The Knowledge Use in DCMF

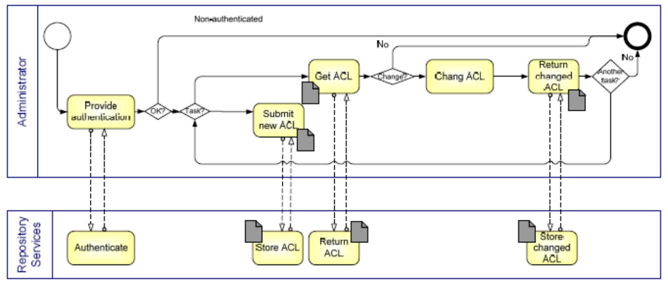
Repository, Processes and Products

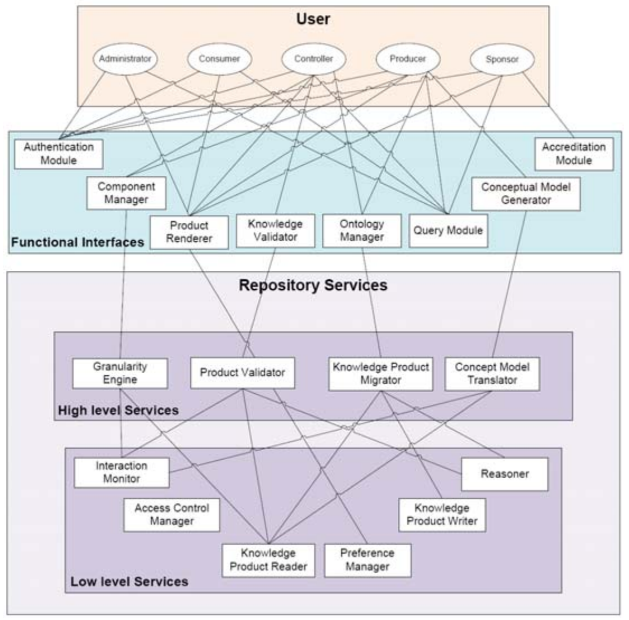
VAHID MOJTAHED, EDWARD TJÖRNHAMMAR

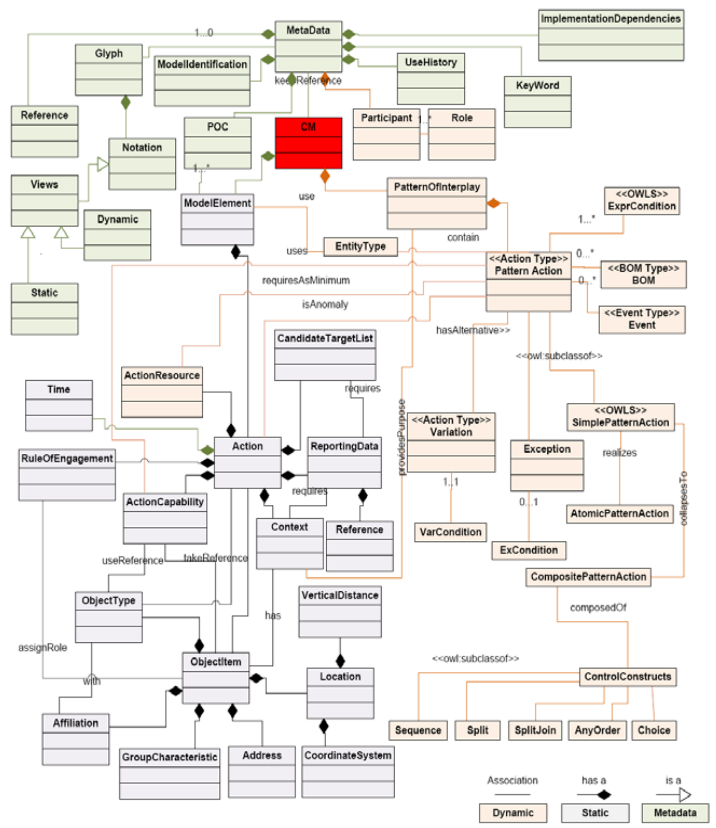
FOI - SWEDISH DEFENCE RESEARCH AGENCY

JELENA ZDRAVKOVIC, ALI HANZALA KHAN

KTH – ROYAL INSTITUTE OF TECHNOLOGY







FOI-R--2606--SE User report **Information Systems**

ISSN 1650-1942 November 2008

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Defence Research Agency
Information Systems
SE-164 90 Stockholm

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

www.foi.se

Vahid Mojtahed, Edward Tjörnhammar
FOI - Swedish Defence Research Agency

Jelena Zdravkovic, Ali Hanzala Khan
KTH – Royal Institute of Technology

The Knowledge Use in DCMF

Repository, Processes and Products

Titel	The Knowledge Use in DCMF
Title	The Knowledge Use in DCMF
Rapportnr/Report no	FOI-R--2606--SE
Rapporttyp Report Type	Användarrapport User report
Sidor/Pages	51 p
Månad/Month	November
Utgivningsår/Year	2008
ISSN	ISSN 1650-1942
Kund/Customer	FM
Forskningsområde Programme area	2. Operationsanalys, modellering och simulering 2. Operational Research, Modelling and Simulation
Delområde Subcategory	21 Modellering och simulering 21 Modelling and Simulation
Projektnr/Project no	E7141
Godkänd av/Approved by	Martin Rantzer

FOI, Totalförsvarets Forskningsinstitut	FOI, Swedish Defence Research Agency
Avdelningen för Informationssystem	Division of Information Systems
164 90 Stockholm	SE-164 90 Stockholm

Summary

Support towards developing interoperable simulation assets has recently been the focus area in the Modelling and Simulation (M&S) community. In that direction conceptual modelling has been recognized as an important step within the simulation development process. Conceptual models will help to improve common understanding between all stakeholders involved in simulation development and thereby being keys to the transformation from user requirements to implemented simulations. DCMF - Defence Conceptual Modelling Framework - is the Swedish Defence Research Agency (FOI)'s proposal on how to deal with conceptual modelling issues within the military domain. The DCMF is also a project at FOI which tries to create a framework for capturing, analysing and representing conceptual models. These conceptual models are formalised descriptions of military missions, operations or tasks. As such they should be kept generic and applicable to as many military scenarios as possible without any greater loss of critical information.

Research on conceptual modelling has been going on at FOI for the past several years within the DCMF project. The project has succeeded to put the basis for a common methodological framework including a conceptual modelling process and a set of other necessary components enabling the development of conceptual models of military operations in a formal way. However, there is still a good amount of work remaining before the framework can be put into operation. The DCMF Process comprises of four major phases which have not been explored to the same level of detail. While the understanding of the early phases is acceptable, this is not the case for the late phases meaning that neither the necessary components nor the detailed process for the Knowledge Use phase has been clearly examined yet. For that reason, the DCMF project has planned to have a clear focus on how such a repository should be designed and organized to warehouse all the DCMF artefacts during its new three years of commission, 2008 – 2010.

The underlying document is our first report of the study and the analysis has been carried out with a deeper focus on the later phases of the DCMF Process, and it will therefore discuss the important issues closer to user and usage of this framework. As such, this study tries to draw a map of the Knowledge Use phase in DCMF including: identifying necessary components, defining detailed processes over the interactions between main users and the repository services, introducing architecture for the repository as well as defining the potential artefacts which will be stored and maintained in the repository.

We will start by exploring the Knowledge Use phase in detail, and then describing it as a process from four different perspectives: organizational, functional, informational and behavioural thus giving us the ability to model and discuss the aspects of the knowledge use distinctively, from each of the outlined modelling perspectives. Furthermore, we will then take a closer look at the most important products of DCMF being; Knowledge Instances (KI), Knowledge Components (KC) and Conceptual Models (CM). These artefacts will in turn be explored in detail in order to be able to propose the necessary set of the information they have to capture as well as the adequate holders needed for keeping track of that information. These products will be stored, maintained and retrieved by different stakeholders from the DCMF repository. The services and functions which the DCMF repository should provide to make this possible, as well as how these services and functions should be structured in order to facilitate the usage will be discussed.

In addition, with this report we wish to inform the community of practice for Conceptual Modelling in M&S about our course of action, as well as presenting the first step towards creating a requirement specification for our further development work.

Keywords:

DCMF, DCMF Artefacts, DCMF Processes, DCMF Repository Services, Conceptual Model, Knowledge Component, BOM, BOM++.

Contents

1 Introduction

1.1 Background

Using Modelling and Simulation (M&S) in the military domain is today a matter of course, but the increasing use of M&S puts high demands on how knowledge is used and managed. As known, a simulation can never deliver an answer of a better quality than the knowledge we model and put into the simulator. Thus, the major challenges are how to acquire, validate and maintain knowledge and how to achieve this with minimum effort.

In order to address issues related to knowledge bases for modelling and simulation, the Swedish Defence Research Agency (FOI) has, since 2003, done research in this area. The comprehensive objective for this research has been to draw up a proposal of a practical and useful modelling process managing knowledge all the way from acquisition to use. The FOI's proposal to how to deal with Conceptual Modelling issues in military business is today defined in a concept called Defence Conceptual Modelling Framework (DCMF). An introduction to this framework has been given in a more extensive FOI methodology report [Mojtahed et al, 2005].

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

1.2 The DCMF project

The Defence Conceptual Modelling Framework (DCMF) is also the name of an ongoing project at FOI aiming to create a framework, including a process for developing the conceptual models of military operations. The conceptual models created by this process are formalised descriptions of real world entities, processes, environmental factors, associated relationships and interactions constituting a particular set of missions, operations or tasks. Constructed models are supposed to be generic and applicable to as many scenarios as possible without loss of critical knowledge. One of the most important benefits of such models is that they may serve as the same basis for all stakeholders of what is to be simulated thus acting as a bridge between the military experts and the simulation developers.

The overall objective for the DCMF is to capture authorised knowledge of military operations as well as to manage, model and structure the obtained knowledge in an unambiguous way. To preserve and maintain the structured knowledge for future use and reuse is also within the scope of this objective. The other goals of DCMF are to achieve the following additional advantages; firstly to produce a disciplined procedure by which the simulation developer is systematically informed about the real world problem to be synthesised, secondly to deploy a set of information standards which the simulation subject matter expert can employ to communicate with and obtain feedback from the military operations subject matter expert, thirdly to provide a real world military operations basis for subsequent, simulation-specific analysis, design, implementation, verification, validation, and accreditation and finally to be the means for establishing reuse opportunities in the eventual simulation implementation by identifying commonality in the relevant real world activities.

1.2.1 Focus of this year's work

As mentioned, the research on Conceptual Modelling has been going on at FOI since 2003 within the DCMF project. The project has succeeded to form the basis for a common methodological framework including a conceptual modelling process and a set of other necessary components. These will enable the development of conceptual models of military operations in a formal way. However, there is still a good amount of work remaining before the framework can be put into operation. An operational use of such a framework also requires shaped and well-tried components.

The DCMF Process comprises of four major phases [Mojtahed et al, 2005], which has not yet been explored in the same depth. While the understanding and thereby the ability to survey the early phases of the process is acceptable, that is not at all the case for the later phases. That is to say that neither the necessary components nor the detailed process for the Knowledge Use phase has been clearly examined yet. The components identified and created in the early phases, such as Knowledge Instances and Knowledge Components, need to be stored in a repository for both further developments during the process and further use and reuse by the DCMF end users. For that reason, the DCMF project has planned during its three years commission, 2008 – 2010, to have a clear focus on how such a repository should be designed and organized to be able to warehouse all the DCMF artefacts as well as serve all the needs of DCMF users.

Given this background, the focus of DCMF activities under 2008 has been limited to two major tasks:

1- Explore the later phases of the DCMF process and the components needed in those, as well as establishing a requirement specification on both the repository and the potential artefacts to be stored there.

2- The NATO Research and Technology Agency has set up the Modelling and Simulation Group (MSG) 058 titled “Conceptual Modelling for M&S” to produce guidance on conceptual modelling for M&S development within NATO. NMSG-058 is committed to clarify the process and results of “Conceptual Modelling”, and to recommend best practice to generate, document, and use conceptual models. The second task for the DCMF project is to follow this group and actively contribute Swedish national experience, especially the one worked up during the past 5-6 years within the DCMF project.

1.2.2 Knowledge Use phase in focus

The DCMF process is an iterative process and comprises in large of four main parts; Knowledge Acquisition (KA), Knowledge Representation (KR), Knowledge Modelling (KM) and Knowledge Use (KU) phases. These phases generate many artefacts but from the Knowledge Use perspective there are three of the most important ones; Knowledge Instances (KI), Knowledge Components (KC) and Conceptual Models (CM). These three products will stand for the major content of the DCMF repository. There may be much more metadata and records needed to be maintained in the DCMF repository in order to achieve the desired level of composability and interoperability as well as to live up to requirements of different stakeholders.

The roles and responsibilities of the DCMF stakeholders will naturally drive our work with the Knowledge Use phase. There are many potential users having various missions and tasks, and many of them definitely need to be supported by certain tools and views. Some examples are; an evaluator (VV&A-agent) may use the conceptual model for verifying and validating the implemented models, and as a basis for accreditation of simulation systems; simulation engineers may use the conceptual model as a means and a guide for their modelling activities; a mission planner can use the conceptual model as support when developing scenarios for simulations; etc. All these different stakeholders normally have different needs and must consequently be served by different functions as well as different views of the same CM.

Thus, the DCMF Repository is a central component in the DCMF in which all DCMF products are stored for future use and reuse. Thereby, it requires a stable structure and efficient tools for the management of the data base, including creation, modification and deletion, etc., of products in the database. Another important part of the repository is the functionality of searching for available products. This search should not only be keyword-based but also enable a semantic search enabling the search for concepts and relations between knowledge instances, components and models, this for example being crucial for composability. All these issues will be discussed in this report.

1.2.3 NATO MSG-058 cooperation

The NATO Research and Technology Organization (RTO) Modelling and Simulation Group (MSG) recognized the important role that conceptual modelling could play in interoperable simulation assets, and thus set up the NMSG-058 task group to investigate this area. NMSG-058 started in July 2007 and will finish in spring 2010. NMSG-058's mission is to clarify what a conceptual model is and what it represents. For that reason, this group is investigating methodologies, simulation and software engineering processes, initiatives and technologies, as well as the content of conceptual models. The final objective of NMSG-058 is to provide guidance to the M&S community on conceptual modelling. If possible, a future guidance will be proposed to the international community for standardization via the Simulation Interoperability Standards Organization (SISO). NATO believes that a NATO Task Group, working in conjunction with the SISO, has the unique position to develop a standard that will be used by multiple nations, thus meeting the reusability and interoperability goals.

The NMSG-058 task group also expects that a recommended practice for conceptual modelling will increase interoperability, reusability, and relevant employment of simulations. The reason is that such a document guides multiple stakeholders through an evolutionary process resulting in sufficient common understanding of the semantic significance of simulation representation. NMSG-058 will therefore draft a guide covering:

- the conceptual modelling process;
- the conceptual model artefact;
- the role within simulation development for the defence establishment; and
- supporting standards.

The task group will mainly deliver several interim publications, a draft guidance and a final report.

The current status of the work is perhaps best explained by mentioning that the task group has up to now held quarterly meetings (almost) at which the following topics has been elaborated:

- stakeholder analysis and context;
- specification of a conceptual modelling process;
- specification of a conceptual model artefact; and
- relationship to relevant standards.

Having identified stakeholders and their respective needs and interests, having derived requirements for conceptual model processes and artefacts and, too, having analyzed the relevance of existing standards, the MSG-058 is proceeding to draft best practice guidance for the generation, documentation, and employment of conceptual models for military domain simulations. However, in order to complete this task and to achieve the desired objective several activities are necessary and some significant challenges remain to be resolved. An example is that a definitive commitment must be made to the specific process elements and semantic content of the consequent conceptual model artefacts that may reasonably be expected to be sufficient to meet the needs of the several relevant stakeholder communities.

The group presented a paper [08F-SIW-038] titled *"M&S Conceptual Modelling for the Defence Establishment"*, covering its scope, preliminary results and the work plan for review by the community of practice, at the latest Simulation Interoperability Workshop. However, in this report we will not go any deeper than this into the NATO group's activity but rather focus on the outcomes of the work recently done within the DCMF project (see next section).

1.3 Objective of this report

As mentioned earlier, an extensive introduction to the DCMF concept will be found in [Mojtahed et al, 2005]. In that, as well as in many pervious publications (among others [05F-SIW-038], [Yi et al, 2006], [06E-SIW-028] and [Mojtahed et al, 2008]), we have presented the properties, characteristics, design and experiences of the DCMF as a method. This material should be able to serve as a foundation for anyone who wishes to better understand the framework and the method.

In this report, however, we are not going through the framework again but only focus on the later phases of the DCMF Process and, too, discuss important issues closer to the user and the usage of this framework. Thus, the main objective of this report is trying to draw a map over the Knowledge Use phase in more detail than before, which is to define both the DCMF Repository and the artefacts to be warehoused by the repository. Being able to do that, we have to start identifying the end users of the Conceptual Model (CM) and their requirements. That is necessary for deciding the content of the repository. Moreover, we need to analyse and carefully define the current output of the DCMF Process; its products. Last but not least, we have to define the repository, its architecture and all other crucial issues; its content language, format, services, etc.

To sum up we can say that there are three major objectives to be achieved in this report:

- the first one is of course to report to our principal assigner the work status,
- the second one is to inform the community of practice for CM within M&S and our course of action in hope to get valuable feedback,
- and last but not least, to take the first step towards creating a requirement specification for our further development work.

1.4 Report outline

After this overview, Chapter 2 starts with a brief introduction of the DCMF Process followed by a deeper study of its last phase, the Knowledge Use phase. This phase will then be explored as a process of its own by introducing a multi-aspect process description framework from the four different design perspectives: organizational, functional, informational and behavioural.

Chapter 3 begins with a brief introduction to DCMF products and a discussion on their features. Then, the two most important artefacts: Knowledge Components (KC) and Conceptual Models (CM) will be explored and both their content and format will be discussed.

Chapter 4 presents our first proposal for the DCMF Repository as well as discussing its features and services in moderate detail. After that introduction it will bring out an architecture demonstrating how these features and services will be orchestrated to serve the DCMF users requirements.

It is of significant importance to state that the ideas which will be presented in the rest of this report are under construction and therefore a subject to change; the presented solutions are not yet settled and should thus not be taken as well-tried ones.

Chapter 5 will finally conclude the report by summing up the results and outcomes as well as pointing out difficulties and challenges and, too, mention some future work.

2 Exploring User Processes in the KU Phase

2.1 Introduction

In this section we briefly describe the placement and the objective of the KU phase within the whole DCMF process.

2.1.1 Overview of the DCMF Process

DCMF (Defence Conceptual Modelling Framework) is a framework for the development of conceptual models in the military context. As mentioned in the introductory chapter, the overall goals of DCMF are to capture authorized 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.

The final result of the DCMF - Conceptual Models (CM), are defined as implementation-independent functional descriptions of the real world processes, entities, and environmental factors associated with a particular set of military missions. These descriptions could serve as a frame of reference for simulation development by capturing the basic information about the important entities involved in any mission as well as their key actions and interactions.

The DCMF process comprises of four major phases: Knowledge Acquisition, Knowledge Representation, Knowledge Modelling and Knowledge Use.

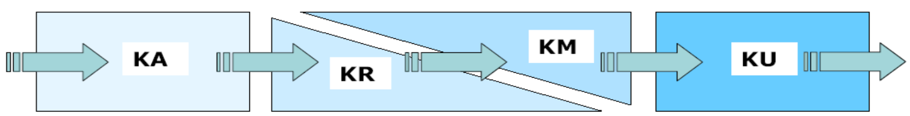


Figure 2-1: Four phases of the DCMF process

Here, we briefly describe each of the phases in Figure 2-1. The Knowledge Use phase is the focus of this report, and as such, it is further explored in detail. For a detailed description of the other DCMF phases, the reader is referred to [Mojtahed et al, 05].

The objective of the *Knowledge Acquisition (KA)* phase is, as the name suggests, acquiring required information from various sources. The important issues involve the delimitations of the requirements scope, the identification of authorized knowledge sources, and finally, the actual engineering (i.e. data acquiring, gathering and documentation).

The purpose of the *Knowledge Representation (KR)* phase is to analyze, structure and formalize the acquired information. This step is about transforming the human-readable information (interview data, case scenarios) to a machine-readable version. The structuring and formalization of information should be performed 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 *Knowledge Modelling (KM)* phase emphasises the semantic analysis and modelling of the information. In this phase pragmatics is also an important part of the analysis and modelling. Another task of the KM is to merge the new conceptual models or components with the ones previously created.

The *Knowledge Use (KU)* is the final phase of the DCMF process involving the actual use of the modelled knowledge. In this phase the connection is strongest to the end-user, and therefore, it is of great interest to visualize the acquired and modelled knowledge in different ways depending on the user's purpose and rights. To enable usage and re-usage of that knowledge, it must be stored in a repository (i.e. DCMF Repository).

Following what is previously outlined, the focus of this report is *to describe the different **uses of the information** (and knowledge) maintained by the DCMF repository*.

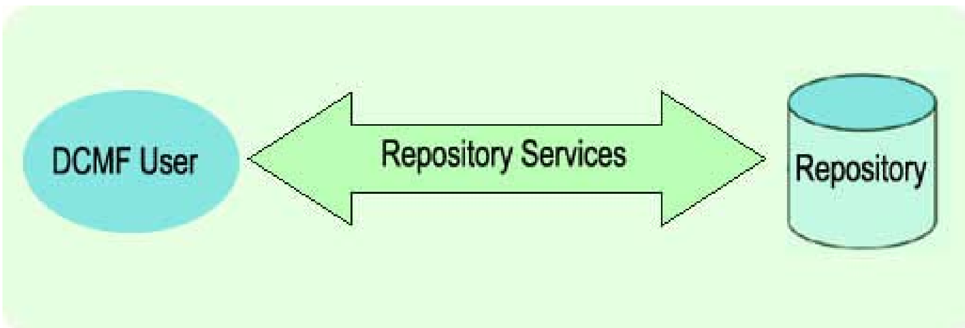


Figure 2-2: KU Phase in relation to the DCMF Repository

2.2 Design Framework for the KU Phase

In this part, we first discuss the general requirements for the KU phase from the perspective of the DCMF. Then we introduce a multi-aspect process description framework. The framework is in turn used for defining the KU phase in detail, that is, from four different process-modelling perspectives.

2.2.1 Requirements for the KU Phase

As stated in the previous section, the Knowledge Use is the last phase of the DCMF process, concerned to presenting the knowledge, gathered and assembled in the previous DCMF phases, to the end-users.

The overall requirement of the KU phase is to provide different information to a number of different end-users, using many usage cases.

The outlined requirement states that, during the KU phase of the DCMF process, as different users will be considered, those users will require different functions and procedures for gathering and scoping the information from the repository leading to a need to consider and analyze the KU phase as ***a process*** itself.

2.2.2 Process Modelling Framework

Processes coexist in the environment of organizations, people and resources. Following [Davenport et al, 92], a process *"is a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs; a structure for* *action"*. There are many other definitions given for a process. Although the main idea behind all of them is that a process is a sequence of activities creating value by transforming an input into a valuable output. Process models provide the capability to analyze the included activities, information and applied business rules.

In this report we will utilize a multi-conceptual framework to classify different aspects constituting a process model (specification). The framework is proposed by Curtis [Curtis et al., 92] and comprises of four perspectives: *organizational, functional, informational* and *behavioural*.

The organisational perspective describes the distribution of the responsibility for executing process activities. The main focus here is on the notion of the *actor*. An actor can be an organizational unit, a human being or a software system. *Actors are commonly modelled as* ***roles****, that is, as a set of the functions that an actor is responsible for performing*. Using the organizational perspective, it is possible to dedicate and control responsibilities of parties engaged in a process.

The organizational perspective of the KU Process: the end-users of the DCMF, as mentioned earlier, are numerous. They may include Orderers and Buyers, whose responsibility is to describe extent and delimitations of the conceptual modelling requirements, Analysts, Designers, Developers, Programmers, who understand the components of military operations and simulation processes, Researchers, Engineers, who use the framework as a tool and aid in the conceptual modelling work, War fighters, who formalize and describe their demands and evaluate the results and, finally, Evaluators and VV&A-agents [Yi et al, 06] who verify and validate models and use them as a basis for the development of simulation models.

Considering different information usages in the KU process, the outlined users may be aggregated into five distinct actors: *Sponsor, Producer, Consumer, Controller* and *Administrator* [Mojtahed et al, 05].

- *Sponsor*: this actor (i.e. role) initiates the whole DCMF process, by requiring modelling of a military case. Thereby, when a knowledge product is registered in the repository, the Sponsor must assess and approve it. After an approval, the product is made accessible to the Consumer. If a knowledge product is not entirely approved, it must be returned to the Producer for modifications.
- *Controller*: this actor is responsible for Verifying, Validating, and Accrediting (VV&A) of the products that are stored in the Repository. Verification comprises the control of the product (i.e. model) correctness in respect to the modelling rules. Validation considers the capability of the model to present, in an experimental context, the behaviour of its original. Finally, the accreditation is the activity denoting that the product has been sufficiently verified and validated. Such a product is then given to the Sponsor for usage approval. All notes and comments made the Controller during the verification and validation process should also be stored associated to the product, in that way ensuring traceability of changes.
- *Producer:* this actor has the responsibility to design all the DCMF products that are registered and stored in the Repository. The products of interest include the knowledge products, their metadata, ontologies and the analysed reports (i.e. reports for example

analysed by KM3- Knowledge Meta Meta Model and/or 5W:s – Who, What, When, Where, Why [Mojtahed et al, 05]. Those products are further given to the Controller for accreditation.

- *Consumer:* The consumer is the actor using the Repository to locate and use knowledge products. 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. The results of a search should also contain the data associated with a knowledge product, such as ontologies, history of changes, and header data.
- *Administrator:* this actor/role is responsible for dedicating and controlling the access to the DCMF product stored in the Repository for the other roles: Producer, Controller, Sponsor and Consumer. Thereby, the main task of the Administrator is to create an access control list for a recently created product and, in addition, to maintain existing products' access control lists.

The responsibilities of the KU organizational roles are briefly illustrated in Figure 2-3:

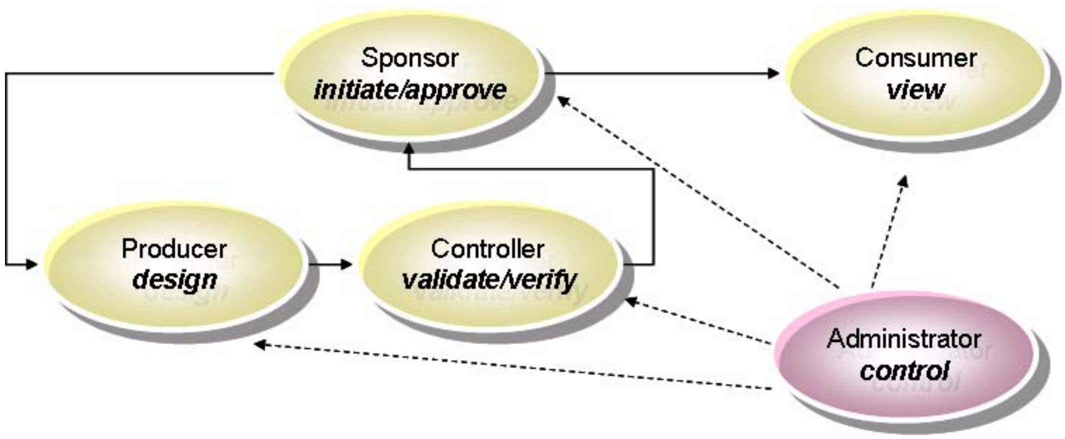


Figure 2-3: KU process – organizational perspective: actors and their responsibilities

The functional perspective considers how a process is decomposed, i.e. what *activities* are to be executed. Activities can be either *atomic,* or *composite*, which are recursively refined to atomic ones. Functionality of an activity is defined by its name, which uniquely identifies the goal of the activity.

The functional perspective of the KU process: it concerns all the activities performed by the actors described in the previous section. From a summarized perspective, the functionality required for the KU process involves the following activities:

- Use of ontologies (addition, deletion, change, verification, validation, etc.)
- Use of knowledge products (creation, structuring, deletion, verification, validation, status tracking, approval, etc.)
- Use of requirements model reports (creation, change, deletion of KM3/5W reports)
- Search for the products stored in the repository (knowledge products, ontologies, etc.)
- User-view generation (for the knowledge products, ontologies, etc.)

- Authentication management
- Access control management

The outlined activities will be explored in detail in Section 2.3, when defining the KU processes from the organizational perspective, that is, for each of the defined actors separately.

The informational perspective concerns the *resources* manipulated in a process. A resource can be either *physical* such as a service, or *informational* such as an artefact. A resource is consumed or produced by an atomic process activity.

The informational perspective of the KU Process: the resources managed during the KU process are of the informational type. In the DCMF Repository context, they are called *products*. They are produced and consumed by the actors listed previously in this section. The DCMF products include Knowledge Products (Knowledge Instances, Knowledge Components and Conceptual Models), Ontologies, Knowledge Reports (KM3 and 5W), and Metadata (for all the listed products). From the perspective of the KU Process, it is of vital importance to define which are the informational resources (i.e. products) used, by which activities and which actors.

The behavioural perspective concerns the flow of data and activities within a process. The *data flow* describes the flow of information resources from one activity to another. The second flow aspect describes the *control flow*, i.e. when an activity is to be executed in relation to others. For specification of coordination rules among activities, process specifications rely on three basic control flow constructs: sequence, parallel execution and conditional branching. The basic and the advanced control flows, such as synchronization, looping and advance branching, make possible the specifying of decisions made according to certain *business rules*.

Behavioural perspective of the KU Process: The control flow of the KU process is quite simple, i.e., it may involve the basic control flow patterns, such as sequence, conditional and parallel flow**,** applied by each of the KU actors when performing their separated and isolated processes. A few more advanced control flow patterns will be needed when the KU is considered in a concurrent-user environment, which is the subject of a future work.

The data flow is a very important consideration of the KU process, as well as of the whole DCMF process. From the KU process perspective, it is necessary to elicit how the informational resources (DCMF products) are used (i.e. processed), to get a sufficient understanding of information transformation in the DCMF context.

Business rules, i.e. decisions on how particular DCMF products will be used in the KU process, are determined in the DCMF process as a whole [Mojtahed et al, 05], as well as in respect to different actors as described previously in this section.

2.2.3 Process Modelling Notation

In this report, we use the Business Process Modelling Notation [White, 04] to express the KU process models. The BPMN is capable to visually model a process in a simple but semantically rich way, which might further be converted to an executable process language, such as BPEL4WS [BEA et al, 03]. In Figure 2-4, we elicit the presentation of the process modelling perspectives (organisational, functional, informational and behavioural), as supported by the BPMN. The presentation will be used in the further sections when presenting the four modelling perspectives of the KU process.

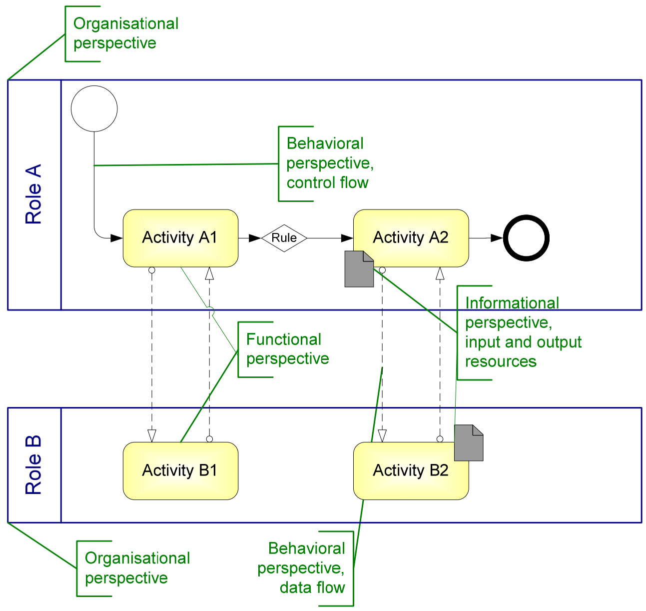


Figure 2-4: The process modelling perspectives captured by a BPMN model

2.3 Exploring the KU Phase in Details

In this section, we describe the KU process in detail, while focussing on the four different perspectives of the process modelling framework described in the previous section. As explained earlier, from the DCMF process perspective, the Knowledge Use phase is primarily concerned with the end-user and his use of the information acquired in the DCMF repository. For that reason, in what follows, we will describe the usages of the repository information, from the organizational perspective, that is, by explaining the process of every actor outlined in Section 2.2.2.

2.3.1 Process of Sponsor

The sponsor, as stated earlier, is responsible for initiating modelling of a military case and for approving the registration of the knowledge products modelling the case. Following an approval, the product becomes accessible to consumers. If a knowledge product is not entirely approved, it must be returned to the Producer for modifications.

In Figure 2-5, we depict the KU process of the Sponsor role in detail. After the Sponsor having been successfully authenticated, he either initiates a new case, or requires the list of non-accredited knowledge products from the repository of an existing case. If the list is not empty, Sponsor will consider the contained products for approvals (that is, KI, KC, or CM). After examining a product, the Sponsor will return the product as approved or non-approved, and the product status will be changed accordingly. If the product is approved it will become visible to the Consumer. Otherwise, it must be re-designed by the Producer. Later on, the Sponsor may examine another product in the list, or, if the list is exhausted, the process will be completed.

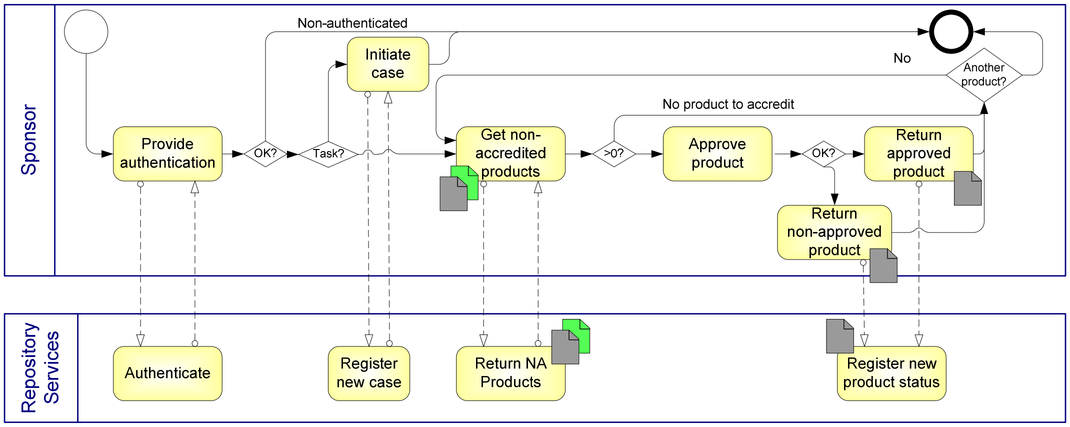


Figure 2-5: Process of the Sponsor

2.3.2 Process of the Controller

The controller has the responsibility to analyze, and thereby, to Verify, Validate and Accredit (VV&A) the products stored in the DCMF repository (recall Section 2.2.2 for details). From the functional perspective, the duty of the Controller primarily includes the assessments of ontologies and knowledge products. After a product has been verified, validated and accredited, it is given to the Sponsor for approval.

In Figure 2-6, we depict the KU process of the Controller in detail. The process starts with an authentication activity. After its successful completion, the Controller acquires a list of non-validated and non-verified products. A repository service will return a list, and if not empty, the Controller will start his task by picking a product for analysis. If the validation and verification have been successful, the Controller will set the “accredited” status in the product’s header area and return in to the repository. If the product is a knowledge product, it will become visible to the Sponsor who needs to approve its use; in the other case, it will become directly visible to the Consumer. However, if the product is not accredited, the Controller will make a note and return the product as “non-verified”, or ”non-validated”, or both, and add the comments for the improvements, thus giving a signal to the Producer that the product needs to be re-designed. Later, the Controller may analyze another object from the list, or, if the list is exhausted, the process will be completed.

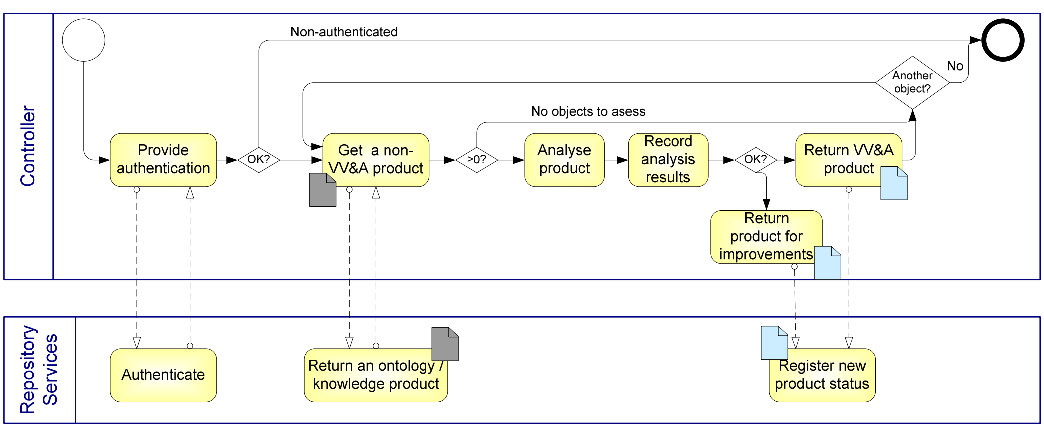


Figure 2-6: Process of the Controller

2.3.3 Process of the Producer

The responsibility of the Producer, as stated earlier, is to design the DCMF objects. Those objects are further registered in the repository and given to the Controller for an accreditation. Thereby, from the functional perspective, the duty of the Producer is to supply the repository with all the DCMF objects.

Figure 2-7 depicts the main activities of the Producer's KU process. As for the other roles, the process starts with the authentication. After its successful completion, the Producer may choose among three main tasks: a) submit an ontology to the repository; b) start a new KA phase, by submitting a KM3/5W report to the repository; and c) retrieve one or more products from the repository for further processing. In the latter case, the Producer may decide only to view the retrieved products(s), or to process (change) them further. The Producer may decide to simply update an ontology or a KM3/5W model; or deleting them. As for the knowledge products, the Producer may create one, such as a KI, a clustered KI, an aggregated KI, or a KC by doing a generalization of one or more KI. The producer may also decide to create a CM from the retrieved KCs. In addition to creation, the Producer may improve (update) a knowledge product, or delete it. The changed product is returned to the repository and its status is updated accordingly. Then, the Producer may choose to process another product, or to complete the design process.

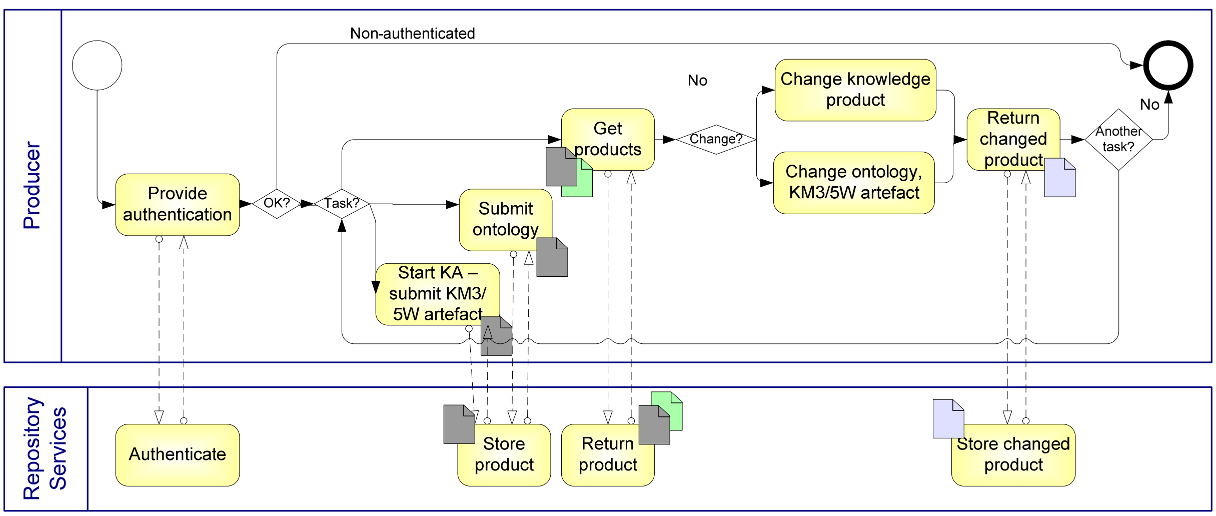


Figure 2-7: Process of the Producer

2.3.4 Process of the Consumer

The main interest of the Consumer is to view, that is, have access to the knowledge products. In addition, the Consumer may be interested in viewing the data associated with knowledge products, such as header information, ontologies, history, and so forth. The consumer may view a product following the Sponsor's approval.

In Figure 2-8, the process of the Consumer is shown in detail. The Consumer starts using the repository data by authenticating himself. After a successful completion of that activity, the Consumer searches for the knowledge product of his interest. The product may be searched by using a unique identification, or browsed by a given catalogue. A repository service will return the required product in a desired view (UML-based, or some other). Further, if the Consumer is interested in seeing some data associated with the obtained product, such as a corresponding ontology, header/history information, etc. he may acquire this information from the repository. Then, the Consumer may complete his “knowledge use” process, or may acquire to view another knowledge product from the repository.

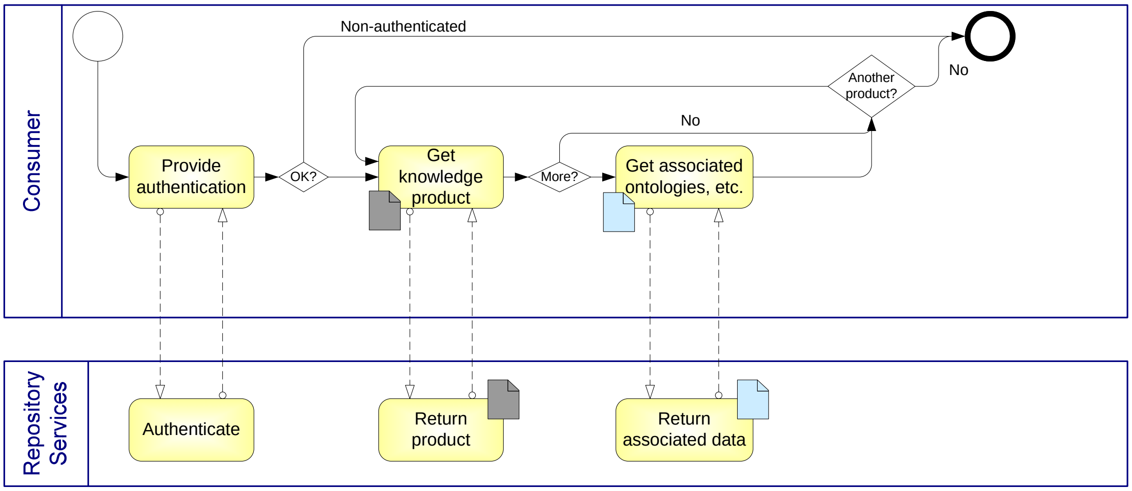


Figure 2-8: Process of the Consumer

2.3.5 Process of the Administrator

The Administrator does not participate in the use of DCMF products. His responsibility is to dedicate and control the access to the repository objects for the other roles – Producer, Controller, Sponsor and Consumer. Thus, the main task of the administrator is to create and maintain product access control lists (ACL).

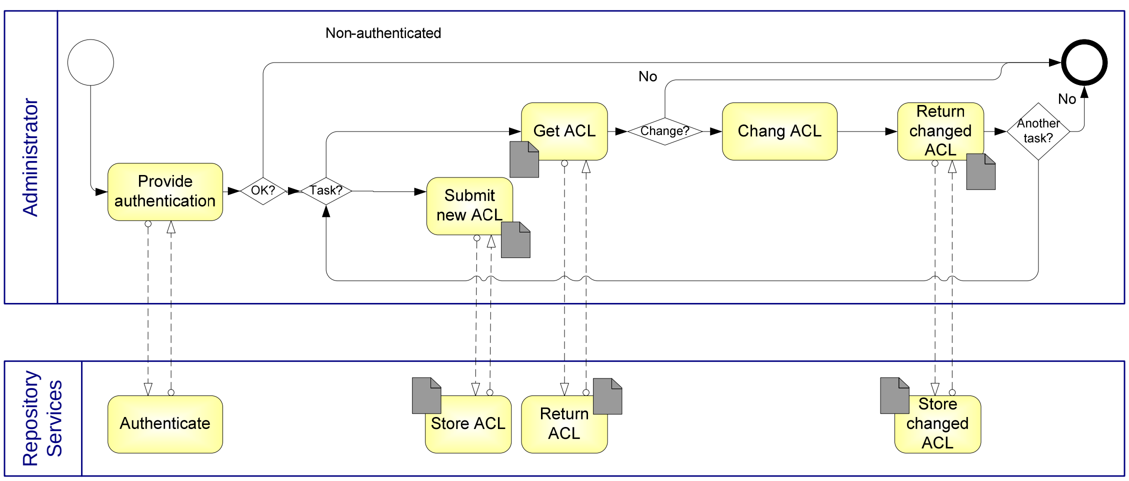


Figure 2-9: Process of Administrator

In Figure 2-9, we illustrate the KU process of the Administrator. As may be seen from the figure, after the authentication, the administrator may choose to create a new access control list or to maintain (change) an existing one. In the first case, after creating a list for a product having a particular ID, the Administrator will submit the list to the repository; in the latter case, the Administrator decides to change an existing product access control list. After retrieving the list from the repository, the Administrator may read, update or delete the list. If a change is made, an updated version of the list is returned to the repository. Then, the Administrator may choose to process another access list, or to complete the process.

As described in [Zdravkovic et al, 05], a process orchestrates a number of activities, which are further realized using concrete services. Aligned with this, the process activities, as presented in this section, within the role “Repository Services” are implemented using the actual repository services specified in Section 4. The informational aspect, i.e. the content of the informational resources (such as knowledge instances, knowledge components and conceptual models) briefly presented in this section, is explored in detail in Section 3.

3 DCMF Products

3.1 Introduction

The DCMF process is an iterative process, in which some parts and steps sometimes are conducted in parallel. By the end of every phase there is one or more products generated which may be used as input for the next or a following phase. Depending on the purpose of acquiring knowledge, some products may be ready for use after perhaps only one or two phases. Figure 3-1 shows the output of all the phases of DCMF process. [Yi et al, 2006]

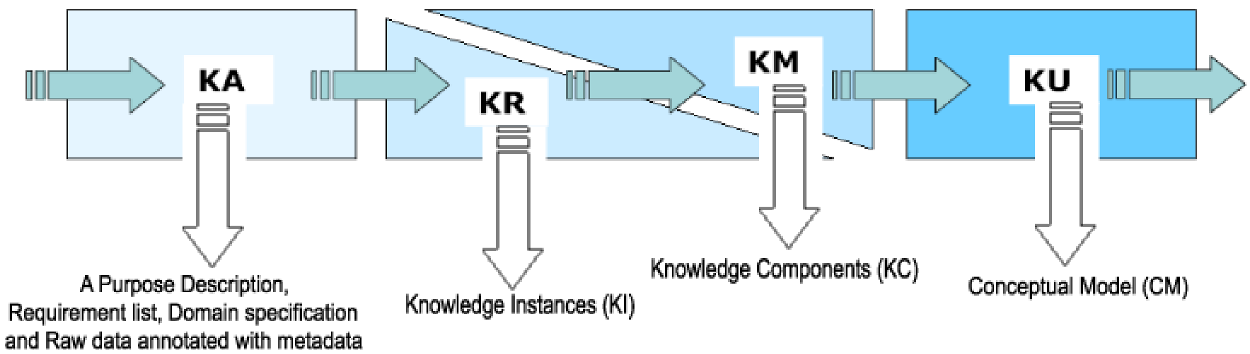


Figure 3-1: The four phases of DCMF process and their outputs.

Data and information is first gathered by knowledge acquisition methodologies. These methodologies and corresponding techniques help to ensure that the right and necessary data is gathered and so performed from the correct sources. When data in a raw text is obtained (all other formats of data; voice, video, etc are also accepted but will be converted to raw text) it is assumed to be unstructured and the first step is thus to analyse and structure it for further use. The data is therefore processed by knowledge analysis and formalisation methodologies employing the appropriate tools. By using these tools, the data is structured and focused according to some world view (Ontology). Smaller sections of this structured data are now 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* (e.g. scenario data). In order to get reusable components, modelling tools and methodologies must be applied. To make sure that no information is misinterpreted the ontologies are necessary. Modelling tools are applied to the knowledge instances in order to get more abstract and reusable components. These components are called *knowledge components* and can with the aid of more methodologies and tools, be composed to form one or more *Conceptual models*. All of these products (Scenario Data, Knowledge Instances, Knowledge Components, Conceptual Models) can and should be stored in some DCMF repository for future use together with metadata specifying 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. Table 3-1 describes input and output of DCMF in more detail. [Mojtahed et al, 2005]

Process Step	Primary Stakeholders	Input	Processing Steps	Output
Knowledge Acquisition	Authorizations Agency, Knowledge Engineer, SME	Initiate/Null	Define Purpose of CM	**A Purpose Description**
		A Purpose Description	Define Need of Knowledge	**Requirement list, Domain specification**
		Requirement list and domain specification	Search for required conceptual model in repository and Identify what is missing.	A register of what kind of information is missing and a list of authorized knowledge sources
		A register of what kind of information is missing and a list of authorized knowledge sources	Gather required information	**Raw data annotated with metadata**
Knowledge Representation	Analysis and formatting expert, Ontology expert, VV&A-agent	Raw data annotated with metadata	Analyze and break down the gathered information	**Knowledge Instances (KI)**
Knowledge Modelling	Analysis and formatting expert, Ontology expert, VV&A-agent	Knowledge Instances (KI)	Process the KIs and map them to domain ontology. KIs will be clustered and aggregated where possible.	Clustered and Aggregated Information
		Clustered and Aggregated Information	Generalize the KIs (clustered and/or aggregated information)	**Knowledge Components (KC)**
		Knowledge Components (KC)	Build the final product, CM according to the requirements.	**Conceptual Model (CM)**
Knowledge Usability	Sponsor, Producers, Consumer and Controller	Conceptual Model (CM)	Create or update the metadata of CM. Deliver the final CM adapted to the end user	**Final CM adapted for the end user**

Table 3-1: DCMF Process and Products

3.2 Knowledge Component Template

The output of Knowledge Representation (KR) phase of DCMF process will be a Knowledge Instances (KI). In these instances knowledge has been divided and categorized in a formal manner by using some analysis methodology (e.g. KM3 and 5Ws). These KI will now serve as basis for a more general and reusable Knowledge Components (KC). They will therefore be taken as an input for Knowledge Modelling (KM) phase where these instances will be further converted into Knowledge Components. As mentioned in Chapter 1 “storing previously acquired knowledge for future use” is one of the purposes of the DCMF.

Knowledge represented and formalised according to the DCMF process is reusable and composable of Knowledge Components (KC). Smaller sections of the structured data are called knowledge instances. The knowledge instances are useful for some purposes, but they are not reusable since they are specific to the source data. In order to get reusable components, 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 conceptual models being the final output of DCMF Process.

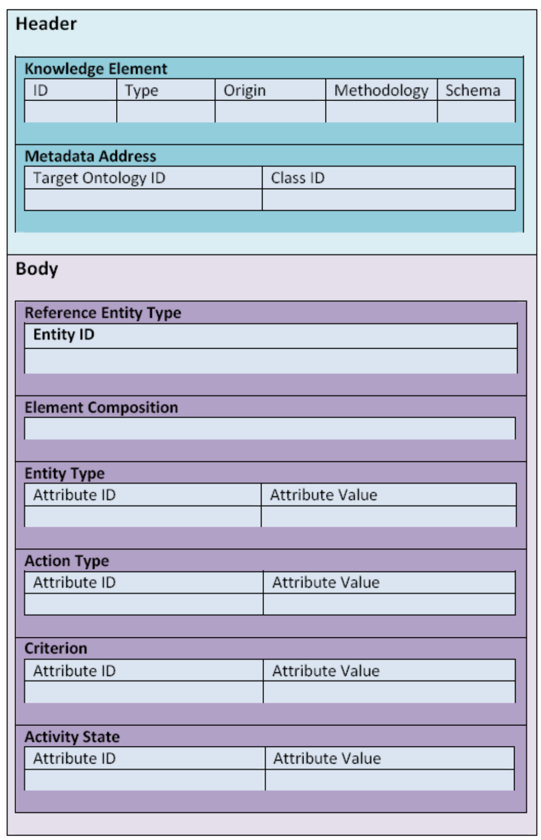


Figure 3-2: Knowledge Component Template

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 be reused in different configurations and as such producing new conceptual models. A conceptual model 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. The information captured and modelled by these KCs has to be stored somewhere in a place holder. Therefore, we need to introduce a template which is flexible enough for KC. Research for the development of a KC holder is an open issue. After having analysed some methodologies we suggest the initial template for knowledge components as shown in Figure 3-2, this template being in the initial phase of maturity and any comments on its structure will be highly appreciated. A template comprises of two parts, one is header and the other is body. A header holds information about KC metadata, such as to which scenario it belongs, what will be the destination ontology as well as addressing KC to a certain class on that ontology. The body is holding the content of KC such as what entities are involved in this KC and what kind of operations they are supposed to perform. In the next section we will go through the header and the body part of this template in more detail as well as trying to explain their building blocks.

3.2.1 Header

The Header of the template consists of two regions; the Knowledge Element and the Metadata Address. The Knowledge Element part holds the reference about the source of KC, while the Metadata Address holds information about the address of destination ontology and the destination class of KC. Let us discuss in more detail about the entities involved in the composition of the Knowledge Element and Metadata Address.

Knowledge Element:

It captures meta information about the body of KC and keeps track of the source of KC as well; it comprises of the five following entities.

ID:	It contains the unique identification of the Knowledge Component, it can be a unique number or/and a combination of Origin and Type as well.
Type:	It holds information about the nature of this Knowledge Element whether this is a Knowledge Component or a Knowledge Instance.
Origin:	It keeps track of the KI history, that is, to which scenario this KI belongs, but after generalization, the KI will convert into a KC. This scenario information can then for example be used to decide in what kind of situation this KC should be employed.
Methodology:	It holds information about the methodology, by which this element information is gathered, either by KM3, 5Ws, SPO or any other methodology.
Schema:	It contains the reference to the schema according to which the body elements have been defined.

Metadata Address:

Following aggregation and generalization, every KC will become a part of some class of ontology. Metadata Address manages references of destination ontology and to which class of ontology this KC should become a part. Metadata Address comprises of the two following entities.

Target Ontology ID: It holds the reference to the destination ontology, to which this KC will be accommodated.

Class ID: It holds the reference to the destination ontology class of which this KC will become a part.

Header

Knowledge Element

ID	Type	Origin	Methodology	Schema
KI3451X	KC	SPCos9087	KM3	918I

Metadata Address

Target Ontology ID	Class ID
87A90	SP134

Body

Reference Entity Type

Entity ID
Swedish Patrol
Contingent in Cosovo

Element Composition

<Swedish patrol, Contingent in Cosovo>

Entity Type

Attribute ID	Attribute Value
Name	Janjeva
Location	Village in Cosovo
Domain	Inhabitants
StartValue	100
StopValue	1000

Action Type

Attribute ID	Attribute Value
Time	May 2002
RoleInAction	<finder, patrol>
RoleInOrgani–sationType	<patrol, ET: Swedish patrol>

Criterion

Attribute ID	Attribute Value
State	found weapons

Activity State

Attribute ID	Attribute Value
State	finding weapons

Figure 3-3: Example of Knowledge Component

3.2.2 Body

The body of the Knowledge Component maintains information about those entities involved in this element. It will define things such as how they are related to each other, in what kind of action they will be involved, at what time a specific action will be triggered and what will be the final state of that action.

Reference Entity Type: Contains the list of references to all entities which will be used in this KC.

Element Composition: Contains the relationships between all entities which are used in this component.

Entity Type: It holds the definition of the conceptual entities that are involved in this component, the definition including their different kinds of attributes and their assigned values.

Action Type: It is used for modelling real world actions. An Action Type can be decomposed into sub-ActionTypes. ActionTypes can be sequenced or ordered through the use of attributes in the ActionType class. An example of using attributes is estimatedTime, which is used to express an approximation of the duration of the activity.

Criterion: It is the start or stop conditions for an ActionType. A start criterion is a set of statements that when they are all true, starts an activity. A stop criterion is analogously defined. Every criterion is connected to a state description indicating the new state for an activity after the start or stop criterion has been evaluated to be true. A criterion may also have effects on the roles associated with the execution of the activity. An effect can either be a new multi valued attribute, a single valued attribute, or a belief. The probability value of a criterion becoming true can be used to handle 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 simply from knowing the probability of activities' criteria becoming true.

Activity state: It keeps track of the status of the current activity, whether it is completed, partially completed or activity failed.

The template of Knowledge Component is under development and certainly a subject to change. There can be more elements involved in it according to the level of description needed while creating different instances for the target ontology.

3.3 Conceptual Model Artefacts

According to [Mojtahed et al., 2005], one of the functional requirements of DCMF-O (Ontology suite with in DCMF) is that it should be able to capture both static and dynamic contents of a Conceptual Model (CM). Following a detailed analysis of this requirement we split it into three parts for better understanding.

- For capturing data centric or static part of a conceptual model we need a holder which can handle data centric content,
- For capturing action centric or dynamic content of a conceptual model we need an action centric holder,
- For compilation of both static and dynamic part so that both parts of a conceptual model remain intact with each other, we came up with the need for Metadata, along with action and data centric holders.

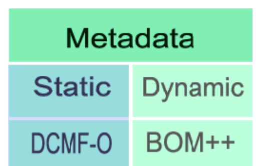


Figure 3-4: Abstract view of Conceptual Models Meta Model for DCMF

The suggested Meta Model for CM:s in DCMF is under development, and thereby the description discussed in this section is subject to future change. Let us further explain all three parts of the suggested Meta Model of CM: 1) Metadata, 2) Static and 3) Dynamic, Figure 3-4.

3.3.1 Metadata

The main role of a metadata template is to facilitate reusability. Thus, metadata provide information enabling inferences to be drawn regarding their reuse potential for supporting the extension and creation of models and simulations. It is important to include a minimum but sufficient degree of descriptive information about the CM in its metadata. An abstract view of the metadata part of the CM template is described in Figure 3-5.

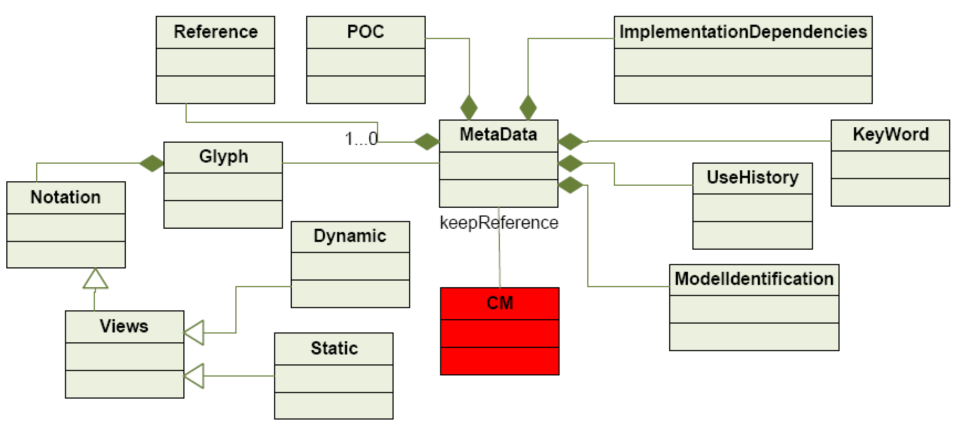


Figure 3-5: Metadata part of Conceptual Model Template.

The above given template can accommodate a number of meta features of the CM, for example: Name, Type, Version, Modification Date, Security Classification, Release Restriction, Purpose, Application Domain, Description, Use Limitation, Use History, Keyword, Implementation Dependencies, Point of Contact (POC), Reference and Glyph. We will now briefly explain which information each feature will address.

POC:	It holds information about an organization or a person having a particular role with respect to the CM.
Model Identification:	It can accommodate information related to the identification of a CM such as: Name, Type, Version, Modification Date, Security Classification, Release Restriction, Purpose, Application Domain, Description, and Use Limitation.
Use History:	It provides a description of where this CM has been used.
Reference:	It specifies a pointer to additional sources of information such as locations in XML documents and references to ontologies (both domain and middle level) which are used by the CM.
Implementation Dependencies:	It maintains a log of all dependencies determined during the development of this CM, such as domain ontologies or any other new concept introduced by the DCMF process during the implementation of this CM.
Key Word:	It holds information about the key words of this CM for future use. It helps users in searching for this CM.
Glyph:	It is responsible for holding the image of CM, which can be used to visually represent a CM in a tool palette or a web repository. *Notation, Views, Dynamic and Static:* These entities are sub-parts of *Glyph* defining the notations by which this CM should be represented to the user or stored in *Glyph*.

3.3.2 Static Content

As mentioned earlier, CM:s comprises of both static and dynamic parts. The atomic part of a CM is the knowledge component and to store this atomic part we need a holder. It has been shown that DCMF-O may fulfil the requirement of a static holder.

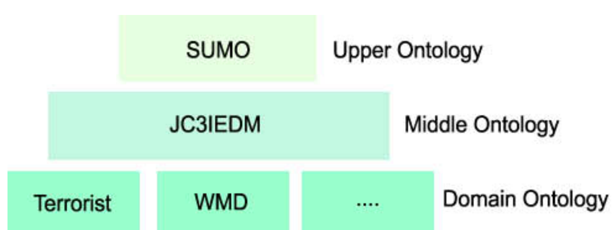


Figure 3-6: DCMF Ontology Architecture

The architecture DCMF Ontology 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.

Middle Ontology Layer: The middle layer is intended to provide a bit more specialized concepts than the top, generic SUMO level. Still, it encompasses a major domain area. In our case, this level should cover the entire range of topics included in the military operations and modelling domain. However, we have chosen to adopt the established standard JC3IEDM as a starting point for this middle level. [JC3IEDM]

Domain Ontology Layer: The bottom layer includes specific and focused domain ontologies such as the Swedish Defence Organisation structure ontology, weapons of mass destruction ontology, terrorist ontology, vessels, etc. Note that we here refer to a *collection* of ontologies rather than to a consolidated single ontology. These ontologies may be application-oriented or task-oriented.

The middle layer of DCMF-O may fulfil the requirements of static holders; it will accommodate KCs which are the atomic part of the CM. The composition of these KCs will lead us to the CM; the dynamics of the behavioural composition of KCs will be stored in a dynamic holder of the CM Meta Model. This issue will be addressed in the next section.

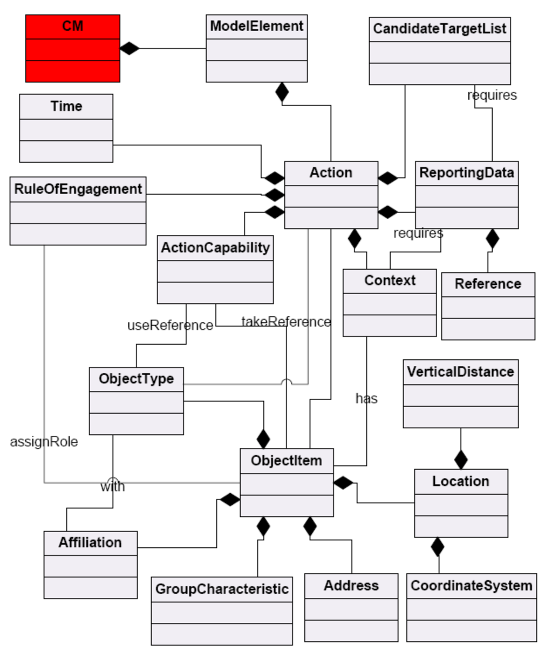


Figure 3-7: Abstract Meta Model for Static Contents of Conceptual Models

By now, we will explain the intended use or the purpose of defining principal concepts of DCMF-O, particularly the middle layer for storing the static part of a CM, given in Figure 3-7.

The interested reader is referred to the [JC3IEDM] specification for a precise technical explanation.

Action, Context and Reporting Data: The 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.*

Rule-of-Engagement and C*andidate-Target-List:* As expected in any military operation, actions are associated with certain *rules-of-engagement* and are applicable to a prescribed *cand–idate-target-list*.

Object Type and Object Item: Objects have been defined as belonging to a particular o*bject-type* or to an individual o*bject-item*. *Object-types* are generally *static* and *persistent*, whereas individual o*bject-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 an o*bject-type*, while actual fuel contained, ammunition left, current operational status of a tank are characteristics of an *object-item*. Both o*bject-type* and o*bject-item* are further classified into extensive hierarchies. (We chose to implement all the second levels of classification. Further on, we chose to model only relevant categories and sub categories. The rest will be studied in future work.).

Action Capability: 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 to be able to carry out the specified action.

Location: 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 the *object-item* concept. Examples include boundaries, corridors, restricted areas, minefields, and any other control measures needed by commanders and their staff.

Affiliation: 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.*

3.3.3 Dynamic Content

As discussed in a previous section, Knowledge Components (KCs) are the atomic parts of a CM, and the composition of those atomic KCs leads us towards a CM. The static part of the Meta Model for Conceptual Model captures KCs as independent components. To obtain a proper CM we need to create a semantic relation among KCs, and we need a holder capable of preserving this relation. After having analysed our previous research, we believe that BOM++ [Mojtahed et al, 2008] may fulfil the requirements of a dynamic holder due to its action centric nature and semantically enriched state machine.

The dynamic part of a CM will capture the activities, actions and decisions performed by atomic KCs of a CM. The improved Conceptual Model Definition of BOM, the BOM++ [Mojtahed et al., 2008] provides enhanced components for representing the behavioural needs of a simulation: Pattern of Interplay, State Machine, Event Type and Entity Type. Together, they all describe the flow and dependencies of events and their related KCs. Figure 3-8 shows the Meta Model of BOM++ which is used in the Meta Model of CM:s.

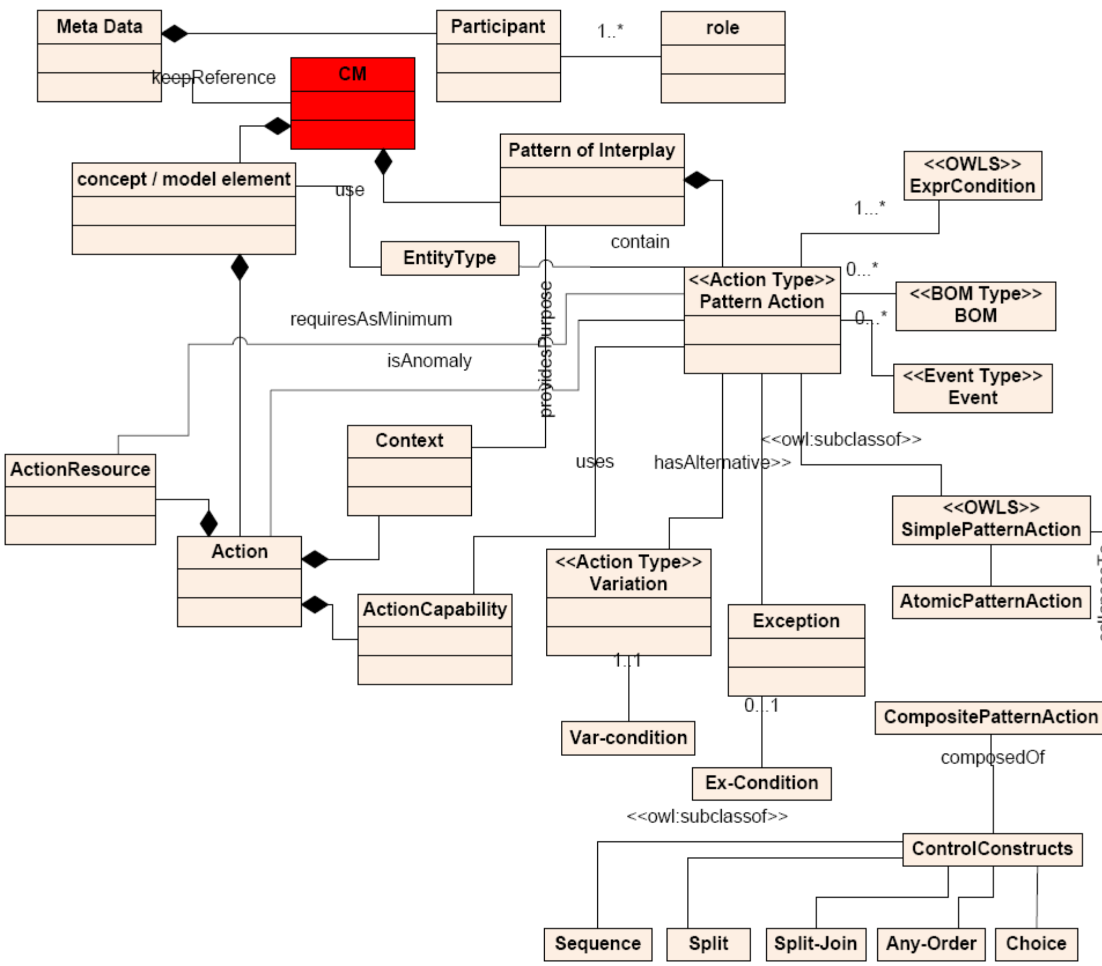


Figure 3-8: Conceptual Model Definition from BOM++ Perspective

Let us move on with the detailed understanding of the four main parts of BOM++ addressing CM behaviours. The interested reader is referred to the BOM specification [BOM] and BOM++ specification [Mojtahed et al, 2008] for a precise technical explanation.

Pattern of Interplay: A pattern of interplay is represented by one or more pattern actions needed to accomplish a specific purpose or capability. The pattern action is described by a pattern description. The pattern description provides a table of all actions taken by the component, and describes what are the conceptual entities and events taking part in the action as well as listing a number of variations and exceptions to the action possibly happening. Each pattern action has one or more senders and receivers providing a means for understanding the behavioural relationship among conceptual entities, which are defined by entity types.

State Machine: Describes the possible states of the conceptual entities as well as the transitions between them. There must be conditions specified for each transition from one state to the next. The states can also have activities which require interactions with other entities and that will be described in pattern of interplay. The state machine template component provides a mechanism for identifying the behaviour states expected to be exhibited by one or more conceptual entities. It lists all states reachable by the entities, and state transitions (what conditions must be satisfied in order to exit the state – via a specified action – into another state).

Entity Type and *Event Type:* The last two parts, e*ntity type* and e*vent type* simply identify and describe the entities and activities used in Pattern of Interplay and State Machine. More specifically, the e*ntity type* template component provides a mechanism for describing the types of conceptual entities used to represent senders and receivers identified within a Pattern of Interplay and to carry out the role of conceptual entities identified within the state machine. While the e*vent type* template component provides a mechanism for describing the types of conceptual events used to represent and carry out pattern actions, variations and exceptions are defined within a pattern of interplay.

3.3.4 Suggested Meta Model for Conceptual Models

By combining our three abstract models (Metadata (Section 3.3.1), Static (Section 3.3.2) and Dynamic (Section 3.3.3)) into one holder we create the Meta Model for Conceptual Models that can accommodate the metadata, static and dynamic contents of a CM.

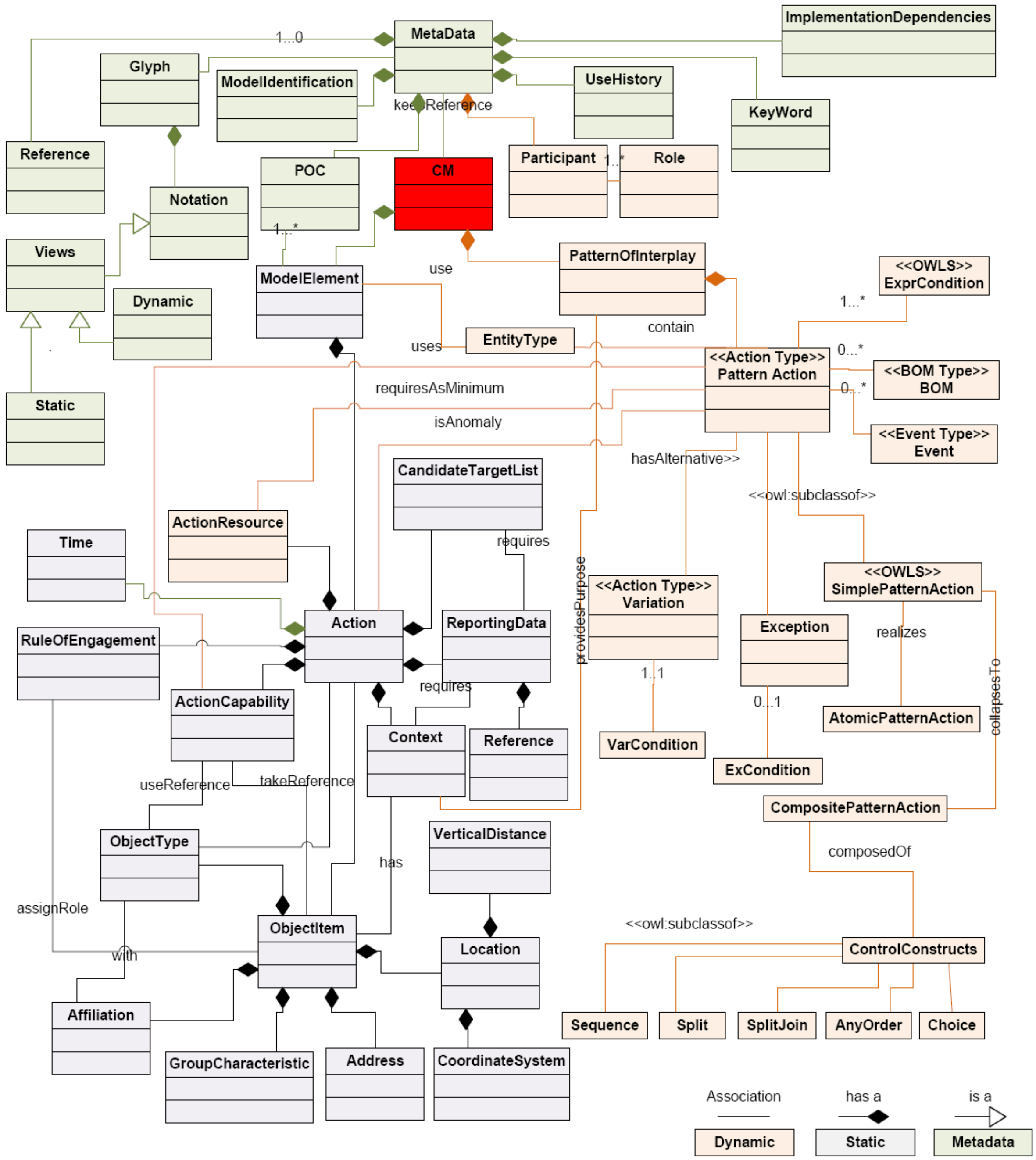


Figure 3-9: Abstract Meta Model for Conceptual Models

This suggested Meta Model for Conceptual Models is at its initial stage of development and it may need a lot of iterations for improvement. Any suggestions regarding improvements of this suggested Meta Model for Conceptual Model will be highly appreciated. An abstract view of Meta Model for Conceptual Models is presented in Figure 3-9.

4 The DCMF Repository

In Chapter 2 we showed how the different roles interact with the repository level services and in Chapter 3 we introduced the core data layout for the most important knowledge products within the DCMF. The structural layout of the repository architecture has yet to be discussed, which is what this chapter will focus on.

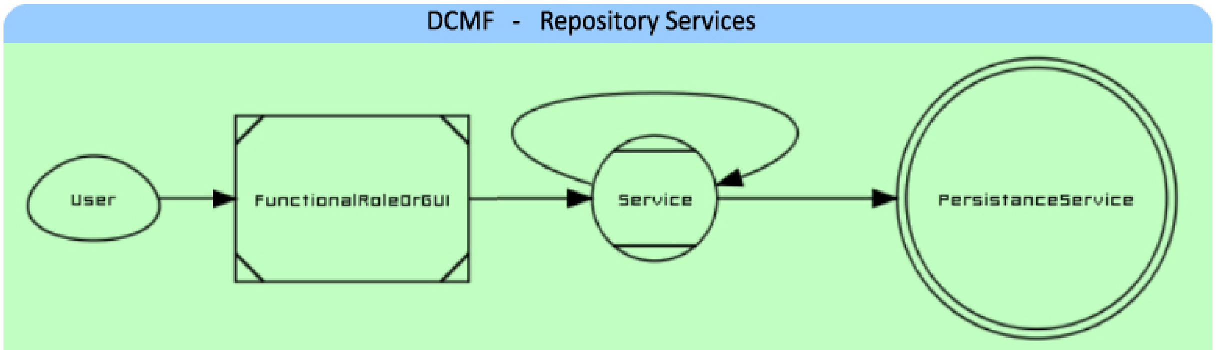


Figure 4-1: DCMF Repository Service Architecture

The DCMF repository is structured from the perspective of user roles and their requirements. Since we are, in this part, more interested in the investigation of the repository level structure and not specifically the processes of roles, we restrict our reasoning to orthogonal functional interfaces. The different ways in which different roles will interact with the repository is limited by the use of their aggregated set of different functional interfaces. These functional interfaces will in their turn interact with corresponding services.

4.1 Assumptions

The service architecture revolves around the use of different products. These products have their own set of descriptions and data but this information is in many cases non-consistent with the surrounding infrastructure. The DCMF as a whole also carries different roles within classes of users which need to rely on secure information exchange.

Identification

All data and users are identifiable through the use of an identifier, not necessarily of the same type. Without this kind of information it would not be possible to establish roles or derive from which data and processes different products have evolved.

We will assume that any user interaction as well as an authorization for any requested functionality is preceded by an authenticating action. The system will always refer to roles and their rights and never to a specific individual. It is up to the administrator to assign real users to different roles.

Cascaded service invocation, or delegation, is solved through the use of a token which is provided at the time of a successful authentication. The token itself is just a short lived certificate for which it can be established that the DCMF certificate authority was the root issuer. Any message then has this token appended to it.

Knowledge Products

From the DCMF Repository perspective there are four major products to keep track of:

Knowledge Instance: generally a coarse grained but structured report

Knowledge Component: a fine grained conceptualization of the report

Ontology: the description of the metadata for any product

Requirements: what defines a conceptual model

Stored products, and their evolution, are always traceable in the DCMF Repository. Internal, and additional, bookkeeping which supports these needs will be added or associated for each of the different types of products.

We believe in a document driven approach for this kind of long-term storage of knowledge products and as such do not implicitly rely on an internal format. Since we need to be able to understand and reason about how any information is structured we need to have a format descriptor available for any stored product. The format descriptor, or tag, is always retrievable for consistent products. For each format description tag there is always an internal represented scheme. This ensures that syntax evaluation can always occur seamlessly by services and functional interfaces alike. The format serialization scheme, most likely described by a format descriptor, is used for automatic syntax validation, and in some cases even semantic validation. We need to be able to ask any product for the identifier of its creator, current owner, validator, accreditator and format tag.

There are however two additional measures needed to be taken in order to ensure a ”as good as possible structure" for any knowledge product. All products, which either can not be expressed in a way a Reasoner understands, or are not structured enough to support this, must be evaluated manually by the controller. This evaluation process halts the evolution of the conceptual model until this task has been finalized. The very same is true for the sponsor role which accredits any validated and/or refined knowledge product.

Network communication

We reserve ourselves, at this point, to assume perfect message exchange. This means that messages are sent over an end to end encrypted channel leveraging on some proved and sufficiently safe algorithm.

Persistence

We define persistence actions in the terms of transactional read and write actions which cannot be decrypted by a third party unless proper rights have been granted. This also implies that data is decrypted right before actual evaluation and then encrypted right back again.

4.2 DCMF Repository Service Architecture

The repository service architecture suggested for DCMF includes three main layers: User, Functional Interface and Repository Services. Repository Services comprise of two parts High Level Services and Low Level Services. Detailed architecture of repository service suggested for DCMF is shown in Figure 4-2.

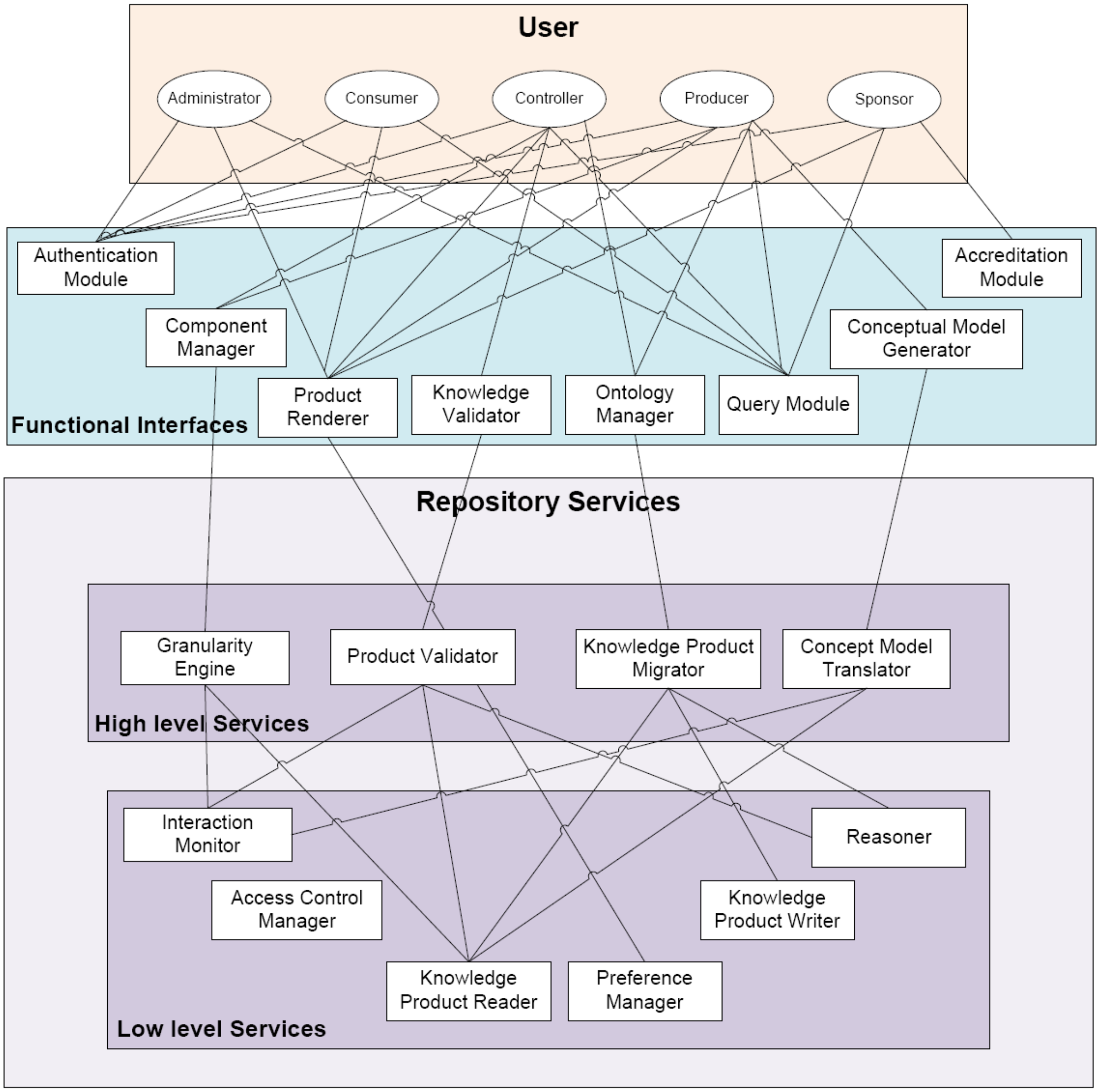


Figure 4-2: DCMF Repository Service Architecture (NB: This architecture is yet under construction and may not include all the details.)

This suggested architecture is under development and can be updated in the future. Let us go further for detail description of these three layers about what kind of components they involve, what kind of services they offer and how these layers communicate with each other.

4.3 User Roles

We have already in Chapter 2 gone through the different users roles within the DCMF but will repeat them here for brevity as well as for the sake of presenting them in another light. We are of course referring to the roles as viewed from within the DCMF and not in some implicit meaning.

Administrator has an excess of work inside the DCMF. She manages access control, i.e. specifies roles and access lists. She verifies that all services are up and running and she mends aged certificates. She is also responsible for data warehousing and migration.

Consumer is just about anyone who needs accredited information. She uses both conceptual models and knowledge components for analysis.

Controller provides essential pieces not only for data fulfilment in the form of validation of knowledge instances and knowledge components but also validates available ontologies.

Producer designs and publishes knowledge products in any of the supported formats. She is also able to create ontologies and alter metadata.

Sponsor approves any validated results for usage as well as injects requirements which are needed to fulfil some conceptual model. She can also follow the process of a conceptual model evolution.

We can now focus on the different types of layers of services and provide some simple invocation examples to make the different services function more apparent. We will not talk about things like data layout or specific user roles but instead from the perspective of a provided function.

4.4 Functional Interfaces

Functional interfaces are the contact areas between the DCMF as a system and the different roles interacting with it. Each interface should in itself perform a distinct function and still allow a developer to cluster them for different roles. There are many different names to describe this kind of user interface concepts some of which being: component, widgets and mash-up. To repeat: each functional interface encapsulates a specific and orthogonal function, with regards to the other components, that any user in the system might want to interact with. The different components are most likely to gain fidelity as the system architecture evolves.

4.4.1 Authentication Module

The authentication module is the first thing any user will see when they interact with the system. The user will provide a certificate, which needs to be signed by the authoritative service and then provide a password. This message will be sent to one of the authoritative services as a signed message digest which will in turn return a temporary token certificate.

Roles: All users in the system will need to use the authentication module to establish their user role.

4.4.2 Component Manager

The component manager is used as a form for any supported knowledge product. Since all the different products are required to provide a scheme describing their composition we can then provide mutable GUI (Graphical User Interface) generated from this description. Components can only be stored and loaded locally. When sent they will be given a unique identifier for which the current user will be set as creator. This module will perform rudimentary and/or necessary consistency checks on any data before injection.

Report Sender: The report sender is a simple version of the component manager and is used to provide a form interface to commit the knowledge instances.

Roles: Both producer and controller need to use the report sender but only controller has the full functionality of the component manager.

4.4.3 Product Renderer

The product renderer will take product data from any supported formatted data and create a view of that data. The main renderers which will be provided are likely to be different types of UML diagrams, i.e. UML Interaction, UML Sequence, and so forth.

Roles: All roles will need to see rendered knowledge.

4.4.4 Knowledge Validator

The main task of the validator is to provide a list of all non-validated products. Additionally the validator provides both a simple log window for the output from the program which runs product validation checks as well as messages with improvement suggestions attached to the product from the sponsor.

Knowledge validation is done through the combined use of the “Component Manager", the “Knowledge Validator" and the “Product Renderer".

Roles: Only the controller will validate knowledge products.

4.4.5 Ontology Manager

Any product may be semantically described by one or several ontologies. Ontology management is done through the use of the knowledge validator, since ontologies are just another product, given the proper rights.

Roles: Both the controller and producer need to use ontology management.

4.4.6 Query Module

The query module is a split interface which provides both ordinary keyword based queries search box as well as a SPARQL graph generator [SPARQL] (where SPARQL stands for Simple Protocol and RDF Query Language, and RDF stands for Resource Description Framework) [RDF]. Complementary functionality of the module is to list the collection of identifiers of the last search.

Roles: All roles will need to search for product identifiers.

4.4.7 Conceptual Model Generator

Model generation will be performed by specifying specific run-time behaviours, in the same language used by the requirements for the conceptual model. This specification will then be sent to the system for construction.

Roles: Only the producer will need to generate conceptual models.

4.4.8 Accreditation Module

The accreditation module is able to inject, initial or additional, requirements for any conceptual model. The user may also follow any conceptual model status of completeness, i.e. the DCMF process for that model.

Roles: Only the sponsor is able to accredit products.

4.5 The Repository Services

The DCMF repository architecture is mainly built by composing two classes of services; the high level services and the low level ones. The high level services provide some clustered or orchestrated functionality and are the main areas of contact for all functional interfaces. The services in this class presumably have a high computational intensity. We will briefly mention all those services we are aware of at present.

4.5.1 High Level Services

Granularity Engine: The GE is responsible for report decomposition, it is possible, or likely, that the different report types contains clustered information. Thus this is where data refinement of reports is done.

Functional Interfaces: Only the “Report Sender" uses the granularity engine.

Low Level Services: The GE talks directly to low level services and uses only ”Knowledge Product Writer" and the “Interaction Monitor".

Product Validator: The product validator validates any product which is not in a consistent state but also tries to find any irrationality in the descriptions.

Functional Interfaces: Only the knowledge validator uses the product validator.

Low Level Services: The validator needs only to talk to the “Knowledge Product Reader", the “Reasoner" and the “Interaction Monitor".

Concept Model Translator: This module is the engine which takes the requirements specification and generates the conceptual model, if the specification can be fulfilled. In other cases it queues the specification as an initiated requirements list.

Functional Interfaces: Only the Conceptual Model Generator uses the concept model translator.

Low Level Services: The concept model translator only need to register incoming requests in the “Interaction Monitor" and read descriptions from the ”Knowledge Product Reader".

Knowledge Product Migrator: Migrates products between syntactic and semantic formats.

Functional Interfaces: The ”Ontology Manager" needs to switch product descriptions if it removes an ontology which other products are using, this kind of behaviour would leave products in an inconsistent state.

Low Level Services: The “Knowledge Product Migrator” transaction can be decomposed into ”Read", ”Redescribe" and ”Write" which maps to the ”Knowledge Product Writer", ”Knowledge Product Reader" and ”Reasoner".

4.5.2 Low Level Services

Low level services do small amount of data processing and read/store data in a persistent state to the same data set. These services have a low computational intensity.

Interaction Monitor: The interaction monitor logs any action taken by any of the upper level services.

Knowledge Product Writer: The product writer is responsible for the storage of products, presumably to a general format.

Knowledge Product Reader: The product reader is responsible for the retrieval of products, presumably in a requested format.

Reasoner: The reasoners main task is to evaluate queries but will also likely be able to do data transformation.

Preference Manager: Retrieves and stores any single user’s preferences.

Access Control Manager: Retrieves and stores all of the different ACLs (Access Control List), but also provides authorization query evaluation.

5 Conclusion and Future Work

We have in this study explored the Knowledge Use (KU) phase of the DCMF in detail, using three major pillars 1) *process*, in what different ways the DCMF users can gather and scope the information acquired in the DCMF repository; 2) *product*, i.e. the major information and models that can be retrieved by the DCMF users; 3) *service*, what functions should the DCMF repository provide, and how these functions should be structured in order to facilitate the use of the stored information in the form of different user-services. The purpose of the study is to contribute to a completion of the DCMF, from the aspect of the structure of the main product of the framework – the conceptual model, and the methods for its management (i.e. persistence and usage).

From a more detailed perspective, the reported study encompasses the following.

- In Chapter 2, we have described the Knowledge Use (KU) phase as a process, and from the four different design perspectives: organizational, functional, informational and behavioural. These gave us ability to model and discuss the aspects of the knowledge use distinctively, from each of the outlined perspectives. From the organizational perspective, we have identified and proposed 5 distinct KU user-roles: sponsor, producer, controller, consumer and administrator. Those roles have been further used as a basis for identifying different functions needed with respect to the repository (functional perspective). Each of the roles is responsible for managing certain information, such as models and artefacts (informational perspective), whereas each information is processed differently, that is in different flows, and with different rules (behavioural aspect). The integration of the four perspectives has enabled sketching of the processes of the proposed KU user-roles.
- In Chapter 3, we have set the focus on the DCMF models and artefacts; we have proposed two major information (i.e. knowledge) outputs: Knowledge Components and Conceptual Models. In order to be able to capture these information outputs we came up with the need for proposing some adequate holders. After a detailed analysis of the previous work on the DCMF, we have outlined a holder template for Knowledge Components; the holder is its initial phase, and in the future we believe that it will grow and become mature enough to accommodate the atomic-level knowledge products. As for a holder for the Conceptual Model, we have introduced an ontology-based Meta Model. The main property of the meta-model is that it can accommodate both static and dynamic requirements aspects; in addition, the Meta Model contains a part that enables the traceability between the different versions of a product, i.e. a back- tracking. This Meta Model of the Conceptual Model is also at its initial stage and in future, more maturity of this model is expected.
- In Chapter 4, we have outlined the architecture for the DCMF repository services, as needed in the KU phase. The architecture includes three major layers: Users, a Functional Interface containing a set of user-functions, and a set of Repository Services. Repository services comprise two classes: high-level services and low-level services. The high-level services provide some clustered or orchestrated functionality, such as provisioning of different products, and are the main areas of contact for all the functional interfaces. These services are supposed to realize the activities of the

proposed KU user-roles, to enable usage of different information, especially Knowledge Products. The low-level services are responsible for reads/stores of data in a persistent state and in an alignment with the architecture of a used repository engine.

In previous years we were more focused on KA, KR and KM. but for completion of DCMF Process life cycle we needed to focus on KU phase. This year we have done so, and we have tried to establish a track for making it possible to achieve the goals of KU phase. After having studied the KU phase in detail we came up with three meta issues which are related to repository: process, product and services. The detailed analysis of these issues during this year make us convinced that we are going in the right direction and the three major pillars introduced in this report seem very promising. So, after maturity of all these three parts we may be able to overcome the goals of KU phase. And after completion of this phase the life cycle of DCMF Process will be completed.

The reader should observe that the work presented in this report is in its initial phase. The use of the presented, multi-aspect process framework enables further exploration and improvement of the Knowledge Use phase, from each of the four perspectives. Furthermore, the alignment between the activities of the user-roles processes and the repository services need to be investigated in more detail. The structure and the content of the Knowledge Product holders is still under a research, as it is mainly influenced with target usages of these products. From a general perspective, the proposals for processes, products and services as given in this study need to be validated, and exemplified. As for further research, we plan to improve the given proposal for the main product, i.e. the Conceptual Model, to support different usages, such as an automatic generation of executable process scripts to enable the simulation and an on-place implementation of given military commands. This effort, as mentioned in the introductory section, needs to be aligned with the work of the NATO NMSG-058 task group, i.e. it needs to follow the recommended practices for conceptual modelling aimed to increase model interoperability, reusability, and the employment of simulations.

6 Acronyms

5W:s	Who, What, When, Where, Why
ACL	Access Control List
BOM	Base Object Model
BOM++	A semantically enriched BOM
BPEL4WS	Business Process Execution Language for Web Services version
BPMN	Business Process Modelling Notation
CM	Conceptual Model
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
FOI	Swedish Defence Research Agency
GE	Granularity Engine
JC3IEDM	Joint Command Control Communication Information Exchange Data Model
KA	Knowledge Acquisition
KC	Knowledge Components
KE	Knowledge Engineering
KI	Knowledge Instance
KM	Knowledge Modelling
KM3	Knowledge Meta Meta Model
KR	Knowledge Representation
KU	Knowledge Use
M&S	Modelling and Simulation
NATO	North Atlantic Treaty Organisation
NMSG	NATO Modelling and Simulation Group
POC	Point of Contact
RTO	Research and Technology Organization
SISO	Simulation Interoperability Standards Organization
SME	Subject Matter Expert
SPARQL	Simple Protocol and RDF Query Language
SUMO	Suggested Upper Merged Ontology
UML	Unified Modelling Language
VV&A	Verification, Validation & Accreditation

7 References

[05F-SIW-038] Lozano M. G., Mojtahed V.: *A Process for Developing Conceptual Models of the Mission Space (CMMS) – From Knowledge Acquisition to Knowledge Use*, 05F-SIW-038

[06E-SIW-028] Kabilan V., Mojtahed V.: *Introducing DCMF-O - Ontology Suite for Defence Conceptual Modelling*, 06E-SIW-028

[08F-SIW-038] NATO-MSG-058: *Conceptual Modeling for M&S*, 08F-SIW-038

[BEA et al, 03] BEA, IBM, Microsoft, SAP and Siebel, 2003. *Business Process Execution Language for Web Services*. In http://www.ibm.com/developerworks/library/ws-bpel, Last accessed on 2008-10-24

[BOM] Base Object Model (BOM) Template Specification. Simulation Interoperability Standards Organization (SISO). SISO-STD-003-2006.

[Curtis et al, 92] Curtis B., Kellner M., Over J.: *Process Modeling*. Communication of the ACM, Vol. 35, No.9, 1992

[Davenport et al, 92] Davenport T.: *Process Innovation:Reengineering work through information technology*, Harvard Business School, 1992

[JC3IEDM] The documentation on *JC3IEDM* is available on: http://www.mip-site.org/publicsite/04-Baseline_3.0/JC3IEDM-Joint_C3_Information_Exchange_Data_Model/, Last accessed on 2008-10-29

[Mojtahed et al, 2005] Mojtahed V., Lozano M. G., Svan P., Andersson B., Kabilan V.: *DCMF – Defence Conceptual Modelling Framework.* Methodology Report, FOI-R--1754--SE, 2005

[Mojtahed et al, 2008] Mojtahed V., Andersson B., Kabilan V.: *BOM and DCMF*, Methodology Report, FOI-R--2362--SE, 2008

[RDF] The documentation about RDF is available on: http://www.w3.org/RDF/, Last accessed on 2008-10-15

[SPARQL] The documentation on SPARQL is available on: http://www.w3.org/TR/rdf-sparql-query/, Last accessed on 2008-10-21

[White, 04] White S., 2004. *Business Process Modeling Notation 1.0*, (BPMN). In http://www.bpmi.org. Business Management

Initiative, Last accessed on 2008-10-29

[Yi et al, 2006] Yi C., Mojtahed V., Lozano M. G.: *REVVA and DCMF- A Link between VV&A and Conceptual Modelling*, FOI-R--2110--SE, 2006

[Zdravkovic et al, 05] Zdravkovic J., Henkel M., Johannesson P.: *Moving from Business to Technology using Service-based Processes,* The IEEE Journal of Internet Computing Vol. 9, Nr. 3, p.73-81, 2005