, Conceptual Models of the Mission Space Technical Framework, USD/A&T-DMSO-CMMS-0002, Revision 0.2.1, 1997. <http://www.dmsomil/briefs/entereff/doc/cmmstf.pdf>.
- [Cra99] Cranefield, S., and Purvis, M. UML as an Ontology Modelling Language, Proc. of the Workshop on Intelligent Information Integration, 16th Int. Joint Conference on AI (IJCAI-99), Stockholm, 1999
- [Cra01] Cranefield, S. UML and the Semantic Web, Proc. of the International Semantic Web Working Symposium, Palo Alto, 2001
- [Dem05] DeMasi, L. et al. Implementing Battle Management Language: A Case Study Using the Command and Control Information Exchange Data Model and C4I-M&S Reference Object Model , Paper 05S-SIW-068
- [DMS05] The Defence Modeling and Simulation Office <https://www.dmsomil> Last visited 2005-11-02

- [Doa02] Doan, AnHai. et al. Learning to map between ontologies on the semantic web. Proceedings to the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA, May 2002
- [Ekl04] Eklöf, M. Et al. Information & Knowledge Management for NBI. FOI--1417--SE. 2004.
- [Fer97] Fernandez, M., A. Gomez-Perez and N. Juristo (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering Workshop on Ontological Engineering. Symposium on Ontological Engineering of AAAI, Stanford, California.
- [Fer99] Fernandez-Lopez, M., A. Gomez-Perez, A. Pazos-Sierra and J. Pazos-Sierra (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. IEEE Intelligent Systems and their applications January/February: 37-46.
- [Fer01] Fernandez-Lopez, M. (2001). Overview Of Methodologies for Building Ontologies. Intelligent Systems 16(1): 26-34.
- [Fik99] Fikes, R., Farquhar, A. (1999). Distributed Repositories of Highly Expressive Reusable Ontologies. IEEE Intelligent Systems 14(2): 73-79.
- [Gar05] Garcia Lozano, M. et al. A Process for Developing Conceptual Models of the Mission Space (CMMS) – from the Knowledge Acquisition to Knowledge Use. 05F-SIW-038. Fall SIW 2005
- [Gas05] Gasevic et al. UMLtoOWL converter project home page.  
<http://afrodita.rcub.bg.ac.yu/~gasevic/projects/UMLtoOWL/> last accessed on 25<sup>th</sup> Sept 2005.
- [Gas04] Gašević, D., Djurić, D., Devedžić, V., Damjanović, V., “Converting UML to OWL Ontologies,” In Proceedings of the 13 th International World Wide Web Conference , NY, USA, 2004, pp. 488-489
- [Gom98] Gomez-Perez, A. (1998). Knowledge Sharing and Reuse. The Handbook on Expert Systems. Liebowitz, CRC Press.
- [Gom95] Gomez-Perez, A., N. Juristo and J. Pazos (1995). Evaluation and assessment of knowledge sharing technology. Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing. N. J. Mars. Amsterdam, IOS Press: 289-296.
- [Gom04] Gómez-Pérez, Fernández-López, M. Corcho, O. Ontological Engineering. Springer. 2004. p 163. ISBN 1-85233-551-3
- [Gru91] Gruber, T. R. (1991). The Role of Common Ontology in Achieving Shareable, Reusable Knowledge Bases. Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference, San Mateo, Morgan Kaufmann, 1991.
- [Gru93a] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specification. Knowledge Acquisition 5(2): 199-220.
- [Gru93b] Gruber, T R. Toward principles for the design of ontologies used for knowledge sharing. Presented at the Padua workshop on Formal Ontology, March 1993.
- [Gru94] Gruninger, M. and M. S. Fox (1994). The Role of Competency Questions in Enterprise Engineering. IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway.
- [Gru95] Gruninger, M. and M. S. Fox (1995). Methodology for the Design and Evaluation of Ontologies. IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Quebec, Canada.
- [Gua95] Guarino, N. (1995). Formal Ontology, Conceptual Analysis and Knowledge Representation. International Journal of Human and Computer Studies 43(5/6): 625-640.

- [Gua97] Guarino, N., S. Borgo and C. Masolo (1997). Logical Modelling of Product Knowledge: Towards a Well-Founded Semantics for STEP. European Conference on Product Data Technology (PDT Days 97), Sophia Antipolis, France.
- [Gua98] Guarino, N. (1998). Formal Ontology and Information Systems. Formal Ontology in Information Systems. FOIS.98, 6-8 June 1998., Trento, IOS Press.
- [IDE05] IDEF1X. A Data Modeling Method available at <http://www.idef.com/IDEF1X.html>. Last visited 2005-11-02
- [Joh97] P. Johannesson, M. Boman, J. Bubenko, and B. Wangler, Conceptual Modelling, Prentice Hall, 1997
- [JC305] The documentation on JC3IEDM is available on: [http://www.mip-site.org/publicsite/Baseline\\_3.0/JC3IEDM-Joint\\_C3\\_Information\\_Exchange\\_Data\\_Model/](http://www.mip-site.org/publicsite/Baseline_3.0/JC3IEDM-Joint_C3_Information_Exchange_Data_Model/). Visited 2005-10-25
- [Kat05] Example of using the 5Ws for analysis <http://kathyschrock.net/abceval/5ws.pdf> Last visited 2005-11-02
- [Law00] Law, A. & Kelton, D. Simulation, Modelling and Analysis. Third Edition. McGraw Hill. 2000.
- [Lun04] Lundgren, M., Garcia Lozano, M., Mojtahed, V. CMMS under the Magnifying Glass – An Approach to Deal with Substantive Interoperability. 04F-SIW-010. Fall SIW 2005
- [MSM95] DMSO, “DMSO 1995 Master Plan”, 1995, <http://www.dmsomil/briefs/msdocs/policy/msmp.pdf>, Last visited 2001-10-05
- [Map05] Ehrig, M. & York S. Ontology Mapping - An Integrated Approach. April 21, 2004 [http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2004\\_mapping\\_TR.pdf](http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2004_mapping_TR.pdf)
- [Men98] Mena, E., Kashyap, V., Illarramendi, A., Sheth, A. (1998). Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. Formal Ontology in Information Systems. N. Guarino. Amsterdam, IOS Press.
- [MIP05] Multilateral Interoperability Programme. <http://www.mip-site.org/> Last visited 2005-11-02
- [Moj03] Mojtahed, V., Garcia Lozano, M., Ulriksson, J., Lundgren, M. An analysis of CMMS focusing on Knowledge Acquisition and Knowledge Engineering. FOI-R—1049—SE. 2003
- [Moj04] Mojtahed, V. & Garcia Lozano, M. Kunskapsrepresentation (KR) och Kunskapsmodellering (KM) i CMMS-konceptet. FOI-R—1433—SE. 2004.
- [Nic05] Nicola, A., Missikoff, M., Navigli, R. (2005). A proposal for a Unified Process for Ontology Building:UPON. DEXA conference proceedings. 2005.
- [Nil01] Niles, I. & Pease, A. Towards a standard upper ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS) 2001
- [Noy99] SMART: Automated Support for Ontology Merging and Alignment Natalya Fridman Noy and Mark A. Musen <http://sem.ualgary.ca/KSI/KAW/KAW99/papers/Fridman1/NoyMusen.pdf>. Last visited 2005-11-02
- [Noy01] Noy, N. F. and D. L. McGuinness (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford, CA, 94305, Stanford University.
- [ODM05] Ontology Definition MetaModel proposal, August 2005, Available at <http://www.omg.org/docs/ad/05-08-01.pdf>

- [OMG05] Object Modelling Group. <http://www.omg.org>. Visited 2005-10-31
- [Org05] Swedish Defence Organisation. <http://www.hkv.mil.se/article.php?id=5600> Last visited 2005-05-09
- [OWL05] OWL Web Ontology Language <http://www.w3.org/TR/owl-features/>. Last visited 2005-11-02
- [Per05] Perme, D. Integrating Air and Ground Operations within a Common Battle Management Language. 05S-SIW-154
- [PRO05] Free ware tool to manage multiple ontologies in Protégé. <http://protege.stanford.edu/plugins/prompt/prompt.html>. Last visited 2005-11-02
- [RDF05] Resource Description Framework (RDF), <http://www.w3.org/RDF/>. Last visited 2005-11-02
- [Sch00] G. Schreiber et al., Knowledge Engineering and Management: The CommonKADS Methodology, A Bradford Book; The MIT Press Cambridge; London; England, 2000
- [Spy02] Spyns, P., R. Meersman and M. Jarrar (2002). Data Modelling versus Ontology Engineering, ACM SIGMOD Record. 31.
- [Stu98] Studer, R., V. R. Benjamins and D. Fensel (1998). Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering 25(1-2): 161-197.
- [SUO05] Home page of Standard Upper Ontology Working Group <http://suo.ieee.org/>. Last visited 2005-11-02
- [SUK05] IEEE Standard Upper Ontology (SUO) Study Group Knowledge Interchange Format <http://suo.ieee.org/SUO/KIF/> . Last Visited 2005-11-02
- [SUM05] This site contains information about the SUMO (Suggested Upper Merged Ontology) <http://ontology.teknnowledge.com/> . Last visited 2005-11-02
- [Ter05] Domain Ontology of terrorism. <http://reliant.teknnowledge.com/DAML/Terrorists.owl> Last visited 2005-11-02
- [Tha04] REVVA, VV&A Process Specification (PROSPEC) "User's Manual". v1.3. THALES JP11.20-WE5200-PROSPEC-D5201. 2004
- [Tke05] Table of methods for knowledge elicitation, web page, last visited 2005-10-28 <http://intsys.fin.qub.ac.uk/intsys/0203/abdom/knowledgeelicitation.html>
- [TOV05] Toronto Virtual Enterprise Ontology project . More info on: <http://www.eil.utoronto.ca/enterprise-modelling/tove/> (last accessed on 14<sup>th</sup> Oct 2005)
- [Tri05] Apte, P.R., Shah, H., Mann, D. 5W's and an H of TRIZ Innovation <http://www.triz-journal.com/archives/2001/09/d/> Last visited 2005-11-02
- [UML05] Unified Modeling Language <http://www.uml.org/>. Last accessed 2005-11-02
- [Usc96a] Uschold, M. (1996). Building Ontologies: Towards a Unified Methodology. Expert Systems '96, the 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems, Cambridge, UK, 16-18 December 1996.
- [Usc96b] Uschold, M. and M. Gruninger (1996). Ontologies: Principles, Methods and Applications. The Knowledge Engineering Review 11(2): 93-136.
- [Usc95] Uschold, M. and M. King (1995). Towards a Methodology for Building Ontologies. Workshop on Basic Ontological Issues in Knowledge Sharing" IJCAI-95.

- [Usc98] Uschold, M. (1998). Knowledge level modelling : concepts and terminology. The Knowledge Engineering Review 13(1): 5-29.
- [W&V05] <http://www.fmuhc.mil.se/fmpub/pubdok/armmat.pdf>. Visited: 2005-08-15
- [W3C05] Homepage of World Wide Web Consortium. <http://www.w3.org/2001/sw/> Last visited 2005-11-02
- [WBS04] Ehrig, M. & Sure, Y. Ontology Mapping - An Integrated Approach. April 21, 2004  
[http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2004\\_mapping\\_TR.pdf](http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2004_mapping_TR.pdf) Last visited 2005-11-02
- [Wik05] The wikipedia linguistics site <http://en.wikipedia.org/wiki/Linguistics>. Last accessed 2005-11-02
- [WMD05] Domain Ontology for Weapons of Mass Destruction. Available at:  
<http://reliant.teknowledge.com/DAML/WMD.owl> Last visited 2005-11-02
- [XML05] Extensible Markup Language (XML) <http://www.w3.org/XML/>. Last visited 2005-11-02





An ActionType is used for modelling real world action. An ActionType has start and stop conditions to be able to state when an activity starts and when it is finished. An ActionType can be decomposed into sub-ActionTypes through the ActionTypeComposition. ActionTypes can be sequenced or ordered through the use of attribute in the ActionType class. The attribute EstimatedTime is used to express an approximation of the duration of the activity.

## **2 ActionTypeComposition**

An ActionTypeComposition (not in the figure) is a subclass of ElementComposition . It is used for modelling sequences of ActionTypes, or, in other words, capturing the order of actions.

## **3 AtomicFormula**

An AtomicFormula is an atomic statement about a RoleInActionType. An AtomicFormula can be part of more complex formulas expressed in CNF or DNF. An AtomicFormula is modelled as the super class of three different classes of formulas: a) StateFormula , which concerns states, b) AttributeFormula, which concerns attributes, and c) BeliefFormula, which concerns beliefs.

Example: [Officer Tired] AND [Angle\_Of\_Attack, 40.8].

## **4 AttributeComposition**

An AttributeComposition is the super type of the four different types DistinctValueAttribute, MultiValuedAttribute, RangedAttribute, and MultiRangedAttribute. It has no other function than to group those types.

## **5 AttributeFormula**

An AttributeFormula is a subclass of AtomicFormula. It is a formula in CNF or DNF, and it concerns the attributes of a RoleInAction.

A RoleInAction must not be associated with any ElementAttribute, so the AtomicFormula controlling the RoleInAction may have an empty AttributeFormula slot.

Example: (Battle unit 1 (Angle\_of\_Attack 14))

## **6 AttributeType**

An AttributeType is a group of ElementAttribute names.

## **7 AttributeValueRule**

An AttributeValueRule is a specification rule for an ElementState. In other words, it is a definition of the constraints for a valid state.

A state may be thought of as being (at least) a set of non-optional attributes of a model element. The values of those attributes may be one of four different types: a) DistictValueAttribute, b) MultiValueAttribute, c) RangedAttribute, and d) MultiRangedAttribute. An AttributeValueRule defines which attributes a model element has, and also the types of the attributes' values as just indicated. If a model element has an attribute value which is not conforming to an AttributeValueRule, then the state is not valid.



## 8 BeliefFormula

A BeliefFormula is a statement about a non-objective fact. A BeliefFormula makes it possible to model the beliefs of a RoleInAction.

## 9 CNFFormula

A CNFFormula is formula expressed in Conjunctive Normal Form. A formula in this form is a set of statements, Conjuncts, connected by AND-connectors. The Conjuncts are in turn connected by OR-connectors.

Example: (s1 AND s2) OR (s3 AND s4).

## 10 Criterion

A Criterion is the start or stop conditions for an ActionType. A start criterion is a set of statements that when true, starts an activity. A stop criterion is analogously defined. Every criterion is connected to a state description indicating the new state which an activity is in after the start or stop criterion has been evaluated to true. A criterion may also have effects on the roles associated with execution of the activity. An effect can be either a new multivalued attribute, a single valued attribute, or a belief.

The probability value of a criterion becoming true can be used to abstract away from the aggregated parts of an activity. Thus, it is not necessary to know all details about all activities being part of a complex activity. Important information can be calculated just from knowing the probability of activities's criteria becoming true.

## 11 DataType

A DataType determines what data type a Value has.

Example: String, Integer, Boolean

## 12 DisjunctiveFormula

A Disjunctive formula is an complex atomic formula where the parts are separated by logical 'OR'. This enables the construction of complex formulas where, when one of the parts are true, the entire formula is true.

## 13 ElementAttribute

An ElementAttribute is a composition of a named domain and the model elements for which the attribute applies.

The attribute 'theAttributeCompositionType' captures the composition of values given the value domain.

## 14 ElementComposition

An ElementComposition is a binary relation between Model Elements where one is designated as being the composed element (called compositionComponent) and one being the composing (called consistsOf). The ElementComposition creates a group of composed elements where the order of the elements is irrelevant.

## 15 ElementInheritance

An ElementInheritance is a subset of the ElementComposition relation where none of the components may be an ActionType. Whatever attributes the consistsOf-element has, are inherited by the compositionComponent-element.

**16 ElementState**

An ElementState (or simply State) is a single valued attribute that takes its value from on the set of values defined by multi valued attributes. A state is associated with a model element.

A valid state is a state that complies with the definitions given for that state in a StateDefinition. States defined by StateDefinitions are considered as objective facts. A state with no StateDefinition is considered as a subjective opinion and not as a fact.

Example: A soldier is Combat ready if [Gun Loaded] AND [Morale 20-200]

**17 EntityType**

An EntityType is a basic generic concept or a thing, abstract or concrete.

Example: Person, Vehicle, Threat

**18 ModelElement**

A ModelElement is the foundational construct in the KM3. All concepts are modelled using ModelElement. A ModelElement can be associated with multi-valued attributes (Attributes) and single valued attributes (ElementStates) and be associated with other ModelElements through ElementComposition.

**19 DistinctValueAttribute**

A DistinctValueAttribute is an attribute that can take a single point value at a time.

Example: 20, A, Battle\_ready

**20 MultiRangedAttribute**

A MultiRangedAttribute is an attribute that can take several intervals of values at a time.

Example: 20-30 AND 50-60, Battle\_ready AND 18-30

**21 MultiValuedAttribute**

A MultiValuedAttribute is an attribute that can take several single point values at a time.

Example: 20 AND 30, Battle\_ready AND 20

**22 RangedAttribute**

A RangedAttribute is an attribute that can take a single interval value at a time.

Example: 20-30

**23 RoleInAction**

A RoleInAction is a defined constituent in an ActionType. All ModelElements that participates in an action do so by taking the roles defined for the ActionType.

A Soldier defined as a RoleInAction may be a completely different thing than that of a Soldier defined as a RoleInOrganisationType.

Example: Battle unit 1, Target

**24 RoleInActionType**

A RoleInActionType is a group of RoleInAction names.

**25 RoleInOrganisationType**

A RoleInOrganisationType is a definition of the capabilities, obligations and prohibitions that can be associated with an EntityType.

Example: Fighter\_pilot, General, Tank

**26 StateDefinition**

A StateDefinition is a rule to define a valid state. A StateDefinition is built by one or more AttributeValueRules. If several AttributeValueRules are used, then they are conjunctively (AND) joined to form the StateDefinition.

If several StateDefinitions are used to define one valid ElementState, then they are disjunctively (OR) joined to do so. One StateDefinition may be used to define several valid states.

Example: [Angle\_of\_Attack 40 AND Fuel 36] OR [Battle\_readiness 13]

**27 StateFormula**

A StateFormula is a pair consisting of an ElementState and a RoleInAction. A StateFormula effectively creates a statement about state of a RoleInAction. Such statements can be used when defining start and stop criteria for ActionTypes.

Example: [The\_wounded Tired], [Battle\_unit\_1 Ready]

**28 StateType**

A StateType is a group of ElementState names.

**29 Unit**

A Unit is a standard by which measurable ElementAttributeTypes are associated to supply a measure for the attribute.

Example: Pound, second, meter

**30 Value**

A Value is an element of a ValueDomain, given as a string of text.

Example: Battle\_ready, 30

**31 ValueDomain**

A ValueDomain defines the valid values of an ElementAttribute. A ValueDomain may be one or several point values, or one or several interval values. A ValueDomain is defined using pairs of values (start and stop). If a stop value is missing then a point value is defined.

**32 ValuePair**

A ValuePair is a pair of Values (sic) where one is called Start and the other Stop. The ValuePair is used when defining ValueDomains.



## **Appendix B – 05F-SIW-038**

### **A Process for Developing Conceptual Models of the Mission Space (CMMS) – From Knowledge Acquisition to Knowledge Use**

*Marianela García Lozano*

*Vahid Mojtahed*

FOI, Swedish Defence Research Agency  
{garcia, vahid.mojtahed}@foi.se

This paper has been presented at 2005 Fall Simulation Interoperability Workshop at  
Orlando, Florida - USA

# A Process for Developing Conceptual Models of the Mission Space (CMMS) – From Knowledge Acquisition to Knowledge Use

Marianela García Lozano  
Vahid Mojtahed  
FOI, Swedish Defence Research Agency  
164 90 Stockholm  
Sweden  
+46 8 5550{3133, 3705}  
{garcia, vahid.mojtahed}@foi.se

## Keywords

CMMS, Knowledge Acquisition (KA), Knowledge Engineering (KE), Knowledge Representation (KR), Knowledge Modelling (KM), Knowledge Use (KU), Ontology, Standardization, Substantive Interoperability, composability, reusability, VV&A

**ABSTRACT:** *Conceptual Models of the Mission Space (CMMS) was originally introduced by the US DoD in 1995 in the Modelling and Simulation Master Plan (MSMP). The vision was to achieve interoperability, composability and reusability of knowledge for modelling and simulation in the military domain. Unfortunately the concept never got completed and all activities seemed to end around the end of the millennium. FOI, the Swedish Defence Research Agency, found the concept interesting and began its own work with the CMMS concept in 2001.*

*Since then FOI has strived to develop the concept to implement and fulfil the promises of the vision. The FOI CMMS project discovered early that the CMMS process was vague and unfinished, so a large part of the work done since the start, has been to develop a basis for a common methodological framework. In this paper we will present our approach and discuss our CMMS process from the start of the knowledge acquisition phase to the end phase where the formalized and modeled knowledge can be extracted and reused. We will also discuss the necessary components, their relationship and techniques that are needed to implement it. The importance of a VV&A presence throughout the CMMS process will also briefly be discussed.*

## 1 Introduction to CMMS

The increasing use of modelling and simulation puts demands on how knowledge is used and managed. Major challenges are how to acquire, maintain and validate knowledge and how to do it as effortlessly as possible.

With the purpose of structuring the modelling and simulation work the US Department of Defence presented in 1995 a vision called the Modelling and Simulation Master Plan (MSMP). The aim was to build an architecture and infrastructure that could increase the reuse and

interoperability of simulation models. CMMS – Conceptual Models of the Mission Space is one the three basic components of the technical framework presented in the MSMP.

CMMS is often described as the first abstraction of the real world. It is a framework for the development of models and it captures the characteristics of objects in a domain given by a scenario, such as their features, interactions, and behaviour. The models, or rather Mission Space Models (MSM) strive to be generic and applicable to as many scenarios as possible without any loss of critical knowledge. MSM:s are implementation

independent conceptual models of objects and activities in the military domain.

CMMS tries to address three problems that commonly arise in model development:

- Authoritative information is not readily available.
- Knowledge acquisition is incomplete and sometimes ambiguous.
- The knowledge that has been acquired (often at considerable expense) is not stored and maintained for future use.

The potential uses and users of CMMS are many, for example, developers of simulation models use MSMs to get a tool to understand the involved entities, tasks and actions in military operations. By having a validated view of a domain provided by an authorised source the likelihood of producing consistent, interoperable and reusable models and simulations increase.

### 1.1 The FOI CMMS Story

Since its introduction in 1995 the CMMS has been replaced by other initiatives (such as FDMS and KIRC, see [1]), but presumably connections can be made to these other concepts, since we believe that they share a common goal as well. For the sake of simplicity we only refer to CMMS in this paper.

The Swedish Defence Research Agency, FOI, carried out a prestudy of CMMS in 2001 to learn more about the CMMS-process, see to which extent and under which circumstances it could be of interest. The study yielded both increased interest and confusion as the available specifications [2, 3] were perceived as vague, partly unfinished and ambiguous. Some of our findings were presented in [4].

Since we in our studies found the CMMS-concept to be interesting and with high potential, our work with it was continued. In 2002 an extended study of available material and methods was conducted and a plan for future work was laid out. 2003 the work focused on the early phases of the CMMS-process such as Knowledge Acquisition. The year after, in 2004, we continued our work and focused on the following phases. During this work we found that a lot of the CMMS-process, -components and -tools to be uncharted territory and therefore we had to basically start from the beginning. We soon discovered that a fundamental piece missing was a

structure by which to model the MSMs. We made a proposal for such a structure and called it the Knowledge Meta Meta Model (KM3). The KM3 represents one possibility to capture system structures and behavior in an object-oriented and rule based way. This model is described in the paper [5]. All of our work done so far has been summarized in our reports [6, 7, 8]

This year our work has mainly focused on making MSM prototypes and following through with the process to see if it is feasible and to gather experiences. With this work we have also been able to further identify necessary tools, methods and techniques. In our work we have found Conceptual modelling, the Semantic Web, Model Driven Architecture, VV&A, computational linguistics, Ontologies, etc. very helpful.

This paper focuses on the CMMS-process and how information is analysed, represented and modelled to suit our needs. The different analysis methods, necessary components, people involved, and process steps are described. The importance of a VV&A- and Inspection-presence throughout the CMMS process is also discussed.

### 1.2 The original CMMS-process

In the CMMS Technical Framework document, see [9], the CMMS concept, process and tools were outlined. The document described the framework, a number of different technical standards, administrative procedures, and layout of the infrastructure needed to build conceptual models. Some parts were described in more detail but others were vague and ambiguous.

The description of the CMMS-process was only laid out on a high level where only the main parts were described. The process was described as a two phase process, where the two phases were Knowledge Acquisition (KA) and Knowledge Engineering (KE). In the KA-phase there were two main steps where the development of a focused context is the first and the gathering of information the other. The KE-phase consisted similarly of two steps where the formalizing of data and construction of MSMs and CMMS was the other.

The description of the CMMS-process was very useful as a starting point, but it did not tell us further how to proceed to develop our own conceptual models. Some tools were also described in the document but it proved very difficult to get access to them. Therefore we saw the

necessity of developing and studying a new process, methods, technologies and tools.

## 2 Overview of the FOI CMMS-Process

In the FOI CMMS process we have split the Knowledge Engineering phase into two. We have chosen to call them Knowledge Representation and Knowledge Modelling. This was done because we found that the engineering phase was very complex and needed breaking down. In order to do so, we analysed the steps and saw that there were two main parts, where the first had to do with the formalising and representation of the acquired knowledge and the second had to do with the modelling of the acquired knowledge. Last but not least is the Knowledge Use phase. Therefore the FOI CMMS process consists in large of four main parts, see Figure 1. By the end of every phase there are one or more products that are used as input in the next phase. The process is not completely sequential, between the different steps in and between each phase there is constantly an iterative process going on. There is especially an exchange when the experiences from the use-part go back into the CMMS-system.



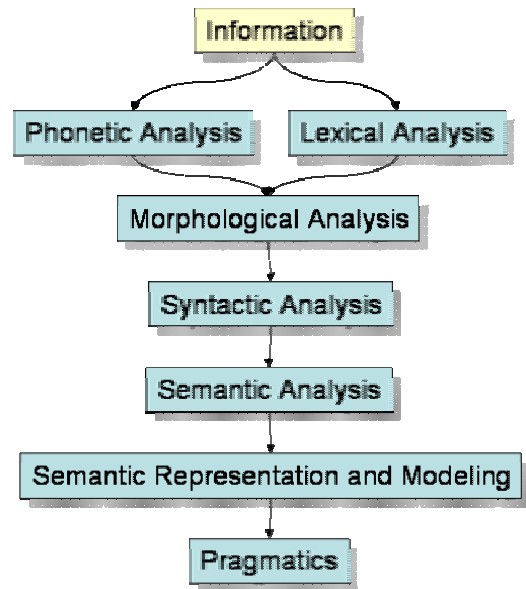
**Figure 1:** The four main parts are KA, Knowledge Acquisition, KR, Knowledge Representation, KM, Knowledge Modelling and finally, Knowledge Use, KU.

In order for the information to be useful for our purposes it must undergo a process where it is analysed, formatted, modelled and then used. The different steps of the CMMS-process aim at doing that by logically modelling natural language from a computational perspective, see Figure 2. We will not dwell on the finer points of computational linguistics but present a brief description of the different kinds of areas and types of analysis, representation and modelling that the information undergoes in the CMMS process.

**Phonetic analysis** – the phonetic analysis is done when information is gathered by interviews, discussions, talks or anything that is spoken. This is done quite unconsciously and allows us to interpret what is being said. This analysis is based on previous experience and may therefore lead us wrong sometimes. This is what helps us to determine if what was said was *rudder* or *robber*, *hat* or *hot* and so on.

**Lexical analysis** – the lexical analysis is like the phonetic analysis but for written information.

**Morphological analysis** – this is what allows us to distil useful and plausible information. It helps us identify the internal structures of words and thus allowing us to further understand what is being said or written. This is also done quite unconsciously if one knows the language well.



**Figure 2:** The information is processed in several steps before it may be useful from a CMMS-perspective.

**Syntactic analysis** – is used to decide how words are combined and arranged to form grammatically correct sentences. This helps us see how something is expressed and decide whether words are nouns, verbs, adjectives and so on which makes it easier to look them up and see what they are implying.

**Semantic analysis** – is used to decide the meaning of words and how they are combined to form meaningful sentences. It has more to do with the use of words than the nature of the entity being referenced by the word (this is the task of an ontology).

**Semantic Representation and Modelling** – this helps us with the representation and modelling of the *meaning* of the world, and indirectly the meaning of a statement or sentence along with its surrounding context.

**Pragmatics** – this is how language, sentences or knowledge is used, in different contexts. It helps us decide what actions to take or how to respond to



information in a certain situation. For our purpose it helps us decide what knowledge is suitable for a certain need.

Read more about linguistics and the individual areas in [10].

The borders between the different phases are a bit fuzzy and it may sometimes be difficult to tell exactly where one phase starts and the other ends. The phases are used to present an overview and abstraction of the process steps discussed in Section 3, see Figure 3.

## 2.1 Knowledge Acquisition, KA

The purpose of the Knowledge acquisition phase is as the name suggests - to acquire knowledge and information. Preferably, the sources used have been authorised by some organisation beforehand.

There are three main steps in this phase:

- The focused context, i.e. the definition of the purpose, the limitations and the need of the knowledge.
- The identification of suitable and authorised knowledge sources.
- The acquiring and documentation of information.

When acquiring information it is important to follow some well documented method that suits the purpose. This is especially important when the information is gathered by interviews, where great care must be taken to avoid unintentional influence by the interviewer.

Examples of methodologies for knowledge acquisition that we have studied are: CommonKADS, Generic Tasks and DESIREE. Ideas from these methodologies have influenced our design of the KA-phase. For more information on the different methodologies and our study of them, see [7, 11, 12, 13].

In the KA-phase there are two important processes that occur in the background when gathering knowledge. There is first a phonetic or lexical analysis depending if the information is acquired by interviews or by written form. Words or expressions that are in a different language or from another domain are translated or mapped to the current language or domain. The second thing that happens is the morphological analysis of the received information. That is, the information that is not relevant to our context is removed, documented in another context or

simply not documented at all. Syntactic and semantic analysis is also done by the Information gatherer.

The persons involved in this phase need to have a deeper understanding of the problem domain and preferably experience in certain areas, for example:

- Authorisation Agency – Identification of suitable knowledge sources, authorising them and describing them
- Information gatherer – experience in gathering and compiling information
- SME - domain knowledge
- Interviewer - knowledge acquirement methods

## 2.2 Knowledge Representation, KR

The aim of the knowledge representation phase is to make information that until now only has been human readable also machine readable. In order to make this possible, the information that has been documented in natural language must be formatted and represented in a machine readable way. The formatting should be done in such a way that no (or little) information is lost in the process and preferably so that the formatted knowledge can be traced back to the source.

Formatting a text can be done with several methodologies; the important issue is *how* the formatted text is to be further processed. At FOI we have looked at two methods; SPO and 5Ws.

SPO stands for Subject, Predicate and Object. This is a generic method that can be applied to almost any domain. In the semantic web framework, SPO is used to structure information. In essence, the written text is analysed and formalised by following the formula of SPO. Read more in [8, 14].

The other method that could be applied is the 5Ws-format, WHO is doing WHAT, WHERE, WHEN and WHY. 5Ws is also used in writing in journalism and for telling a story in literature. Read more about it in [15, 16]

After, or at the same time that the formatting is done, the information should also be mapped to a suitable ontology. The ontology gives the context of the domain, the terms and their relationships and interactions.

We have studied a number of existing ontologies and have found these two to be suitable for our purposes; IEEE's SUMO and NATO MIP's JC3IEDM. Read more

in [17, 18]. We have also seen the necessity of more specialised domain ontologies. Some of those used are predefined and some we have constructed ourselves.

In this phase the information is put through a mainly syntactic and semantic analysis process.

In this phase there are also many different roles that require expertise and experience.

- Formatting - experienced in the appropriate formatting analysis method and technique.
- VV&A-agent – experienced in the VV&A-process and methodology
- Ontology expert – knowledge in the used ontology, experience in mapping and interpretation of information to the ontology as well as knowledge in how to further develop and extend ontologies.

### 2.3 Knowledge Modelling, KM

After the Knowledge Acquisition and Representation phases the resulting products may be usable for some applications. For example, if specific knowledge instances are of interest for a certain application then no further generalization is needed. But, if the aim is to make general and reusable knowledge models then further knowledge modelling is needed.

In the CMMS-project we have during the last two years been designing a model with which we could supply constructs and rules to structure the knowledge. As previously mentioned in Section 1.1 we have called it the Knowledge Meta Meta Model (KM3), see [5].

The biggest difficulty in the knowledge modelling phase is to structure a machine readable formatted text, in for example the 5Ws format, to the KM3 and obtain one or more MSMs. This is in no way trivial and work is currently in progress to try and solve this task. The result of this procedure may be that more information is needed and a return to the KA-phase may be needed.

When the KM3 modelling is finished, the last task is to try and merge these new MSMs to the previously modelled MSMs from the same domain already present in the CMMS Data Base (CDB). The CDB is the collection of acquired MSMs.

To make MSMs as reusable as possible we have thought in the lines of *divide and conquer*. By breaking down the

knowledge in the MSMs to smaller components these knowledge components could then be reused in different configurations producing new MSMs. There are a number of advantages to this course of action: a) it is more flexible and easier to reuse components b) they may be reused for other purposes than they originally were created for and c) components on different levels of abstraction could be combined. The challenge lie in creating well defined interfaces, descriptions and ontologies for the components.

In this phase the information goes through a semantic analysis, representation and modelling process.

### 2.4 Knowledge Use, KU

The final phase of the CMMS- process is the use of the MSMs, knowledge instances and components. In this phase the reuse of knowledge is done either as a part of the KA-phase when knowledge is acquired or as a part of the KM-phase when the KM3-modelled knowledge is merged with previous knowledge.

In this phase the connection to the end-user is the strongest and therefore it is of great interest that the knowledge can be visualized in different ways depending on the domain and the user's purpose.

As previously mentioned, the CMMS-concept was originally suggested as a framework to facilitate modelling and simulation. For that reason there has long been the question of how the connection between an MSM and an implemented simulation model is. The MDA concept, Model Driven Architecture, has been found to a possible candidate for this connection. The concept is based upon the idea that from one and the same conceptual model, code can, through several transformation steps, be obtained for several platforms and languages. Read more about MDA in [19]. MDA is a complex concept that probably complements the CMMS concept very well. This matter needs further study, tests and evaluation.

Other fields of application for the MSMs could be as references for VV&A-efforts, context sensitive reference manuals or even for the identification of services within the Network Based Defence.

This phase deals mostly with the Pragmatics of the modelled knowledge.

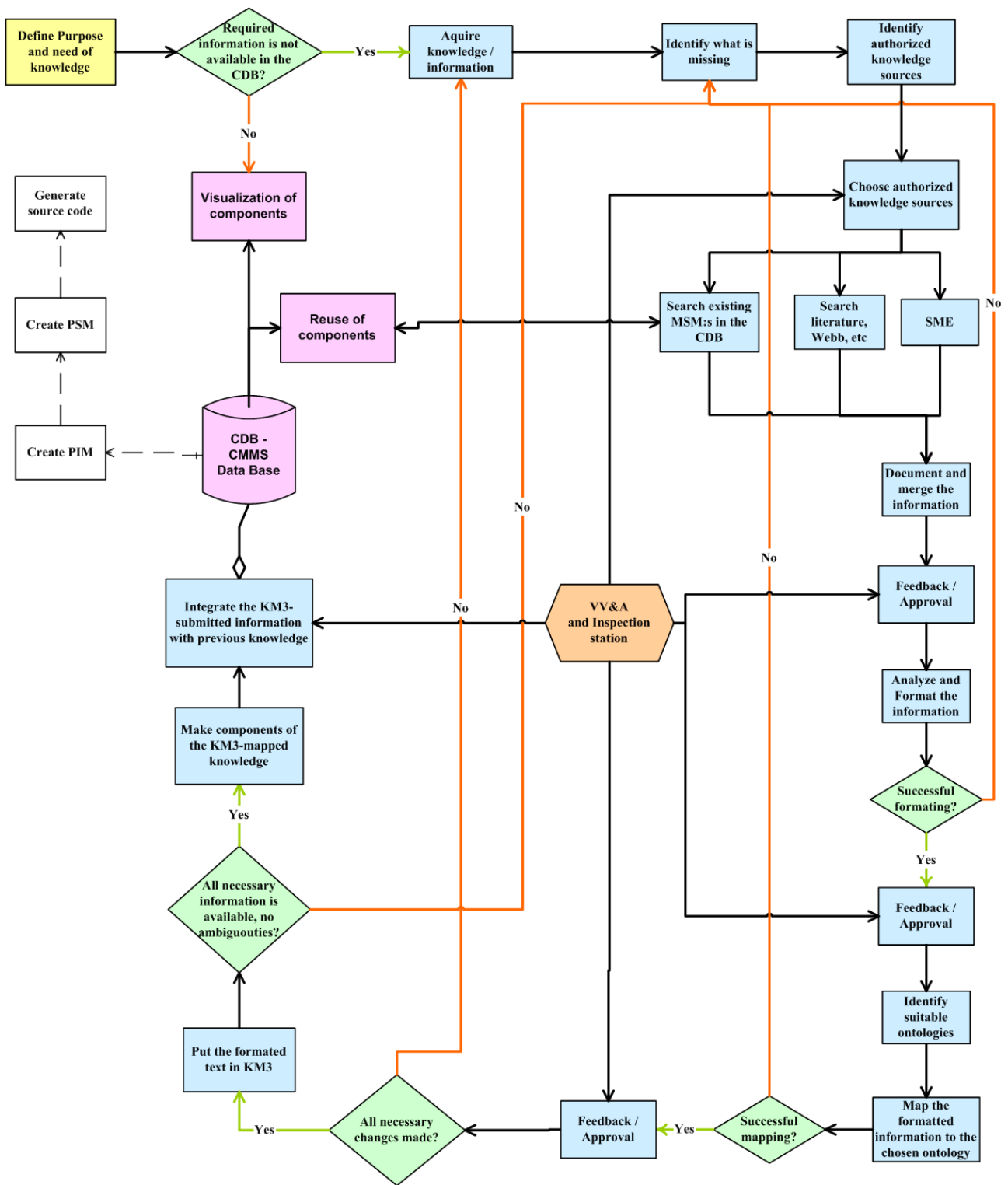


Figure 3: The steps of the FOI CMMS-Process

### 3 The steps of the FOI CMMS-Process

Each phase in the CMMS-process is decomposed into a number of steps and this section describes them. The steps are drawn as boxes in the flowchart; see Figure 3.

#### 3.1 Description of the Notation

This is a brief description of how to interpret the objects, colours and arrows in the process figure.

- The Yellow box is the start of the process.
- Green diamonds are control and decision stations to check whether to move forward in the process or return and iterate some steps.
- The arrows are green if the answer is yes (proceed) and orange if no (return to a previous step).
- Blue boxes are process steps where work is done.
- Pink boxes and database, they are not really a part of the CMMS-process but a part of the CMMS-concept
- The dark yellow hexagon is an organizational unit.
- The white boxes and broken lines denote parts that are not part of the CMMS process but that are interesting from a user perspective.

#### 3.2 The Process steps

The CMMS-process starts when the need of knowledge for some purpose arises. Often it may not be quite clear exactly what is needed, what the delimitations are, or what the exact domain is. An identification of this must then first be made before any information is acquired i.e. a focused context must be developed. In Figure 3 the process starts at the top left corner in the yellow box.

In order to obtain authorized information, suitable information sources must first be found. These sources can be anything from interviews with subject matter experts, SME:s, search in available literature on the web or elsewhere, to search in the CMMS Data Base (CDB) where previously acquired information is stored. The CDB should be the first logical point of search for information. If nothing is found there or only part of what

was required was found, it is necessary to move on in the process and try to acquire information through the other available channels. A clear notion of what is missing or required is necessary; one could say that the most important step in acquiring information is to identify what is missing.

When it is known what is missing, the proper authorized knowledge sources may be identified and chosen. The suggestion and choice of appropriate knowledge sources should be accredited by a VV&A-agent or some other accredited authority.

When information has been gathered by, for example, interviewing SMEs, all of the obtained information should as far as possible be documented and then presented to the SME for approval. If there have been several interviews on different occasions the information from each one could if suitable be merged. The documentation of information will help ensure traceability throughout the process and help us with the feedback and approval from authorized sources. As previously mentioned in Section 2 all information in the CMMS-process undergoes an analysis consisting of several steps, for example the documented information from an interview has undergone both a phonetic and morphological analysis by the interviewer. All such analysis phases are prone to introduce some errors in the information and these errors should preferably be caught as soon as possible.

When the documentation has been done and approval given, the information should be analyzed and formatted to a common format thus expressing the information with a common syntax and semantics. This format could be of different kinds such as SPO or 5Ws, as previously mentioned in Section 2.2, depending on what the final purpose is. One further advantage is that a common format makes the information easier to make machine readable and thus easier to automate certain steps.

Since any analysis or formatting implies a certain loss or modification of knowledge, there should also be a control mechanism here to minimize the loss or distortion. The feedback and approval should be given by the interviewed SME or some other accredited person. If the analysis or formatting has not been done correctly and cannot be approved an iteration of previous steps in the process is required. Each iteration through the process is costly in both time and money and the number of iterations should therefore be minimized.

For the analysis and formatting to work, they should always be done with a chosen domain in mind. Ontologies and dictionaries are invaluable tools that provide terms, structures and relations for a specific domain and thus make it easier to interpret the information. For each domain there may be several ontologies on different abstraction levels or with different focuses but the ones best suited for the purpose should be chosen. We will not discuss how to build ontologies in this paper but there are some well known methods, see [20].

The analysis, formatting and mapping of the information to the chosen ontologies is usually done in parallel. The feedback, control and approval are done by different persons and instances. As previously mentioned the SME gives feedback to the interpretation and analysis of the information, the ontology expert approves the ontology mapping and the VV&A-agent approves that the analysis has been done from the point of view of the initial purpose.

The next step is to build the MSMs. The Knowledge Meta Model provides rules and constructs for the creation of the MSMs. The KM3 can formally be seen as a model specification, consisting of object oriented concepts aimed at primarily capturing dependencies in and between activities. In our work with the CMMS process we have realised the limitations of focusing on military processes as a way of reusing knowledge. We have therefore chosen a more component oriented approach where we pay more attention to the actions that a process is comprised of. This is further discussed in [4].

The formatted knowledge is structured according to the KM3. Unfortunately, to create reusable components, the formatted and ontology mapped information is not enough. Usually not all necessary information is available at this stage. With the KM3-modelling new requirements arise and more acquisition of knowledge may be needed and if so new iterations would be required.

Throughout the process all products that have been the result of different steps are placed in the CMMS Data Base. This gives a higher degree of traceability and possibility to reuse and make components interoperable.

If the purpose is to make reusable knowledge components the information may sometimes need to be de-identified before it can be integrated with previously submitted KM3-modeled information. This means that information that is specific to a certain person or situation is generalised and brought to a higher abstraction level or

specified further. For example, a recognizance mission is done differently according to the object of interest, the environment, the time of year and the available equipment and situation. The pragmatics is an important factor in this step. Work is still in progress in this area, but naturally some kind of VV&A is also needed here.

Visualization of knowledge is a difficult and challenging area. In our work we have not focused on this part yet, but it is vital step needed to convey knowledge.

### 3.3 VV&A

The observant reader has noticed that traceability and VV&A have reoccurred as a common feature throughout the entire CMMS-process. In several steps there is some kind of feedback to earlier steps if there has been any missing, misinterpreted or ambiguous information. It can also be that the analysis, formatting or modelling hasn't been done correctly and therefore needs more work before approving. Sometimes it has been an SME that has had to approve and sometimes a VV&A-agent or some other authorised person.

As described in the CMMS process description there are many different roles and functions that are involved in different steps. Further analysis of their authority and necessary competences should be done. For VV&A-agents the CMMS process and the final product (MSM:s) can be a great aid in the VV&A-process. The question of how cooperation can be done and how the connections between the two fields of VV&A and CMMS look like, should also be analysed and clarified.

Another important connection is to remember what initiates a run through the CMMS-process, the need of information for some specific purpose. Regardless of whether the sought information can be retrieved from previously acquired knowledge in the CMMS Data Base or not, new models are created by performing the steps of the process. VV&A has a natural role here in securing that the correct information is acquired, that the correct model is produced, that it is valid for the purpose and used in the correct way.

## 4 Ongoing and Future Work

Previous years we focused mainly on the Knowledge Acquisition phase and performed some experiments in

acquiring information in a systematic way. We studied some methodologies and got some ideas from them. We then moved on to tools and language related issues such as formatting methods and ontologies.

This year we have focused on the later phases of the process by trying to make prototype MSMs. We have applied the CMMS-process on a hypothetical scenario thus testing the developed tools and theories. We have discovered a number of issues, for example the methodology as it is today is heavily dependant on the formalisation of the scenario. If two method experts analyse and formalise a common scenario description they will likely end up with two different formalisations. Future work involves designing stricter guidelines for the analysis and formalisation of information.

In our approach we chose a hierarchy of ontologies with SUMO as the top level ontology and JC3IEDM as the middle level ontology. We found that mapping the formalised information into this ontology structure is difficult. It is difficult because of the discrepancy in information content between the ontologies and the formalised information. This could be solved by further knowledge acquisition or a better alignment between the ontologies and the formalised information.

The KM3 has been limited to using it as an analysis tool and its full potential as a foundation for the reusable knowledge library has not been fully explored yet.

All steps in the proposed CMMS-process have been done manually and we are looking into where and what tools are needed to automate the process to the largest possible extent.

All of our ongoing work will further be described in a report by the end of this year.

## 5 Discussion and Conclusions

With our work in the CMMS field we have found the CMMS concept to be a good idea worth pursuing. We have also found the task to be very large and if true knowledge reuse is to be achieved more have to join in on the work with it. One very simple conclusion one already can draw is that much more effort is needed if we seriously wish to reach our highly placed goals. Nevertheless we need to point out that if true interoperability between simulation models built on conceptual models is desired then a real international cooperation in this field is necessary.

What we have tried to do is to analyse the CMMS process regardless of domain or available tools. We have identified several steps that need to be taken in order to reach the goal. For these steps we have also studied some methodologies and techniques that can be used, thus laying the foundation for the process.

Presented in this paper is the main design for the FOI CMMS-process. With the aid of this process many necessary tools and methods have been identified. Some of the involved organisational roles, structures and functions have also been identified with this work, for example VV&A. This means that we at least need a knowledge transfer between CMMS and VV&A. We have found that a large part of the methodology presented in this paper may also be used in other areas requiring Knowledge Management as well.

Most of our reports are in Swedish, but a detailed report of the proposed CMMS-project in English is planned for the end of this year. More details of the process, components and our current work will be described there.

## References

- [1] Functional Description of the Mission Space, <https://www.dmsomil/public/transition/fdms/>
- [2] DMSO, "DMSO 1995 Master Plan", 1995, <https://www.dmsomil/public/library/policy/guidance/500059p.pdf>
- [3] "Conceptual Models of the Mission Space – Technical Framework", DMSO, revision 0.2.1, 1997 <http://www.dmsomil/briefs/entereff/doc/cmmstf.pdf>
- [4] M. Lundgren, M. García Lozano, V. Mojtahed, "CMMS under the Magnifying Glass – An Approach to Deal with Substantive Interoperability", Conference proceedings FSIW'04, 2004, 04F-SIW-010, <http://www.sisostds.org>
- [5] B. Andersson, M. García Lozano, V. Mojtahed, "The Use of a Knowledge Meta Meta Model (KM3) when Building Conceptual Models of the Mission Space", Conference proceedings FSIW'05, 2005, 05F-SIW-040, <http://www.sisostds.org>

- [6] V. Mojtahed, M. García Lozano, J. Ulriksson, G. Holm, (In Swedish) "*CMMS – Konceptuella Modeller av Militära Operationer*", FOI Stockholm, 2001, FOI-R—0629—SE
- [7] V. Mojtahed, M. García Lozano, J. Ulriksson, M. Lundgren, (In Swedish) "*Analys av CMMS-konceptet med fokus på Kunskapsanskaffning (KA) och Kunskapsbearbetning (KE)*", FOI Stockholm, 2003, FOI-R—1049—SE
- [8] V. Mojtahed, M. García Lozano, (In Swedish) "*Kunskapsrepresentation (KR) och Kunskapsmodellering (KM) i CMMS-konceptet*", FOI Stockholm, 2004, FOI-R—1433—SE
- [9] DMSO, "Conceptual Models of the Mission Space Technical Framework", USD/A&T-DMSO-CMMS-0002, Revision 0.2.1
- [10] The wikipedia linguistics site <http://en.wikipedia.org/wiki/Linguistics>
- [11] F. Brazier, J. Treur and N. Wijngaards, "*The Acquisition of a Shared Task Model*", Artificial Intelligence Group, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands, 1995
- [12] B. Chandrasekaran, et al., "Generic tasks and task structures: History, critique and new directions", *Second Generation Expert Systems* (pp. 232-272). Berlin: Springer-Verlag, 1993.
- [13] G. Schreiber et al., "*Knowledge Engineering and Management: The CommonKADS Methodology*", A Bradford Book; The MIT Press Cambridge; London; England, 2000
- [14] Semantic Web site <http://www.w3.org/2001/sw/>
- [15] A. Tolk, et al. "*Coalition Battle Management Language*". Conference proceedings FSIW'04. Paper 04F-SIW-103, Fall Simulation Interoperability Workshop, <http://www.sisostds.org>
- [16] Interview with James Stovall <http://www.secondaryenglish.com/stovall.html>
- [17] Suggested Upper Merged Ontology <http://www.ontologyportal.org/>
- [18] MIP site <http://www.mip-site.org/>
- [19] OMG Model Driven Architecture <http://www.omg.org/mda/>
- [20] M. Fernández, "*Overview of Methodologies for Building Ontologies*". Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends. (IJCAI99). August. 1999. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/4-fernandez.pdf>

## Authors

**MARIANELA GARCIA LOZANO** is a Research Engineer at the Department of Systems Modelling at the Swedish Defence Research Agency (FOI) and she has been a part of the CMMS project since its start in 2001. She holds a Master of Science in Computer Science and Engineering and she is currently working on her Ph.D. in the area of Embedded Simulations. Her research interests are in the areas of Conceptual Modelling and Distributed Computing.

**VAHID MOJTAHED** is a senior research officer working at the Department of Systems Modelling at the Swedish Defence Research Agency (FOI). He has been working on modelling and simulation for the past ten years and has led the CMMS project at FOI since 2001. He holds a Master of Science in Computer Science and Engineering from Chalmers University of Technology, Gothenburg, Sweden. His research interests are in the area of Conceptual Modelling, Simulation Framework and Information Operations and Warfare.





## **Appendix C – 05F-SIW-040**

### **The Use of a Knowledge Meta Meta Model (KM3) when Building Conceptual Models of the Mission Space**

Birger Andersson  
Marianela García Lozano  
Vahid Mojtahed  
FOI, Swedish Defence Research Agency  
{birger.andersson, garcia, vahid.mojtahed}@foi.se

This paper has been presented at 2005 Fall Simulation Interoperability Workshop at  
Orlando, Florida - USA

# The Use of a Knowledge Meta Meta Model (KM3) when Building Conceptual Models of the Mission Space

Birger Andersson  
Marianela García Lozano  
Vahid Mojtahed

FOI, Swedish Defence Research Agency  
164 90 Stockholm  
Sweden  
+46 8 5550{3133, 3705}  
{birger.andersson, garcia, vahid.mojtahed}@foi.se

## Keywords

CMMS, Conceptual Modeling, Knowledge Meta Meta Model, KM3

**ABSTRACT** *Conceptual Models of the Mission Space (CMMS) was put forward as a component of a vision for a future technical framework for modeling and simulation by the US Department of Defense in 1995. In 2001, a project was initiated at the Swedish Defense Research Agency (FOI), with the purpose of studying and developing a CMMS framework and assessing the feasibility of the concept.*

*The FOI project found the original CMMS concept complex and partly under-specified. For instance, a structure for knowledge management that could deal with concepts of different levels of abstraction, and both static and dynamic data was missing. Therefore an activity centered Knowledge Meta Meta Model (KM3) was suggested, designed and developed by FOI specifically to enable construction of conceptual models according to the CMMS vision.*

*Presented in this paper is the KM3, its features and the rational behind its design. Its importance and relation to other parts of the CMMS concept is also discussed. The theoretical results of the work so far with the KM3, indicates that it is a promising approach but that it needs further development.*

## 1 Background

Conceptual Models of the Mission Space (CMMS) was put forward as a component of a vision for a future technical framework for Modeling and Simulation (M&S) established by the US Department of Defense (US DoD) in their MSMP - M&S Master Plan in 1995. The main purpose of the CMMS concept was to facilitate and support development, reuse and interoperability between simulation models. DMSO – the US Defense Modeling and Simulation Organization, defines CMMS as being “The first abstractions of the real world that serve as a frame of reference for simulation development by capturing the basic information about important entities involved in any mission and their key actions and interactions” [1].

CMMS are simulation and implementation independent models. They are models of real world processes, entities, environmental factors, associated relationships and interactions constituting a particular set of missions, operations or tasks. The models strive to be as generic and applicable to as many scenarios as possible without any loss of critical knowledge. These views are consistent

with the views on conceptual modeling put forward in [2] where a conceptual model is considered to be “... the most fundamental model of an enterprise. A model of an enterprise as seen from the object perspective, including static structural aspects as well as dynamics and rules” The CMMS process is not just about the models themselves, but also includes a methodological framework for developing the models.

One benefit of the CMMS is that they act as a common description for all stakeholders of what is to be simulated, and thus serves as a bridge between the military experts and the developers. The military experts own the mission processes and are an authoritative source when validating the content of the conceptual models. CMMS also serves as a platform for communication among stakeholders working with the simulation models.

### 1.1 The FOI CMMS Project

The use of modeling and simulation in the Swedish Defense is increasing and, as a consequence, so are the concerns on how to secure, validate and maintain knowledge and how to keep efforts in doing so at a

minimum. In this context, the original CMMS concept was considered as a promising approach, and to see to which extent and under which circumstances it could be of interest for the Swedish Defense, the Department of Systems Modeling at the Swedish Defense Research Agency (FOI) carried out a study in 2001-2002. The main emphasis was on the aspects of reuse, interoperability and maintenance of conceptual models, as well as on the applicability and relevance for the Swedish Defense. The result of the study indicated that CMMS was an interesting concept but that it still was in an ambiguous, vague and partly unfinished state. An additional analysis, now called the FOI CMMS project, was initiated in order to further develop the concept and implement a framework that could support it. This work is still in progress.

The FOI CMMS modeling process follows roughly the original high level CMMS process, which starts in a knowledge acquisition (KA) phase, and ends in a knowledge engineering (KE) phase. The FOI CMMS process runs from a knowledge acquisition (KA) phase, via knowledge representation (KR) and modeling (KM), to a knowledge use (KU) phase [3]. The general flow of information in the FOI CMMS process is that partially structured information is gathered from some sources. For example, it could be interviews from witnesses of an event. The information is further structured according to an ontology (world view). That is, if the event is a combat scene or a terrorist attack, the reports from the witnesses are interpreted in terms of the ontology. A number of ontologies may be applicable for interpreting the same event, but ideally one is chosen, resulting in a common understanding of the event. The interpreted information is then subsequently transformed into a common format and stored as a reusable model. This common format is, in the FOI CMMS project, described by a Knowledge Meta Model (KM3).

Anything functionally equivalent to the KM3 was not specified in the original CMMS concept, but it was considered a necessity in the FOI CMMS process. The intention when producing the KM3 was not to construct a grand "unified model description language" applicable to the whole CMMS process. It rather represents one possibility to capture system structures and behavior in an object-oriented and rule based way. The purpose of this paper is to overview the main features of the KM3 and present some design choices done when constructing it.

## 2 The Knowledge Meta Meta Model(KM3)

The KM3 can formally be seen as a model specification, consisting of object-oriented concepts, primarily aimed at capturing different dependencies in and between activities. In practice this means that the KM3 is a specification for the creation of generic and reusable conceptual models of objects and processes of (military) interest. In other words, KM3 is a tool for structuring knowledge about objects and processes as conceptual models. The main objective with KM3 is to produce generic templates of knowledge based on types of instances. KM3 is mainly used as a specification for the construction of models, which in turn are used to model knowledge at an instance level. KM3 is, in this respect, a model for how to make models.

### 2.1 Design Choices

A simulation model expressed in the KM3 format should be minimally independent of which ontology was chosen as an interpretation of an event. Minimally, but not completely, since the KM3 has some ontological commitment resulting from being, for instance, object oriented, which implies that the ontologies used must recognize objects as part of their world view. More on quality of conceptual modeling and ontological commitment can be found in [4, 5].

KM3 is used as a means for capturing knowledge. Within the KM3, knowledge is defined to be the ability of an object to recognize alternative solution strategies for a problem to be a) applicable in a situation, b) select among them, and c) execute it. To understand all the effects and consequences of the performance of a strategy is considered to have deep knowledge about a situation. To see several applicable strategies in a situation, without being able to fully grasp their consequences are considered to having broad knowledge about the situation. Ideally, all objects should be defined having as broad and as deep knowledge as possible. This means that an object ideally should have a number of solution strategies available when confronted with a problem, and should also be able to fully understand the consequences of applying them.

The implication of this knowledge view is that it should be possible to define an object with its relations to other types of objects to the extent and level of detail that is needed for the modeling purpose. It should also be possible to add new information to the model as it becomes available, change available information as it is reassessed, and delete information as it becomes obsolete.

## 2.2 Demands

A number of demands were put on the KM3 that influenced its design. It should be a) *activity centric*, b) it was necessary to be able to capture the *static* and the *dynamics* aspects of objects in the same model, c) the models should be *reusable*, and it should be d) possible to model *uncertainty* of activity execution. Items a), b) and c) are further explained below, whereas item d) can be commented on directly. Uncertainty in execution of activities is in the current version of the KM3, not fully covered. It is modeled as an attribute of the rules which act as the start and stop conditions of an activity. For instance, we can express that the chance of an activity to start is 0.85 (on a scale from 0 to 1) and, when started, the chance of it stopping is 0.34. Clearly more modeling power than this is needed and part of the current research is devoted to it.

### 2.2.1 Activity Centric

A common notion is that a process is a partially ordered set of activities with a well-defined goal [6]. The activities are indivisible units of work that are combined to form processes. KM3 is primarily focused on the description of activities. The reason for this is to make the abstraction level more generic and increase the reusability of the models. A process, where the ordering of activities is significant, is considered less reusable than the modeled activities by themselves. Also, a characteristic of military operations is that they normally can be seen as combinations of activities rather than monolithic processes. The generic activities are combined to form descriptions of processes. For example, the process of driving a car from point A to point B under different weather conditions would result in a large number of process descriptions, whereas an activity centered description of the same process would result in a smaller number of models on a more generic level, where some minor part would be focusing on the differences between the processes.

Activities in KM3 are modeled as objects. To model the execution of an activity in KM3, all other objects that are influenced by the activity execution should be modeled as well. This includes other activity objects that may be seen as parts of a main activity. Every activity is modeled as a named set of state and rule descriptions. States are attributes of the activity object and rules are conditions for changing the attribute values. When a condition is true, values of the activity attributes changes and also attributes of other objects may be affected. In this way,

the dynamics of an entire model is captured through rules of execution of activities.

### 2.2.2 Static and Dynamic Descriptions

A static description of an object is an enumeration of its attributes and the domains from which the attributes can take their values. For example, a Person object can be described by giving it a name and filling the values for the attributes Length and Age. The permissible values are constrained by the domain definitions from which the attributes can take their values. This snapshot description is often sufficient when reasoning about the object. However, to be able to express how an object evolves, it is necessary to include dynamical descriptions of the object. In KM3 this is done by the inclusion of rules into the object descriptions. The rules determine how the states of objects changes as the objects evolve. Dynamic rules do not say anything about, e.g. the range of an attribute's values. They say which attribute value can legally precede another attribute value. Often the rules are formulated as being either obligations, recommendations, or prohibitions. An obligation means that an object state must follow another; a recommendation means that a state may follow another, and a prohibition means that a state must not follow another.

The static and dynamic views of an object are complementary. A Person object has the attributes Name, Length, and Age taking values from defined domains. This is the static view. It also has a rule, which says that by each passing of a year, its Age attribute should be incremented with 1. This is the dynamic view. Any modeling tool, including the KM3, should have the possibility to model both static and dynamic views of an object. In KM3 this is done through the modeling constructs attributes and rules.

### 2.2.3 Encapsulation

Similar to classic object orientation, the KM3 uses encapsulation to package and protect model elements. Being the fundamental elements, activity objects encapsulate the entity objects that are associated with the activities. The only part of the entities that are visible to the activity is the current state of the entity. No part of an activity is visible to an entity. The state of an encapsulated entity becomes automatically part of the state of the activity. The main benefit of adopting this view on encapsulation of model elements, together with focusing on activities, is that it results in modularization of the model, which increases its reusability.

### 2.3 The Model

The KM3 consists of four main components; 1) Model element, 2) Attribute, 3) State, and 4) Rules. Model element is analogous to the class concept in object orientation, i.e., it can be thought of as a grouping of similar object with similar properties. Attribute is analogous to multivalued attributes. State is analogous to a single valued attribute, and finally, rules are analogous to operations and are used to control activities and state changes. Following is a presentation of the concepts and their relations. The presentation is on a high level without going deep into the details. A detailed analysis can be found in [7].

#### 2.3.1 Model Element

Model element is the foundational construct of the KM3. Through this construct, the objects that are part of activities as well as the activity object themselves are modeled. Every model element is recursively decomposable and may be specialized through object oriented inheritance or aggregation. Common for all model elements is that they can be associated with attributes and states. Model elements can be of four different types: EntityType, RoleInAction, RoleInOrganizationType, and ActionType.

Figure 1 is read by starting from the ActionType construct (bottom right, in the figure). This corresponds to first finding a verb in a sentence. After this the other entities related to the activity are determined. This corresponds to finding the nouns and other word classes. If a required element is missing, information about it must be gathered, described and integrated into the model before finishing the modeling of the activity.

Deciding on which entities should be associated to the verb is governed by the RoleInActionType. Action role types can be grammatical constructs such as subject or object, or named roles e.g., "the vehicle" or "the wounded". Every action role is associated with an entity type that in effect classifies the role as to be played by an entity type described by a noun.

The difference between RoleInAction and RoleInOrganizationType is that different Types can play different roles in different contexts. A RoleInOrganizationType is characterized by rights (obligations, recommendations and prohibitions) of a position in a military organization. An RoleInAction defines the actors within a specific activity. In other words, it is possible to define similar names for concepts but their meaning is never the same. The rights associated

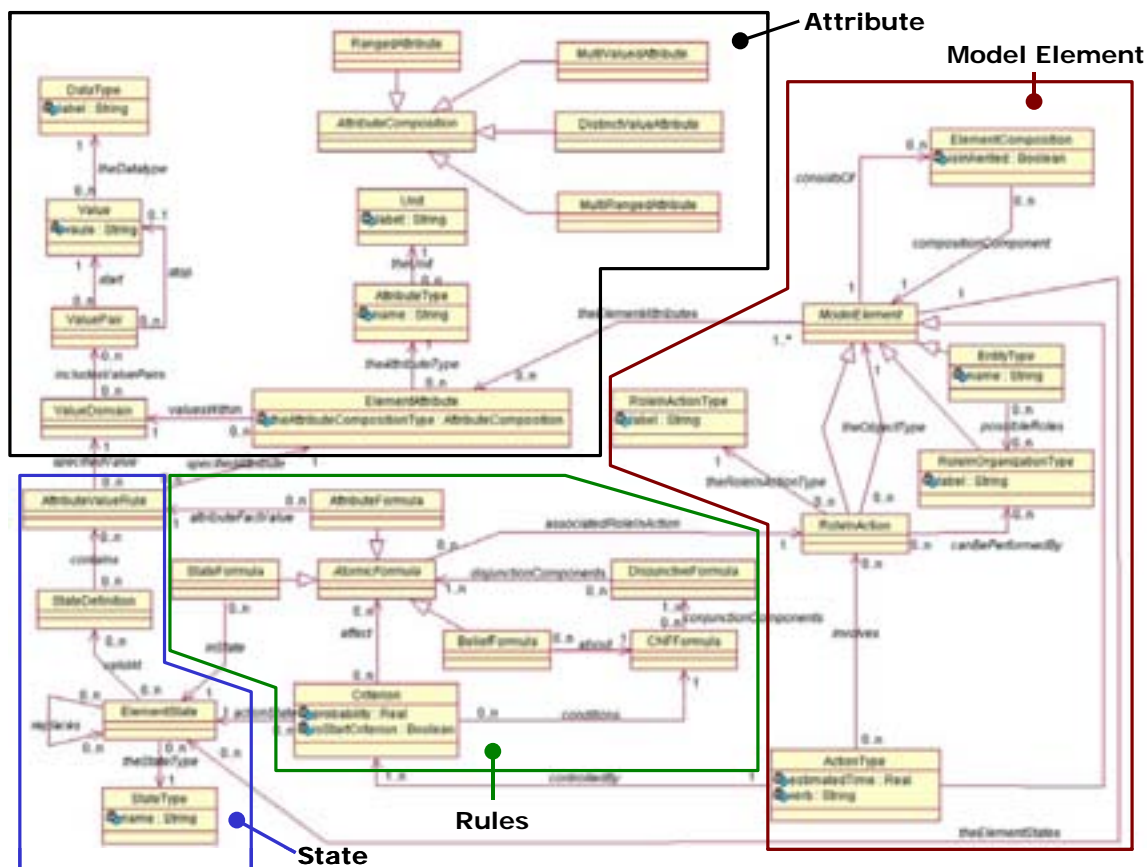


Figure 1 The Knowledge Meta Meta Model (KM3)

with an organizational role are a role in an activity.

It is important to realize that there is a difference in the level of abstraction between activity roles and the other model elements. A role in an activity denotes which type of entity it is played by. Activities associated with activity roles are not explicitly expressed in the model but are derived by the association.

When the activity roles has been fully determined, the pre- and post-conditions as well as states of the activity is specified. The definition of entity types are done independently of definitions of activities and each entity type can be associated with one or several different activities.

### 2.3.2 Attribute

An attribute describes an optional, measurable characteristic of a model element. Through a measurement the model element is classified according to some unit of measure. An attribute is always associated with at least one model element but there is a possibility that an attribute can be shared among several model elements. An attribute is always associated with one of the following classification names: Interval, MultiInterval, Value, or MultiValue. Interval means that only an interval is a valid value for the attribute. Multi-interval means that several intervals simultaneously are valid values for the attribute. Value means that only one point value is valid for the attribute. Multivalue means that many values are valid for the attribute simultaneously.

Value domains establish the extension of the attribute given its classification. A value domain is defined by values giving the upper and lower bounds of its applicability. If the upper value is 'null', then the value domain is a distinct value. Otherwise it is an interval.

Every value in a value domain is associated with a data type. This design is for e.g., enabling the automatic conversions of units of measure. For example, the attribute 'length' measured in the unit centimeter is associated with the model element 'person'. The attribute should be classified as a Value, since we only want a person to be of one length at a time. Our value domain will thus be defined through a pair of values where the lower end is, say, 0 and the upper end is 250. All values are of data type Integer. In this example we have created an attribute that can be described by a number (at least 0, at most 250) and a unit (centimeter). The attribute can take exactly one value at a time.

### 2.3.3 State

A state is the set of values of a non-optional attribute of an Activity Role model element. The values of the non-optional attributes comprise the state of the model element. If an Attribute is associated with a model element (an Attribute being optional), then the state of the attribute is part of the state of the model element. Every state is associated with an activity Role model element and is analogously associated with a type.

As values of states changes, one state replaces another. All value changes are taking place in Roles participating in Activities. Permissible state changes will take place according to the rules that constrain all possible state changes. A valid state of a Role is a state in which all values are in compliance with a state definition. That is, all optional and non-optional attributes of a Role must have values that are permissible in a state definition.

### 2.3.4 Rules

The dynamics of model elements in KM3 are captured by rules. Through the specification of rules, all changes of model elements are described. A rule specifies the conditions under which an activity starts and ends. The shape of a rule is a pair, where the first element represents an activity role, and the second element represents an atomic formula. An atomic formula is a statement about the state or attributes of a role. Every rule can be classified as being a state-formula, an attribute-formula, or a belief-formula. Beliefs do not necessarily need to be objectively true to trigger an activity, but rather reflect the subjective views of a person.

From atomic formulas more complex formulas can be constructed by connecting them conjunctively or disjunctively. A disjunctive formula is created by connecting conjunctive atomic formulas by means of OR-connections (a.k.a. disjunctive normal form, DNF). A conjunctive formula is created by connecting atomic disjunctive formulas by means of AND-connections (a.k.a. conjunctive normal form, CNF).

Criteria are created from atomic rules. A criterion determines the start and stop conditions for activities, which in turn results in state changes of the activity, and possibly also in other model elements. In other words, the state changes are the effects from some activity criterion becoming true.

## 2.4 Example of a KM3 structure

After this overview of the main KM3 concepts we exemplify its use as an analytical tool in a hypothetical scenario. This is an example of a (simplified, for brevity reasons) structure where concepts from the KM3 have been used to analyze a situation report. The work has been done manually.

This small example is a very simplified snippet of a larger scenario designed to test various types of modeling difficulties. It is not the main intended use of the KM3 as has been explained and will also be elaborated upon in a

future technical report planned for publishing by the end of this year. The structure is lightly commented to make it easier for the reader to follow.

### 2.4.1 Scenario

The scene for this scenario is Kosovo where NATO's and other nation's forces are involved in a joint Peace Support Operation (PSO). The situation in this hypothetical scenario is that a patrol from the Swedish peace keeping force has discovered a break-in in a weapons depot and filed a report into the information system. An intelligence officer receives the report and starts a further investigation.

---

Part of the original scenario text:

... An officer at the Intelligence department receives the information automatically due to the predefined subscriptions for his role. The officer creates a main information element to be used as a point of reference when the errand is explored further. To find more information regarding the case, the officer searches the GIS-system with a special interest for the area around village Y. He finds a weapons depot close to it. The information in the GIS concerning the weapons depot is interlinked with additional information. This information concerns a particular individual, Major C. S. He has connections with organized crime in Country X where he has lived. He is the leader of the military unit responsible for the weapons depot. ...

---

Analysis of the text using the KM3:

*/\* the entity type Intelligence department is a part of the Swedish defence\*/*

ET : Intelligence department

ElementComposition :: part of : < Intelligence department, Swedish defence>

*/\* the entity type NBI is a part of the Information and communications system\*/*

ET : NBI

ElementComposition :: part of : <NBI, Information and communication system>

*/\* the entity type GIS is a part of the NBI\*/*

ET : GIS

ElementComposition :: part of : <GIS, NBI>

*/\* An ActionType related to another action type (occurring After)\*/*

AT : Searching ET : GIS for additional information

Time : May 2002, AFTER AT : Main information element creation

RoleInAction : <searcher, subscriber>

RoleInOrganizationType : <subscriber, ET : Subscribing officer>

*/\* Start and Stop criteria for the action \*/*

Criterion : (prob 1, isStartCriterion T,

[ET : Main information element AND ET : relevant information]

ActionState : searching for additional information

Criterion : (prob 1, isStartCriterion F,

[ET : Main information element AND ET : relevant additional : T] OR [ET : Main information element AND ET : relevant additional : F]

ActionState : search for additional information has occurred

/\* Effect of the action. A "case file" is updated. Additional EntityTypes with connections are discovered \*/

ET : Main information element

State : Updated with additional information

ET : Weapons depot

Attribute : Close to ET : Village Y

Attribute : Responsible commander is ET : Major C. S.

ET : Major C. S.

Attribute : Connection to organized crime, Organization

Attribute : Lives in Kosovo, Year

StartValue : 1995

StopValue : Now

Attribute : Lived in Country X, Year

StartValue : Before 1995

StopValue : 1995

Attribute : Leader of a military unit responsible for guarding ET : Weapons depot, Military unit

### 3 Conclusions and Future Research

We have in this paper presented the vision Conceptual Models of Mission Space (CMMS) and the Swedish Defense's interest in this vision. As part of the CMMS process we have identified the need for a common structure for describing conceptual models. We have designed one such structure, called Knowledge Meta Meta model (KM3), to take this role. A number of demands were put on the KM3; the models produced using it should be activity centric, able to capture the static and the dynamics of objects in the same model, be reusable, and it should be possible to model uncertainty of activity execution. We have presented the main features of the KM3, and discussed how the demands influenced its design.

Future research involves a validation of the CMMS process starting from an only partially structured information input, and ending in an information structure ready for use in the construction of a simulation model. The validation is expected to improve the CMMS process, and also the KM3. At this point we already know that we need to be able to model uncertainty and risk in a more elaborate way than we do today. We also need to clearly determine the exact relations between the different parts of the CMMS process and its artifacts, and also the uses of them. For instance, in this paper we have used the KM3 as an analysis tool which is not its main intended function.

Another area for future research is how to be able to determine the right level of abstraction when

modeling. Clearly different stakeholders put different demands on models of the same event. For instance, a commander's view of an event is very different from that of an infantry soldier and their different views result in different simulation models. A third area of future research concerns the storing and maintaining of simulation models. This also includes work on languages for querying the model store.

### References

- [1] *Conceptual Models of the Mission Space*. DMSO, rev.0.2.1, 1997. [www.dmsol.mil/briefs/msdocs/policy/msmp.pdf](http://www.dmsol.mil/briefs/msdocs/policy/msmp.pdf). 2002-09-03
- [2] Boman, M., Bubenko, J., Johannesson, P., Wangler, B. *Conceptual Modelling*. Prentice Hall, ISBN:0-13-514879-0. 1997
- [3] García Lozano, M., Mojtahed, V., *A Process for Developing Conceptual Models of the Mission Space (CMMS) – From Knowledge Acquisition to Knowledge Use*. Report ID 05F-SIW-038. Presented at SISO 2005 FALL SIW Workshop. Orlando, Fla., 2005.
- [4] Lindland, O. I., Sindre, G., Sölvberg, A. *Understanding Quality in Conceptual Modeling*. IEEE Software, Vol. 11, no:2, pp. 42-49, 1994.
- [5] Opdahl, A. L., Sindre, G. *Concepts for Real-World Modelling*. Colette Rolland, François Bodart, Corine Cauvet (Eds.): Advanced Information Systems Engineering, CAiSE'93, Paris, France,



June 8-11, 1993, Proceedings. Lecture Notes in Computer Science Vol. 685 Springer, ISBN 3-540-56777-1, 1993,

[6] Hammer, M., Champy, J., *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperBusiness. ISBN 0-88730-640-3, 1993

[7] Mojtahed, V., Garcia-Lozano, M., *Kunskapsrepresentation (KR) och Kunskapsmodellering (KM) i CMMS-konceptet*. Metodrapport FOI-R—1433—SE. ISSN 1650-1942. 2004

### Author Biographies

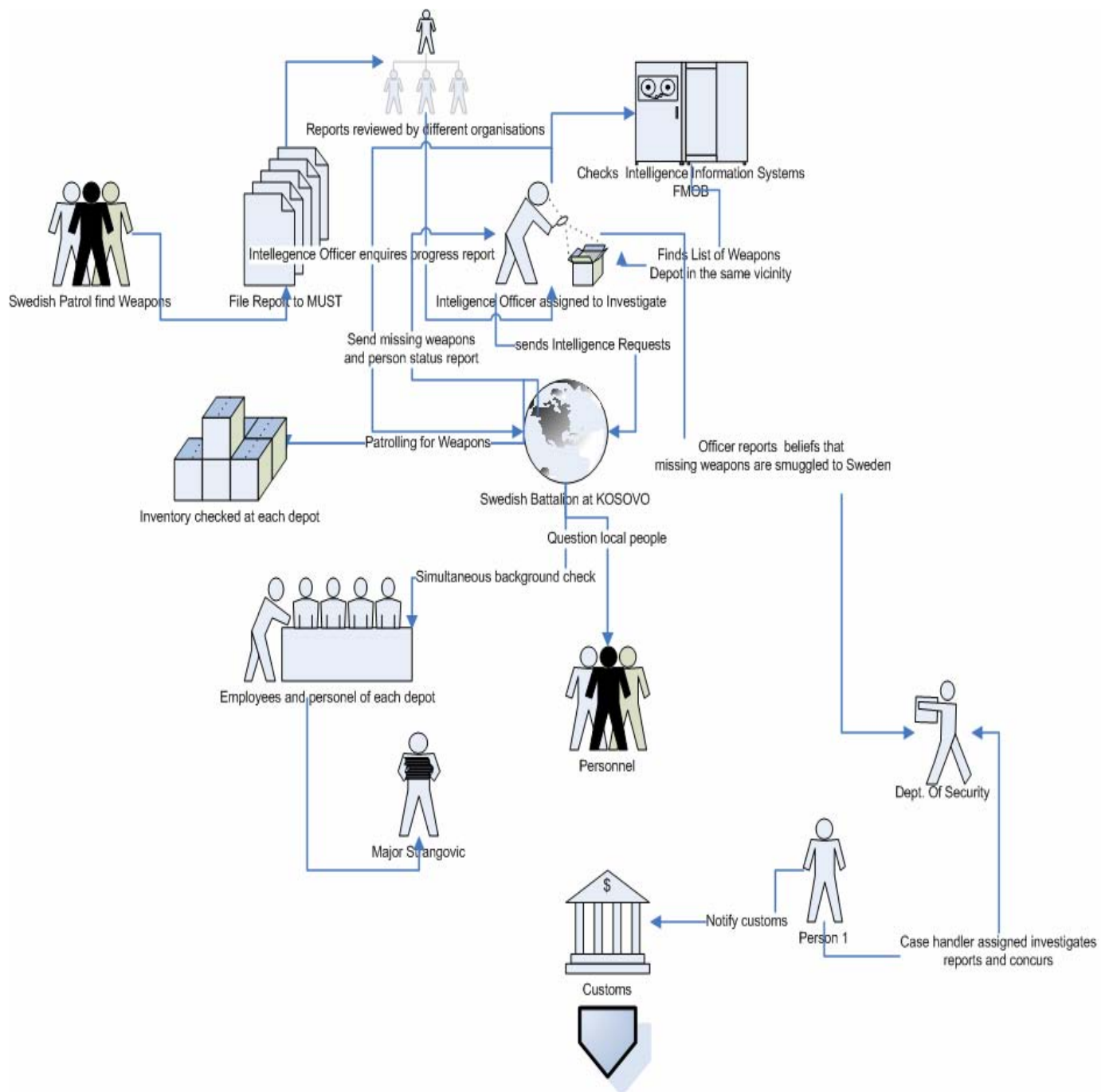
**BIRGER ANDERSSON** is a Research Engineer at the Royal Institute of Technology in Stockholm. He is, since the beginning of 2005, a member of the FOI CMMS project at the Department of Systems Modelling at the Swedish Defence Research Agency (FOI).

**MARIANELA GARCÍA LOZANO** is a Research Engineer at the Department of Systems Modelling at the Swedish Defence Research Agency (FOI) and she has been a part of the CMMS project since its start in 2001. She holds a Master of Science in Computer Science and Engineering from The Royal Institute of Technology (KTH), Stockholm, Sweden. Her research interests are in the areas of Conceptual Modelling, Distributed Computing and Symbiotic Simulations.

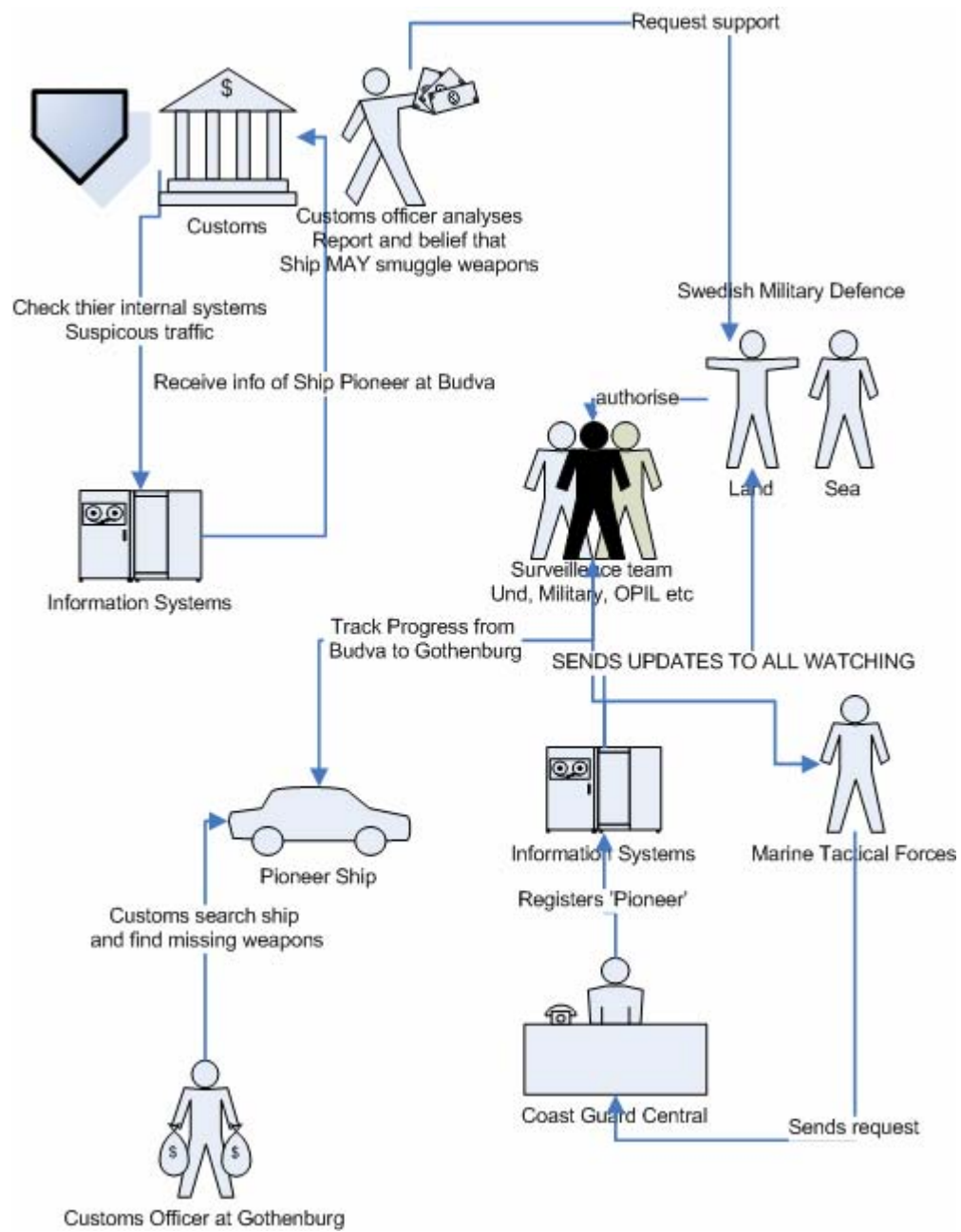
**VAHID MOJTAHED** is a senior research officer working at the Department of Systems Modelling at the Swedish Defence Research Agency (FOI). He has been working on modelling and simulation for the past ten years and has led the CMMS project at FOI since 2001. He holds a Master of Science in Computer Science and Engineering from Chalmers University of Technology, Gothenburg, Sweden. His research interests are in the area of Conceptual Modeling, Simulation Framework and Information Operations and Warfare.



## Appendix D – A graphical use case scenario diagram



**Figure 1:** Part I of the Use Case Visualisation



**Figure 2:** Part II of the Use Case Scenario Visualisation