

BOM++, a Semantically Enriched BOM

Vahid Mojtahed

FOI, Swedish Defence Research Agency
Division of Command and Control Systems
Department of Decision Support Systems
SE-164 90 Stockholm, Sweden
vahid.mojtahed@foi.se

Birger Andersson, Vandana Kabilan, Jelena Zdravkovic
Department of Computer and Systems Sciences
Royal Institute of Technology and Stockholm University
FORUM 100, SE- 164 40 Kista
ba@dsv.su.se, vandana@dsv.su.se, jelenaz@dsv.su.se

Keywords:

Conceptual Modelling, Composability, Reusability, Interoperability,
DCMF, BOM, Ontology, OWL, XML, Mission Space Model

ABSTRACT: *The Defence Conceptual Modelling Framework (DCMF) is the Swedish Defence Research Agency's (FOI) proposal to deal with conceptual modelling in the military domain. The vision of the DCMF is to enable composability, interoperability and reuse of knowledge for modelling and simulation. The final products, the conceptual models, are called Mission Space Models (MSMs), following the original CMMS proposal (proposed by the US DoD). However, the representation formalisation for the MSMs is still under research. To find a suitable representation template we are currently investigating several proposals of which the Base Object Model (BOM) is one. In this paper, we introduce two different proposals for semantic enhancements of BOM for enabling representation of the MSM conceptual models.*

1. Introduction

The Defence Conceptual Modelling Framework (DCMF) is a project at the Swedish Defence Research Agency (FOI) which attempts to create a framework for capturing, analysing and representing conceptual models. Those conceptual models are formalised descriptions of real world processes, entities, environmental factors, associated relationships and interactions that constitute a particular set of military missions, operations or tasks. Such conceptual models should be generic and applicable to as many military scenarios as possible without any loss of critical information. The DCMF also includes a method for developing the conceptual models of military operations.

The end products of the DCMF, the conceptual models, are called Mission Space Models (MSMs). For all stakeholders involved in Modelling and Simulation (M&S) process, these MSMs are meant to be a common description of the content to be simulated; it also serves as

a bridge between military experts and simulation developers.

The final representation format of MSM, which should encapsulate the conceptual model, is yet not settled. There are several proposals for representation formats of the MSMs and in this paper we discuss our analysis and views regarding one such formalisation -- the Base Object Model (BOM). The BOM is a concept created by Simulation Interoperability Standards Organization (SISO) to enable composability and reuse for High Level Architecture (HLA) simulations. The BOM development started in 1997, and since 2006 it is a SISO standard.

The DCMF demands that the semantics of the produced MSMs come from relating them to an ontology. The main benefit of this is that the models are improved in terms of interoperability, reusability, and composability. When considering the BOM as a semantic base for modelling the MSMs, there was a need to find a way to enrich the BOM with more semantics than it presently carries. When

adding semantics to a BOM, one consideration is whether to strive for *an incremental change of the existing semantics, or to allow a larger, conceptual change*. The benefit of a small change is that existing implementations remain undisturbed, while the risk is that all envisioned goals of the change can not be reached. A large change, on the other side gives developers and users access to new opportunities, while the risk being that old solutions and ways of working may be endangered.

In this study we present the both outlined approaches for extending the BOM to support modelling of the MSMs. The first solution, having incremental change as its principal idea, is that the existing schema structure of the BOM is extended. A reference to an external entity, an ontology, is added. By referring to the ontology additional information can be added to the conceptual model of the BOM. The second solution bases on the ontological conceptualisation of the DCMF and tries to enrich the BOM sufficiently to capture the semantic information processed by the DCMF. The final outcome of this approach is the BOM-DCMF ontology.

The paper has the following structure. Section 2 introduces related works. In Section 3, a detailed analysis of the BOM is given, from the DCMF perspective. Sections 4 and 5 outline our two approaches for extending the BOM. Section 6 concludes the paper with a reflection on the results and future research directions.

2. Overview of related concepts

In this section we give an overview of the concepts that are in focus in our works or are related to our proposals for extending the BOM.

2.1 HLA Overview

The High Level Architecture (HLA) is an architecture for designing simulations in military domain. The HLA was developed under the leadership of the US Defence Modelling and Simulation Office (DMSO) to support reuse and interoperability across the large numbers of different types of simulations developed and maintained by the US Department of Defence [1].

A collection of models which interoperate with each other according to HLA in a simulation with a specific purpose, are called a federation. Every single model in the federation is called a federate or a federation member. Federation Development and Execution Process (FEDEP) Model defines a generic, common sense systems engineering methodology for developing HLA federations which needs to be tailored to meet the needs of individual applications [2]. At the top level the FEDEP has a

sequence of six basic steps for developing HLA federations: 1) Define Federation Objectives, 2) Develop Federation Conceptual Model, 3) Design Federation, 4) Develop Federation, 5) Integrate and Test Federation, and finally 6) Execute Federation and Prepare Results. BOM and DCMF are mapped to the FEDEP step 2, "Developing Federation Conceptual Model" in the simulation development process.

2.2 BOM Overview

The BOM (Base Object Model) is a concept created by Simulation Interoperability Standards Organization (SISO) [3] to enable composability for HLA simulations. The open standardization of BOM representations is considered essential for encouraging its development, distribution and use. The BOM concept is based on the assumption that piece-parts of simulations and federations can be extracted and reused as modelling building blocks or components in different contexts and simulations.

A BOM contains the necessary elements both to create the conceptual model and to specify and document the interface for a simulation model. BOMs are specifically identified in the FEDEP as a potential facilitator for providing reusable object model components used for the rapid construction and modification of federates and federations.

The most interesting element from a DCMF-perspective is the Conceptual Model which deals with the knowledge while the other elements have a more administrative focus. The Conceptual Model consists itself of four parts:

- *Pattern of Interplay*, describing the interactions between different entities and the activities involved in the interactions.
- *State Machine*, describing the states that the conceptual entities can have as well as the transitions between these states.
- *Entity Type* and *Event Type*, identifying and describing the entities and activities used in Pattern of Interplay and State Machine.

2.3 DCMF Overview

The increasing use of modelling and simulation in the military domain puts high demands on how knowledge is managed and used. Major challenges are how to acquire, validate and maintain knowledge and how to achieve this with a minimum of effort. To address issues relating to knowledge bases for modelling and simulation, the US DoD introduced in 1995 a concept called Conceptual Models of the Mission Space (CMMS).

It was discovered that some parts of the specifications of the CMMS concept were vague and unfinished, and a lot

of the necessary components, methodologies and tools to finish the process, were also missing. The Swedish Defence Research Agency suggested a long-term plan in order to successfully implement the concept within the Swedish Armed Forces and called the project DCMF, Defence Conceptual Modelling Framework. DCMF is thus a framework for the development of conceptual models in military context and it captures the characteristics of objects in a domain given e.g. by standard operating procedure or a scenario, such as their features, interactions and behaviour. The framework consists of a number of components which together support developers in the task of creating high quality conceptual models out of unstructured data. Examples of components are a specified modelling process, ontologies and specifications of the format of the models.

3. Analysis of the BOM from the DCMF Perspective

As mentioned earlier, the most interesting part of the BOM concept from a DCMF perspective is its conceptual model. That is almost the only part which handles the knowledge while the other parts more or less focus on administrative activities. Therefore we have constrained and focused our study of the BOM to only the conceptual model part of it.

Below, we briefly summarize the requirements on the knowledge storage or representation model (a Mission Space Model or MSM) that we have followed in the DCMF project. The knowledge representation should:

- be able to handle unstructured information as input sources,
- be flexible and adaptable,
- facilitate easy reuse and sharing between similar systems,
- support semantic interoperability,
- be easy to develop and fill by the knowledge engineer as well as the simulation modeller,
- support both formal and informal views of the same knowledge,
- be verifiable and authenticable,
- cover static as well as dynamic procedural aspects of the domain,
- result in a consistent model no matter which domain expert carries out the analysis and modelling of the same knowledge,
- support composability of bigger simulations by reusing individual MSM components.

BOM fulfils some of the requirements imposed by DCMF, such as reusability, sharability, composability and syntactic interoperability. However, it does not support other requirements as mentioned above. BOM is not primarily knowledge representation formalism but a wrapper for encapsulating metadata and key information about other knowledge components. As such we found it insufficient to support our needs. However, we would like to benefit from the both proposals, that is, *the advantages* from a standard specification like BOM, and complement it with *the semantic richness* of an ontology like our DCMF-O (DCMF - Ontology).

Let us now take a closer look at similarities and differences between DCMF and BOM from some different perspective:

- *Common Domain and Purposes:* Both DCMF and BOM have the common goal of representing conceptual models of the military domain. Mission Space Models, MSMs, are the end products in DCMF and correspond to a BOM assembly (collection of BOMs). An MSM can, depending on the application or end user, exist in different formats where BOM could be one of these formats.
- *Different Focus in the Knowledge Management Process:* The DCMF and the BOM have different focus in the Knowledge Management Process. While the DCMF concentrates on the early phases such as acquiring, analysing, representing and formalising the knowledge, the focus of the BOM is on the composability of knowledge components in order to create a conceptual model.
- *Differences in Strategy:* the DCMF focuses on producing verifiable, authenticated and traceable knowledge components, that is, to produce conceptual models that can be traced back to its source of information. The BOM does not track how the knowledge is acquired and introduced to the BOM format, and therefore does not include a defined process for developing conceptual models. This could be one of the enrichments supplied from the DCMF.
- *Syntactic versus Semantic Interoperability:* The BOM structure supports syntactic descriptions of a conceptual model while the DCMF (i.e. DCMF-O) also supports the semantics of a conceptual model. Therefore conceptual models developed by the DCMF can be semantically interoperable. With the DCMF-O, a conceptual model can be expressed in OWL (Web Ontology Language) while the BOM is constrained to only XML (eXtensible Markup Language).

Following this outline, we want to identify and define requirements for a more expressive BOM, i.e., a BOM++. This aims to enable a BOM for a higher level of interoperability. Today the BOM is expressed with XML which is a good foundation to guarantee interoperability at the syntax level. If the BOM could be expressed with OWL, using ontologies, it would also support the semantic-level interoperability. Automatic composition of BOM Assembly is not possible today. The semantic richness and formal representation of knowledge instances in DCMF can make this possible. The current BOM is too general which can cause ambiguity and a risk for interpretation problems when composing BOMs. More semantic expressivity will decrease those risks and give the opportunity to more detailed descriptions, more specialized BOMs.

As mentioned in the introduction section, one consideration when adding semantics to a BOM is whether to strive for an incremental change of the existing structure, or to allow a larger change. In this study we define and present the both levels of the BOM extension:

- The principal idea behind Approach 1 (Section 4) is that the existing schema structure of the BOM is extended. A reference to an external entity, an ontology, is added. By referring to the ontology it becomes possible to add additional information to the conceptual model of the BOM.
- Approach 2 (Section 5) builds on the ontological conceptualization that we have adopted and worked with in the DCMF project. Our DCMF supports the action-centric analysis of military scenarios and also uses an action centric data model as its core for the ontology. Given that we already had an established work in the military scenario ontology domain and we had some ideas on how behavior and rules should be described, we set effort to extend the BOM into an DCMF-BOM ontology.

4. Approach 1: Incremental extension of the BOM

This approach concerns the opportunity to enrich or extend the current specification of the BOM with relevant parts of the DCMF process. We focus here on two aspects: a) the level of quality and b) improved reusability.

The level of quality addresses the process by which a BOM is produced; the reusability addresses the way by which a BOM is related to an ontology. The goal is to enhance the BOM specification, and to propose incremental extensions of the specification. That is, the

level of change should be kept to a minimum. The main reason for this is that the BOM specification has matured and that simulation components based on the BOM functions well within HLA supporting simulation environments. To leverage this we aim at avoiding any large disruptions.

It must be emphasized that the following is a proposal for enriching and extending the BOM. It is not a runnable solution. The main purpose for it is to discuss possible shortcomings of the current specification and argue for that the proposed directions are useful.

4.1. Level of Quality

The BOM standard documentation does not include a discussion on how to proceed to make certain that the BOMs produced can be of an assessable level of quality. In contrast, the DCMF process is explicitly designed to produce reusable, interoperable and composable conceptual models of the mission space with a certified level of quality. Therefore, we propose that the DCMF process can be used when producing BOMs. The BOM standard specifies in great detail such important information as how component interfaces can be designed and the data types that can be used. This makes it possible to validate a BOM against the specification. However, this proposal addresses the issue of ensuring that the information in a particular BOM is valid (or at least with assessable uncertainty) if this is requested. As the BOM makes use of the eXtensible Markup Language (XML) we propose to include a XML tag in its Meta-data section stating the origin of the BOM and by which process it is produced.

The BOM Template Specification has a table format for declaring which meta-data that should be included in a BOM. To this table we propose to add the category ProductionProcess as seen in Table 1. The “<!-- snip -->” construct indicates that information is omitted here.

Category	Description
<!-- snip -->	...
Purpose	This field shall specify the purpose for which the BOM was developed
ProductionProcess	This field shall specify the formal process by which the BOM was designed (e.g., DCMF)
<!-- snip -->	...

Table 1: Extended Model Identification

4.2. Improved Reusability

An important tool to improve reuse and interoperability of models is to establish what concepts exist in a universe of discourse and how they are related. The models are then

expressed in terms of those established concepts. A name for this kind of tool is 'ontology'. In the DCMF process it was discovered that an ontology was needed in order to produce the conceptual models of the mission space and a methodology for making a multi-layered ontology of the military domain was made and tested. The particular ontology is described in the DCMF proposal [4] and it is based on MIP's JC3IEDM [5].

The BOM standard documentation does not include the concept of ontology. It only explains the main conceptual model components: Entity Type, State Machine, Event Type, and Pattern of Interplay. The entity type component describes the types of conceptual entities that represent senders and receivers of information in a pattern of interplay. The state machine component identifies the behaviour states expected to be exhibited by one or more conceptual entities. The event type component describes the types of events that is the result of, or triggered from, actions, relevant in a pattern of interplay. There are two types of events, triggers and messages. Finally, the pattern of interplay component identifies a sequence of actions (including variations and exceptions) in a BOM.

For Approach 1, as explained earlier, we apply a mechanism by which we can extend and enrich the BOM by referring to an external entity - an ontology. In order to do this we have considered a similar situation in another XML-based language, called XPDL [6], which is used for modelling business processes.

Extensions in the XPDL

The Workflow Management Coalition (WfMC), the owner of the XPDL specification, acknowledges that the constructs supplied in XPDL are enough to model all aspects of a specific workflow. Therefore, they have included the two ways for how to extend the XPDL. Both ways involve the concept of "Extended Attributes". This concept can be used in all entities (e.g., activities, transitions, etc.) in the XPDL to allow e.g. tool vendors to extend the functionality of the specification to meet individual needs. There are two basic ways to extend the XPDL: through *anonymous attributes* or through change of the *namespace extensions*:

- *Anonymous Extended Attribute* provides a name and a value for the extension without the need to qualify the extension with a namespace.
- *Namespace Qualified Extensions* can be used to extend XPDL elements by adding child elements or by adding attributes. The extensions can be validated against a vendor provided schema. In order for a vendor to add namespace-qualified extensions, it first needs to extend

the XPDL schema to add the extensions in the places in which the XPDL schema allows the tool to add extensions. The new generated schema should contain, in addition to the XPDL namespace, the tool namespace. The extension places are marked in the XPDL schema by the construct: "[namespace="##other" processContents="lax"]".

Our proposal on extending the BOM is similar to the Namespace Qualified Extension.

Extending the BOM

Using the information gathered from the XPDL regarding extending mechanisms and with the constraint that we should minimally disrupt the current BOM, we here propose some extensions to the BOM specification. The particular extension concerns the Entity Type of BOM's conceptual model. However, the mechanism for extension is more general and can be applied to the other components of the conceptual model as well.

The extension is quite simple. The original structure of an Entity Type is that it has a name and that it aggregates one or several characteristics, each having its own name. The extension should be thought of as the Entity Type become able to aggregate some additional structures that take their names from an external source. Listing 1 is a commented piece of XML schema code, where the extensions are explained in place close to where they are applied. Listing 2 is an example of an instance of the schema.

```
// We define a namespace alias and import a schema. Note
// that the names used are fictive. The code below is to
// illustrate the main ideas.

<xs:schema xmlns="http://www.sisostds.org/schemas/bom"
  xmlns:omt="http://www.sisostds.org/schemas/IEEE1516.2-
  2006"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:modelID="http://www.sisostds.org/schema/modelID"
  xmlns:dcmf="http://www.foi.mil/schemas/DCMF"
  targetNamespace="http://www.sisostds.org/schemas/bom"
  elementFormDefault="qualified">
  <xs:import
    namespace="http://www.sisostds.org/schemas/IEEE1516.2-
    2006"
    schemaLocation="IEEE1516.2-2006-D2v0.82.xsd"/>
  <xs:import
    namespace="http://www.sisostds.org/schemas/modelID"
    schemaLocation="ModelID_v2006.xsd"/>
  <xs:import
    namespace="http://www.foi.mil/schema/DCMF"
    schemaLocation="DCMF_v2007.xsd"/>
```

```

// We are only interested in the extension of the EntityType
part. The removed code is replaced by a "snip" comment.
We add, to the original entityType, the option to include one
relationType from the DCMF schema, and an option to
include an unlimited amount of Entities from the DCMF
schema. Thus, on top we have the BOM, then the optional
relation, followed by optional DCMF entities.
<!-- snip -->
<xs:complexType name="entityType">
  <xs:sequence>
    <xs:element name="name"
type="modelID:IdentifierType"/>
    <xs:element name="characteristic"
type="characteristicType" maxOccurs="unbounded"/>
    <xs:element name="semantics" type="modelID:String"
minOccurs="0"/>
    <xs:annotation>
      <xs:documentation>lexicon entry for this entity
type</xs:documentation>
    </xs:annotation>
    <xs:element name="name"
type="dcmf:RelationIdentifierType"/>
    <xs:element name="dcmfRelation"
type="dcmf:RelationType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="semantics" type="modelID:String"
minOccurs="0"/>
    <xs:annotation>
      <xs:documentation>lexicon entry for this entity
type</xs:documentation>
    </xs:annotation>
    <xs:element name="name"
type="dcmf:EntityIdentifierType"/>
    <xs:element name="dcmfEntity"
type="dcmf:EntityType"
maxOccurs="unbounded"/>
    <xs:element name="semantics"
type="modelID:String" minOccurs="0"/>
    <xs:annotation>
      <xs:documentation>lexicon entry for
this entity type</xs:documentation>
    </xs:annotation>
  </xs:sequence>
  <xs:attributeGroup
ref="modelID:commonAttributes"/>
</xs:complexType>
<!-- snip -->

```

Listing 1: BOM Schema Extension

The following listing (Listing 2) presents an instance of the schema above. The listing is an example of a NATO-force that has as its member's one Surveillance patrol and one Medical unit. The proposed schema extension enables us to add new members (entities) to "the parent" (entity types) and to define their relationship:

```

<BOM xmlns="http://www.sisostds.org/schemas/bom"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:omt="http://www.sisostds.org/schemas/IEEE1516.2
xmlns:modelID="http://www.sisostds.org/schema/modelID"
xmlns:dcmf="http://www.foi.mil/schemas/DCMF"
xsi:schemaLocation="http://www.sisostds.org/schemas/bom
BOM_v2006.xsd">
<!-- snip -->
<entityTypes>
  <entityType>
    <name>NATO-force</name>
    <characteristic>
      <semantics>ID</semantics>
    </characteristic>
    <name>Force members</name>
    <dcmfRelation>
      <semantics>PartOf</semantics>
    </dcmfRelation>
    <name>Surveillance patrol</name>
    <dcmfEntity>
      <semantics>SP-ID</semantics>
    </dcmfEntity>
    <name>Medical unit</name>
    <dcmfEntity>
      <semantics>MU-ID</semantics>
    </dcmfEntity>
  </entityType>
<!-- snip -->

```

Listing 2: BOM-extended instance

4.3 Summary of Approach 1

The presented results indicate that incremental extensions of BOM can be made in the form of extensions of the BOM schema. Those extensions amount to adding (or altering) tags that refer to other schemas external to the BOM. As we previously stated the particular extensions given in the previous sections are not to be considered as final. Their main purpose is to explicate how such extensions can be handled in the BOM.

We think that extending the BOM by relating it to an ontology can have advantages. The main advantage is that simulations described by a BOM (or rather, a BOM Assembly which is a set of BOMs) are given the possibility to relate to common understanding of the world. That is, by relating to the same ontology the participants in a BOM Assembly agree on what concepts exist and how they are related. A consequence of this is that the level of misunderstanding among the participants is lowered. The benefits of the proposed extensions are:

- It represents a minimalistic approach for extensions of the BOM. The work done in constructing the standard remains relevant.
- By relating a BOM to a production process the quality of the BOM can be assessed. A production process, such as the DCMF, can be more or less strict in the demands put on information used in it. It is possible for an organization to make a BOM in any fashion and then publish it for others to use. However, the selection of a BOM can be made more reliable by including information about the process that led to its design.
- Adding more information to a BOM makes it more specialised. By extending the BOM as proposed, a more fine-grained selection process may be employed. A user may search in a repository for BOMs to make an assembly. The added information leads to a more suitable assembly as the risk of finding and using a sub-optimal BOM is reduced.

5. Approach 2: BOM Ontology

While XML and other similar specifications cater to the syntactic interoperability, issues on semantic heterogeneity shall still lack, as indicated by Cui, Jones and O'Brien [7]. One way to overcome these lacks is to specify the meaning and the intended context explicitly for the terminology used using an ontology. Thereafter, we define sets of mappings or translations between each set of terminologies. Ontologies are known to promote easy understanding, shared consensus view as well as semantic interoperability.

In this approach, we thus explore the use of ontology as a knowledge representation and as a medium for semantic interoperability. The context of application is the BOM specification and its use in the DCMF project. In the *first step* we propose minor ontological changes to the existing structure of BOM and end up by converting the XML document into an ontology-language, the Web Ontology Language (OWL) document [8]. In the *second step*, we enrich the BOM ontology with concepts from (Web Ontology Language Service) OWL-S [9] ontology to capture the procedural aspects of the domain. The procedural constraints are an integral aspect of the military operations that the DCMF project intends to capture. Other aspects of the domain may be further enhanced in a similar manner but we leave that for future work.

5.1 Step 1: Building the BOM Ontology

The specification of the BOM is an example of a *data model* and adheres to the ER data modelling commitment

of entities, types and relationships (aggregations, compositions and so on, see Figure 1).

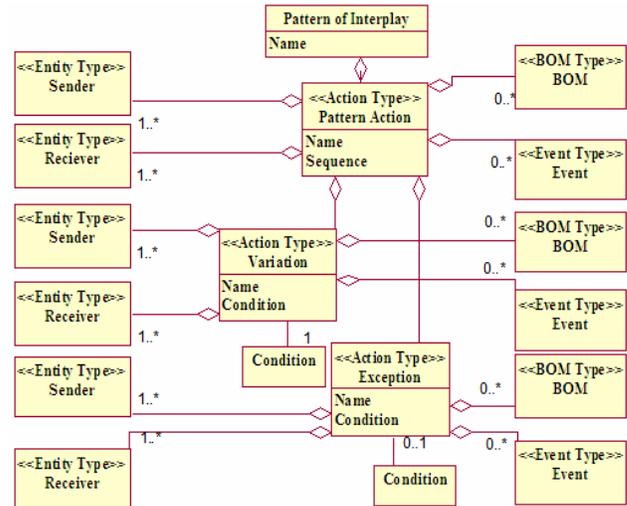


Figure 1: The original BOM/Pattern of Interplay model.

However, our requirement for the DCMF project bases on an *intensional* perspective. We would like to capture and model the meaning (semantics) and behaviour (pragmatics) of our military operations domain. Hence, we re-analyze the information that is to be modelled in the BOM and propose a modified conceptual model from the DCMF perspective as shown in Figure 2.

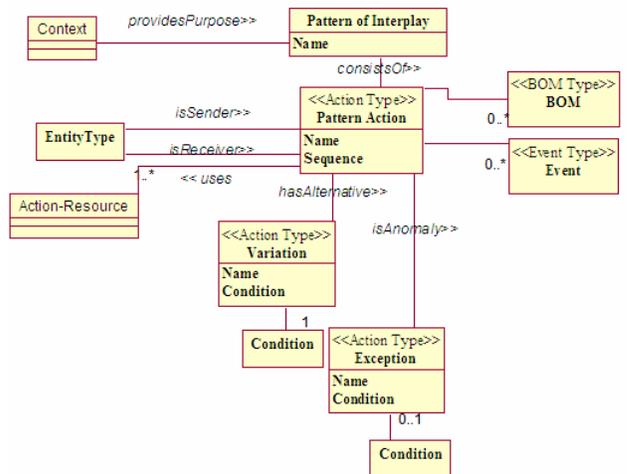


Figure 2: A modification of the BOM conceptual model

Some of the modifications made may be summarized as: a) we remove the use of multiple ‘instances’ (objects) of *sender* and *receiver* and introduce the generalized concept

of “Entity”. Semantically each *entity* may act as either *sender* or *receiver* at different points of time and in different *pattern actions*. Therefore, ‘sender’ and ‘receiver’ are different *roles* that an *entity* may play or enact out. So, we introduce the semantic relationship of *isSender* and *isReceiver* instead of a terminological entity called *sender* and *receiver*; b) we also semantically enrich the model with additional domain specific concepts such as Action-Resource, Context etc. These come from the MIP standard Joint Command Control Communication Information Exchange Data Model [5]. For more details, the reader is referred to the study [4].

We have adopted and followed the OMG prescribed standard Ontology Definition Metamodel [10] as the mapping/transformation rules from the UML conceptual model into the OWL ontology. We use Protégé Ontology Editor [11] for the actual proof of concept. So far, we have used a manual process for transforming our conceptual models in UML into OWL using Protégé. However, methods and tools for the automatic transformation are available but we have not tested them yet. After we implement the concepts as an ontology, using the Protégé for instance, we can verify if our conceptualisation has been transferred accurately.

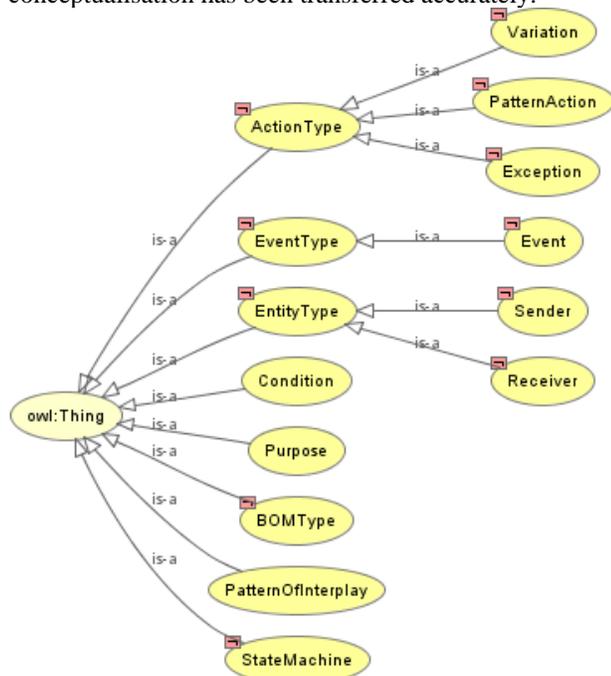


Figure 3: A tree view of the BOM ontology in Protégé

Figure 3 is a graphical view of the concepts of BOM defined in Protégé Ontology Editor and shows the

hierarchy of concepts that have been modelled. This graph is automatically extracted from the implemented ontology design. As we can see the structure follows the conceptual model proposed in Figure 2. Hence, this is a first verification that our implementation of the proposed BOM-DCMF ontology is correct.

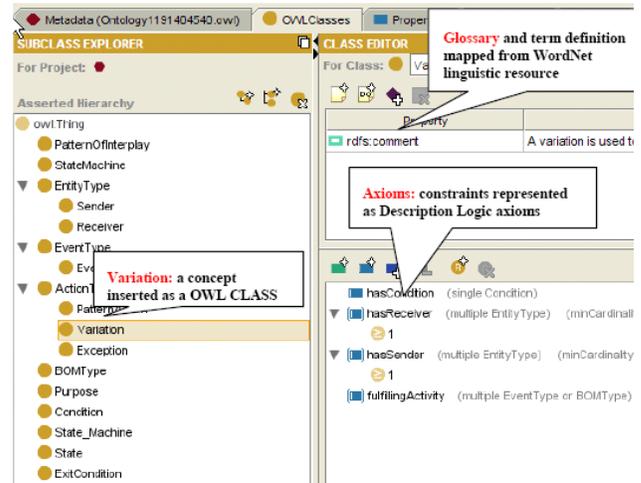


Figure 4: A screenshot of the BOM-DCMF Ontology in the Protégé Ontology Editor

Figure 4 shows a screenshot of our Proof Of Concept implementation where we can see that we have defined the concepts in a hierarchical tree view. In the right hand top column we see that we have annotated the concept with definitions from the WordNet [12]. This is called semantic enrichment. We also add word-sense tags so that we can easily traverse to get the synonyms, homonyms and other semantic relationships to the given concept. In the bottom right hand corner we see the axiomatic properties defined. For example, a Variation has been defined so that it can have one or more Sender or Receiver. It also states that the variation may be fulfilled via a fulfilling activity that could be one or more events or BOMs.

Example. To test our proposed BOM-DCMF ontology, we have used the example from the specification as seen in Table 8-3 in the BOM user guide [13] for the Weapons Effect Pattern of Interplay example. (Table 8-3 is reproduced here in Table 2 for easy cross reference.)

Pattern of Interplay		Seq	Name	Sender	Receiver	Event	BOM	Condition	Note
Weapons Effect	Pattern Action	1	Weapon Fire	Firing Entity	Target Entity	Weapon Fire	na	na	na
	Pattern Action	2	Munition Detonation	Firing Entity	Target Entity	Munition detonation	na	na	na
	Pattern Action	3	Damage State Update	Target Entity	Firing Entity	Damage state update	na	na	na

Table 2: Example of Pattern Interplay, the BOM User Guide.

We have input the same data into our BOM-DCMF ontology. In Figure 6 we see a graphical output visualization of the same, as it exists now in our ontology. As we can see, there is no loss of information from the original XML version of the BOM (Table 2). Instead, we see now a more meaningful and expressive model that is clearly easy to read and comprehend. At the same time, the inbuilt logic implies a better search and information retrieval capability.

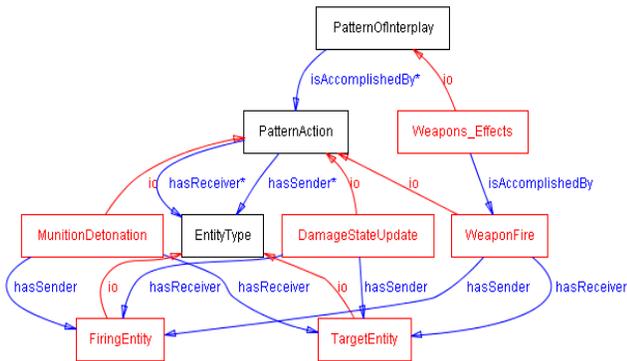


Figure 6: Example Pattern Interplay instantiated in the BOM-DCMF ontology

5.2 Step 2: Adding Behavioural Aspects

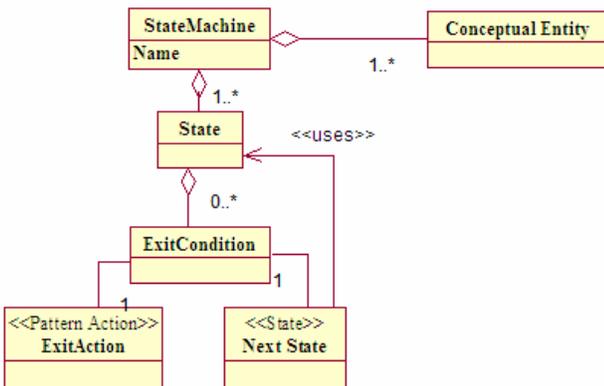


Figure 7: State Machine conceptual model from the BOM

The original State Machine conceptual model that has been proposed in the BOM specification is presented in Figure 7.

The state machine template is used to identify the behaviour states that are anticipated for a conceptual entity, in a given pattern of interplay. The state machine uses a pattern action as an exit condition when transitioning from one state to another. The state machine in the BOM is analysed mainly from the conceptual entity perspective and each state machine depicts only the states of a conceptual entity life cycle, like the example illustrated in the BOM user specification extracted in Figure 8. This does not give us the overall behaviour of the entire pattern action or the pattern of interplay.

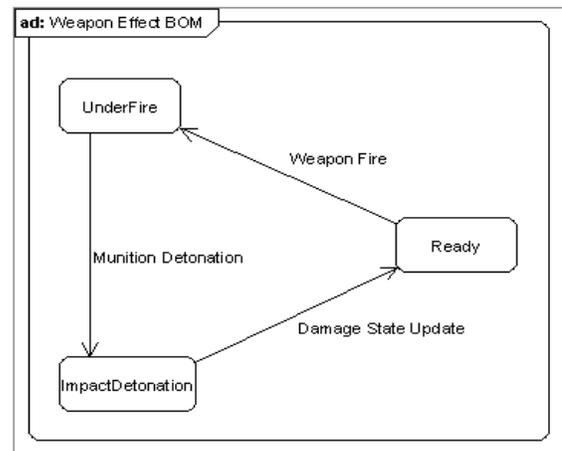


Figure 8: State Machine describing the role of the Target Entity in the Weapons Effect pattern of Interplay.

With the state machine description as provided in the BOM specification, from the DCMF perspective, we observe the following:

1. The model is inherently designed from a data-centric perspective. In DCMF we have adopted and argued for an action-centric perspective where actions themselves are objects of interest. As such, the DCMF has a higher level of abstraction than the BOM state machine.
2. The BOM state machine caters to a simple linear sequence of states, where one state can lead to another and that leads to another till a final exit condition is reached. This is a rather simple case to be assumed. It does not cover the possibility of forks, joins and other possible alternatives for transition of states. We agree that a single state can be followed by one state, but an action taken may lead to several consequences as well.
3. The BOM state machine has facility for capturing only the exit conditions. There is no mention of the pre-

condition for enabling a particular state machine. A pattern action could be part of other state machines. The same pattern action may be ‘composition block’ in more than one pattern of interplay. In all these cases, the cause and effect relationship needs to be tracked and maintained. We also foresee this to be a requirement when determining the composability of individual BOMs. For example, how should one determine if *BOM A* can be matched semantically with *BOM B*? One way would be to see if the pattern actions and state machines defined in them could be related or made to interoperate.

4. The state machine model in the original BOM stands alone and does not integrate with the Pattern of Interplay well. The only link is the Pattern Action (exit action) which is assumed to have been already defined in the Pattern of Interplay. The Pattern of Interplay gives an overview of the sequence of events or actions that are taking place; the state machine provides the dynamic states that are associated with the pattern of interplay. The entity type defines the entities that are associated. However, we do not get a composite view of the actual scenario, who is doing what, when and how. What is the cause, and what is the effect, what is needed to reach the effect and so on. We need to put together the data from all these views.

The above are some of the shortcomings that have necessitated an improvement of the existing BOM specification. By adding simple semantics, we get the improved conceptual model for the State Machine as indicated in Figure 9.

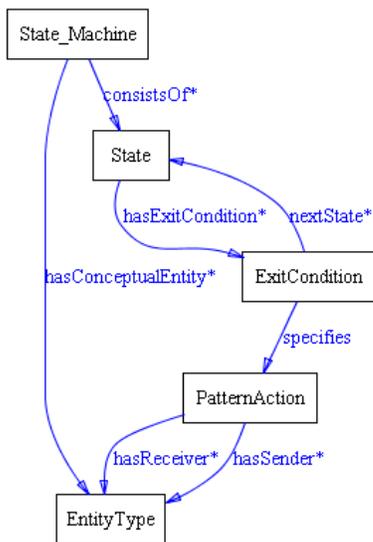


Figure 9: Modified State Machine conceptual mode

We have introduced a semantic relationship of next state instead of the class next state. We have removed the aggregation of states in a state machine and introduced the semantic association of ‘*consistsOf*’. Similarly, we have removed the class ‘*Conceptual entity*’ and replaced it with the object property relationship ‘*has Conceptual Entity*’ having the range defined to ‘*entity type*’. We also reuse our defined relationship of *hasReceiver* and *hasSender* from our modified pattern of interplay model to further strengthen the association between *PatternAction* and *EntityType*. Here too we have made a semantic change, by introducing the concept *Exit condition* that can specify or point to a required pattern action.

As discussed earlier, we would like to have the possibility to express the overall behaviour seen from the level of the entire pattern of interplay as well as the possibility to express complex decisions and alternatives. Hence, our proposal to modify the BOM to include the behaviour is to improve the pattern of interplay itself to express the high level behaviour. We may use the state machines for individual or detailed state transition descriptions. For this reason, we use OWL-S process ontology as our reference.

In the BOM, we see the Pattern Action and Pattern of Interplay have similar ideas as in the OWL-S process ontology, where Pattern Action may be seen as the simple process, and Pattern of Interplay as the Composite process. The OWL-S provides a number of control constructs, using which we can model the composite processes like *sequence*, *split*, *if-then-else*. This is the advantage that we miss in the BOM and therefore we propose introducing such constructs in the BOM following the same rationale and pattern as in the OWL-S. The state machine part of the BOM allows us with some primitive modelling of behaviour as related to the states that are produced as a result of a certain action and the exit conditions that satisfy them. But still that is not sufficient if we have to capture the semantics of a military operation that needs to be carefully planned and coordinated.

In the following, we present our final result from the two step modification that we have done to the Pattern of Interplay (BOM). In the first step, we have conceptualised the existing Pattern of Interplay as an ontology, added semantic concepts from linguistic resources, built in mappings to our domain ontology. Now, we build in the behavioural aspects by introducing the constructs from the OWL-S process ontology.

Thus, incorporating the behavioural aspects from the OWL-S process ontology, we now propose a modified Pattern of Interplay as shown in Figure 10 (we name it BOM-DCMF ontology). This version of Pattern of Interplay has been modified with concepts from the OWL-S and the DCMF ontologies. It covers the static information about the pattern of action to be taken, as well procedural flow aspects as indicated by the *SimpleAction*, *ComplexAction* and *AtomicAction*.

The state machine diagram may still be used to provide additional information of individual states and their sequences. As can be seen by the proposed BOM-DCMF ontology, we tend to favour the activity diagram more than the state machine of UML. An activity diagram is a special kind of state diagram where the sequences of activities within a process are modelled along with their conditions and results.

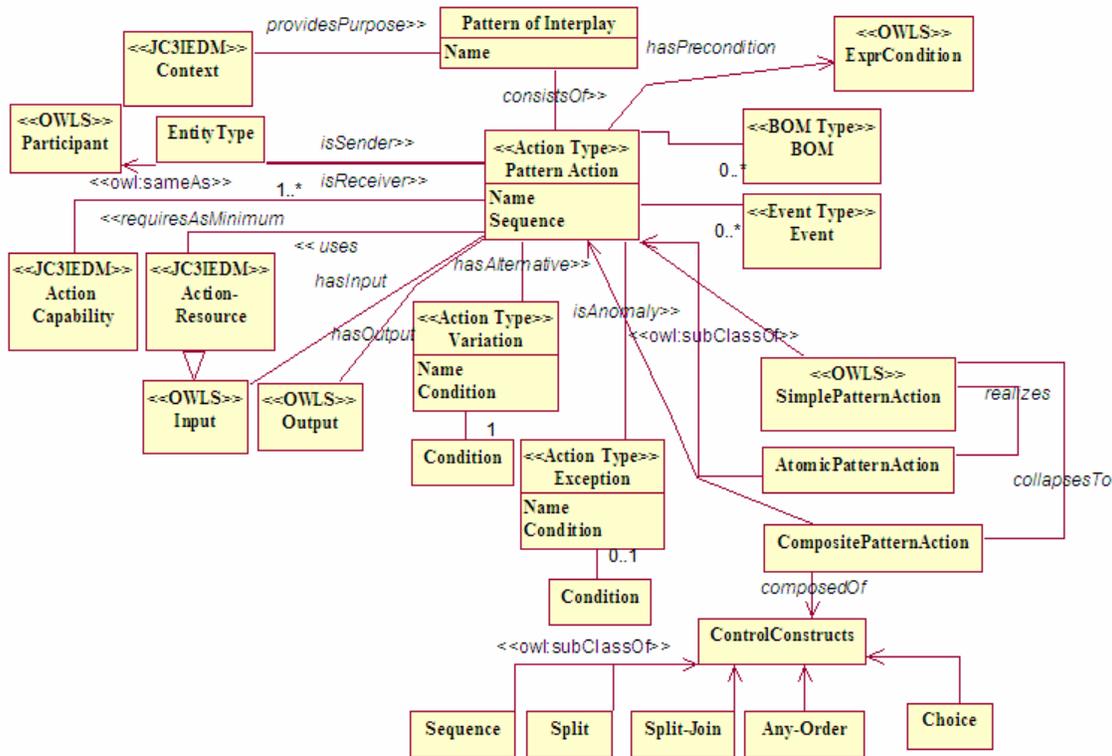


Figure 10: The complete BOM-DCMF Ontology including static and behavioral aspects

The proposed BOM-DCMF ontology as shown in Figure 10 is sufficiently enriched to capture and model the knowledge (semantic information) processed by DCMF. It is to say it may work as a final template for capturing the end result of the DCMF process. Thus, the above mentioned conceptual model may be considered as one representation format for a MSM. We have also applied the BOM-DCMF Ontology to one of our case scenarios from DCMF, but as a result of having short of place the reader is referred to the FOI methodology report [14].

5.3 Summary of Approach 2

Under Approach 2, we have looked at the BOM as a potential candidate for knowledge representation of MSMs. We have proposed a two-step enhancement to the

conceptual idea of the BOM conceptual and state machine models that has led us to proposing the BOM-DCMF ontology. In the first step, we have semantically transformed the existing data-centric conceptual model of the BOM into an ontological conceptualization. We have enriched it further by adding annotations to linguistic knowledge resource like WordNet. At the end, we have obtained a BOM ontology which is close to the original BOM specification conceptually. This intermediary version may be re-used by others who wish to have an ontology version of the existing BOM specification. In the second step, we have looked to fulfilling the specific requirements of the DCMF. Mainly we have focused on the capturing and representation of the procedural aspects of military scenarios. Such procedural aspects are

graphically represented using UML activity diagrams. We have further utilized the standard specification for the OWL-S process ontology as our reference target ontology. The final result is a modified and enriched BOM ontology which we named the BOM-DCMF Ontology. Though we have not carried out extensive evaluations of the proposed BOM-DCMF ontology, we have demonstrated its utility by applying it to some of our case scenarios from the DCMF. This proposed ontology is specific to the use envisioned in this particular project. However, the design rationale and methodology followed is equally applicable in all situations that have an action-centric perspective.

6. Conclusion and Future Work

We have in this paper proposed two approaches for extending the Base Object Model (BOM). The purpose of the approaches is to enable a richer conceptual modelling from the semantic perspective, to support improved reusability and interoperability of conceptual models for simulation purposes. The BOM is extended within the paramount of our ongoing proposal, Defence Conceptual Modelling Framework (DCMF), aimed to enable composability, interoperability and reuse of knowledge for modelling and simulation.

The two suggestions differ on amplitude of changes to BOM's existing structure and design rationale. While the first approach tries to figure out how the existing BOM specification may be extended semantically with minimal changes to its existing structure, the second approach allows integration of an ontology to the BOM. The first approach involves an incremental change of the BOM, as its principal idea, by adding a reference to an external entity, an ontology. By referring to the ontology additional information can be added to the conceptual model of the BOM. The second approach bases on the ontological conceptualisation of the DCMF and enriches the BOM sufficiently to capture the semantic information processed by the DCMF. The final outcome of this approach is the BOM-DCMF ontology.

The proposed approaches offer a number of benefits:

- Increased reusability of BOM instances by better clarifying of its content.
- A higher level of interoperability since the real meaning of things in the current context is taken in consideration.
- Reduced simulation development time by simplifying the simulation model development
- Possibility to verify semantic compatibility of BOM candidates and consequently be able to detect semantically incompatible candidates.

- Improved possibility to automatically search, find and combine suitable BOMs for simulation that is automation of a BOM Assembly.
- Encapsulation of the end-result of the DCMF in a standardized format as the BOM, increasing thus the usage degree of DCMF considerably.

We have just carried out a simple proof of concept demonstrations but much work remains to be done, before a semantically richer BOM is reality. We hope to utilise some of the lessons learned from this work and it should provide us with ample guidance on our quest for the quintessential MSM representation.

References

- [1] Homepage of DMSO describing the HLA available at: <https://www.dmsomil/public/transition/hla/>
- [2] HLA FEDEP model, Version 1,5, December 8 1999, DMSO. Available at: <https://www.dmsomil-public/library/projects/hla/guidelines/fedepv15.pdf>.
- [3] Homepage of *Base Object Model*, www.boms.info,
- [4] Mojtahed, V., Gracia Lozano, M., Svan, P., Andersson, B., Kabilan, V., *DCMF – Defence Conceptual Modelling Framework*. Methodology Report, FOI-R--1754--SE, 2005, available at: <http://www.foi.se>
- [5] *Joint Command Control and Consultation Information Exchange Data Model*, MIP - Multilateral Interoperability Programme - www.mip-site.org/
- [6] Workflow Management Coalition. Workflow Standard Process Definition Interface - *XML Process Definition Language v.2.*, 2004
- [7] Zhan Cui, Dean Jones and Paul O'Brien, *Issues in Ontology-based Information Integration*, Proceeding of IJCAI, 2001
- [8] *OWL Web Ontology Language Overview*. Available at: <http://www.w3.org/TR/owl-features/>
- [9] *OWL-S Semantic Markup for Web Services*. W3C Member Submission 22, November 2004 <http://www.w3.org/Submission/OWL-S/>,
- [10] *Object Management Group's Ontology Definition Metamodel*. Third revised submission to OMG/RFP ad/2003-03-40. 2005. Available at: <http://www.omg.org/docs/ad/05-08-01.pdf>
- [11] *Protégé Ontology Editor*. Open source tool –hosted by the Stanford University. Available at: <http://protege.stanford.edu/>
- [12] Consortium, W. W. W. (Web Resource), *Wordnet in RDFS and OWL*, <http://www.w3.org/2001/sw/Best-Practices/WNET/wordnet-sw-20040713.html>
- [13] *Guide for Base Object Model (BOM) Use and Implementation*. Simulation Interoperability Standards Organization (SISO). SISO-STD-003.1-2006. 31 March 2006
- [14] Mojtahed, V., Andersson, B., Kabilan, V., *BOM and DCMF*. Methodology Report, FOI-R--2362--SE, 2008