

Research on Owl-based Bom Ontology Model

Zhong Wei

School of Mechatronics and Automation
National University of Defense Technology
Changsha, 410073, China

Huang Kedi

School of Mechatronics and Automation
National University of Defense Technology
Changsha ,410073, China

Abstract—Model semantics is the base of model reuse, interoperation and composition. For the deficiency in semantic expression of Base object model (BOM), Ontology Web Language (OWL) was applied to re-describe the BOM specification. And a BOM ontology model was established which might supply guidance for meaningful composition between simulation components in a higher level. The results of the experiment reveal that the BOM ontology model is suitable for describing syntax and semantic information of simulation component. A knowledge components based on the BOM ontology model can be quickly searched, matched and composed and will greatly improved the developing efficiency of the simulation system.

Keywords- XML, Semantic, Composition, BOM, Ontology, OWL, SWRL

I. INTRODUCTION

Basic Object Mode(BOM) is an XML document which was developed in the open community for the purpose of supporting composable and interoperable data modeling of simulations and their capabilities[1]. With the maturity and popularization of BOM the composability problem[2] gradually emerges. XML provides standard schema and rich meta data for BOM's interface description, ensuring syntactic-level interoperability between BOMs. However, since a) XML itself does not have enough semantic representation capabilities; b) the data-centric conceptual model of the BOM makes BOM be deficient in the behavior representation[4], the current BOM standard lacks the required semantic information to support semantic-level composability of the model.

To overcome these lacks, researchers have studied and put forward some feasible solutions: [4] discusses how to improve the composability of BOM by introducing a semantic attachment, which enhances the semantic expressiveness of BOM with OWL-S(OWL for Services); [5] [6] offers an approach where BOM was extended, to become BOM++, through reference to an external ontology. All these approaches propose to alter BOM standard by some way. It did work in some cases, but destroyed the integrity of the original BOM and possibly resulted in being not compliance with the methods and concepts that are aligned with the original BOM specification. This paper presents an approach to re-describe BOM with OWL, while keeping the original BOM specification unaltered. By establishing an ontology model for BOM using OWL, the presented approach could provide for not only eliciting and creating ontology-based

knowledge components(KCs) or knowledge instances(KIs), but have the concomitant benefit of increasing the capabilities of the components for interoperability and composability from the semantic perspective.

The paper is organized as follows: Section 2 and Section 3 gives overviews of the basic concepts underlying our work. In Section 4, we describe the design and construction process of the BOM ontology. In Section 5, we test our proposed BOM ontology through a detailed composition practice. The final section concludes the study and describes further research directions.

II. BOM OVERVIEW [1][4]

The latest definition for BOM is [1]: A piece part of a conceptual model, simulation object model, or federation object model, which can be used as a building block in the development and/or extension of a simulation or federation. BOM template might be structured into four major parts: Model Identification, Conceptual Model, Model Mapping and HLA Object.

1) Model Identification. This part provides the essential metadata needed so that the BOM can be discovered and properly reused, including Point of Contact (POC), Name, Type, Keyword, VVA, Purpose, etc.,

2) Conceptual Model. The Conceptual Model provides a mechanism to identify four different components for representing the needs of a simulation: Pattern Of Interplay(POI), State Machine, Event Type and Entity Type.

Pattern of Interplay, “describing the interactions between different entities and the activities involved in the interactions”[5].

State Machine, “describing the states that the conceptual entities can have as well as the transitions between these states”[5].

Entity Types and Event Types, “identifying and describing the entities and activities used in Pattern of Interplay and State Machine”[5]

3) Model Mapping. This part is where conceptual entities and events are mapped to their HLA Object Model representations. It bridges the Conceptual Model with the HLA Object Model that is described in the fourth part of the BOM.

4) HLA Object Model. This part contains the information that is found in a normal Federation Object Model (FOM) or Simulation Object Model (SOM), such as objects, attributes, interactions and parameters, and it should conform to the HLA Object Model Template (OMT).

From the above overview, it's easy to find that the Conceptual Model is the most interesting part of BOM from the semantic perspective. It describes action-based flow and dependencies between entities and events through POI and State Machine. While the other parts bear some importance, they are more data-centric and not germane to the purposes of this article.

III. ONTOLOGY AND OWL

Ontology is used to capture knowledge about some domain of interest. An ontology describes the concepts in the domain and also the relationships that hold between those concepts. Today, ontologies have been adopted in many business and scientific communities as a way to share reuse and process domain knowledge.

Ontology description language is used to describe ontology which allows users to create clearly, formal concept descriptions for models. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the W3C[6]. It is developed as a vocabulary extension of RDF and derived from the DAML+OIL Web Ontology Language[6], for publishing and sharing ontologies on the World Wide Web. OWL provides three sublanguages designed for use by specific communities of implementers and users: OWL Lite, OWL DL and OWL Full. OWL Lite is the simplest, and the most limited OWL, supporting those users primarily needing a classification hierarchy and simple constraints, providing a quick migration path for thesaurus and other taxonomies. OWL Full have the strongest expression capability. It is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL DL bases on the Description Logics(DL)[7], supporting those users who want the maximum expressiveness while retaining computational completeness and decidability. In this paper, OWL DL is chosen to allow for reasoning and inference in BOM ontology while keeping rich expressiveness in the meantime.

IV. BUILDING THE BOM ONTOLOGY MODEL

As already mentioned, original BOM can't afford semantic-level interoperability and composability for models because of its inherent defects. Ontology, as a formal representation of the shared concept, is known to promote easy understanding, shared consensus view as well as semantic interoperability. Thus, if we could express BOM with OWL, establishing a BOM ontology model, and create KC/KI for each BOM-based component based on the ontology mode, then we should be able to make best use of the ontology to reduce ambiguities of the concepts, and satisfy the semantical requirements of the application.

A. Modeling and Composition Assumptions

Before describing our road map to the BOM ontology model, it is necessary to clarify our modeling and composition assumptions about the simulation:

1) simulation models can be seen as combinations of components and events. These components are event-driven

and act upon occurrence of events and communicate with each other through sending messages[8].

2) To avoid any confusion, we assume that each BOM just represents one entity type, which would be helpful for rapidly classifying and locating of the desired knowledge component in the repository.

3) There is one to one mapping between entity type and state machine. Though it is valid that one entity type has several state machines, it rarely happens in practice, i.e. different state machine usually maps to different entity types.

4) In the condition that the components is in right state, a message can be sent whenever the precondition of corresponding action becomes true. Similarly, a message can be received whenever the pre-condition of respective receive event becomes true [8].

5) The horizontal composition of actions is assumed. The horizontal composition and the vertical composition are two types of usual composition. We are not going to discuss the later in this paper.

B. Constructing the BOM Ontology Model

An OWL ontology consists of Individuals, Properties, Classes and Axioms, consequently, the constructing process of the BOM ontology is carrying out right around these aspects. We use Protégé 4.1 as our OWL editor. Protege is an open source platform created by Stanford [9]. The tool offers capabilities for graphically-oriented ontology development using the Protégé Editor, as well as a number of Plug-Ins and other services.

Building Classes and Relevant Hierarchy of the BOM Ontology Model

OWL Classes are interpreted as sets that contain individuals. Most OWL Classes of the BOM ontology model could be directly derived from correlative concepts in BOM, such as Model Identification, Conceptual Model, etc. The key point here is how to arrange the class hierarchy in the Ontology.

Ontologies are based on IS_A relationships, subclass is special case of its superclass. Specially, class "Thing" is the uppermost class in OWL. All the entities must possess an IS_A relationship to "Thing" and every individual must belong to at least one class. There are three ways to decide class hierarchy in an ontology: Top-Down, Bottom-Up and Hybrid. The Top-Down method firstly specifies the most general concepts, then gradually specializes those concepts in detail until all the concepts needed are clarified. The Bottom-Up method, by contrast, begins with the most specialized concepts, then generalize from bottom to top. Hybrid method starts with the decided concepts, then generalizes and specializes respectively. Since we have determined the primary concepts of the BOM ontology, we employ the Hybrid method naturally in practice(Fig.1).

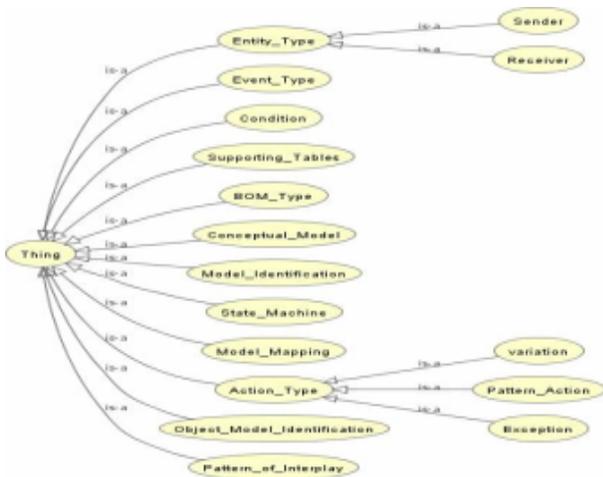


Figure 1. A tree view of BOM ontology

Building Properties of the BOM Ontology Model

It's insufficient to describe concepts only by classes and relevant hierarchy. Supposing a "Plane" class, people would also like to know its inherent characteristics (e.g. location) or external relationships (e.g. how it is related with "Radar") except the "plane" concept itself. OWL provides several methods for expressing information of the concepts/classes, primarily through Properties. OWL Properties are binary relations on individuals. There are two main types of properties in OWL: object properties and datatype properties. Object properties relate individuals to other individuals, while datatype properties relate individuals to data values, like integers. Depending on the definition of the OWL Property and the BOM structure, OWL Properties for BOM ontology can be generated by following three rules:

1) Direct Mapping Rule. The attributes of the concepts (e.g. "sequence", "name", and "receiver" of a "pattern action" in BOM) directly maps to corresponding OWL object properties (e.g. hasReceiver) and datatype properties (e.g. hasSequence, hasName) in terms of whether the attribute can be expressed as data values or relationships between individuals.

2) Indirect Mapping Rule. Some relations are implicitly contained in BOM DIF. They could be more effectively carry out if they are explicitly expressed as OWL object Properties, rather than as pure XML, in particular automated reasoning. Typical object properties of BOM ontology fall in this category includes "hasPart", "hasEvent", etc.

3) Complement Rule. Sometimes, we need to import extra properties to the BOM ontology to describe relationships absent in BOM. These kind of properties were usually proposed to meet special requirements of the application. E.g. to support horizontal composition between the components, we import "Action Mode (IN/OUT)" properties, by which we can decide whether the actions in the two components are composable or not.

Building Axioms of the BOM Ontology Model

After building OWL classes and properties, we have found a preliminary framework for BOM ontology, revealing simple relationships (IS_A) between concepts and terms. Nevertheless, it's still hard for this framework to describe a complicated domain model. To enable automated reasoning and inference, we should completely define the concepts by OWL Axioms.

Three kinds of axioms exist in OWL: Class Axiom, Property Axiom and Individual Axiom. Class axiom is built on the basis of the class description, describing different relationship between concepts. Similarly, Property Axiom describes relationship between properties and Individual Axiom describes relationship between individuals. For more instructions on the three OWL axioms, please reference [7]

Restrictions are the core of the OWL class axiom. They are used in OWL DL class axioms to provide local constraints on properties in the class, enabling knowledge mining and deducing in a deeper level. OWL Restrictions can be divided into two types: value constraints and cardinality constrain. Its function in BOM ontology could be expressed as follows:

1) Defining a defined class. A class that has at least one set of necessary and sufficient conditions is known as a Defined Class. The reasoner can only automatically classify classes under defined classes. That is to say, to enable automated classification and reasoning, all the classed in BOM ontology should be defined as a defined class.

2) Defining Closure Axioms. Reasoning in OWL is based on what is called the open world assumption (OWA)[9] which means that we can not assume something doesn't exist until it is explicitly stated that it does not exist. Closure axioms illuminate the range on a property by using restriction.

Building SWRL Rules of the BOM Ontology Model

OWL DL inherits advantages from DL, having rich expression and complete reasoning ability; however, it is constrained by DL too. OWL DL has defect in expressing general rules. For more complex inference, the user often needs to complement OWL with more expressive rules, such as those supported by Semantic Web Rule Language (SWRL) [11]. SWRL allows users to write Horn-like rules expressed in terms of OWL concepts to reason about OWL individuals or infer new knowledge from existing OWL, which provide more powerful deductive reasoning mechanisms than with the core OWL. The role of SWRL in BOM ontology model is mainly reflected in following two aspects:

1) Reasoning Rules of OWL Class/Property. Some relationships is better describe in conditions rather than in Class/Property Axioms. For example, consider an OWL property of class Company, which is constrained by the condition "If the Company and the plane are from the same side, we decide that the Company cannot fire at the plane."

It's hard to describe this restriction by axioms in OWL, but would be easily expressed with SWRL:

$Plane(? x) \cap hasflag(? x, side1) \cap Company(? y)$
 $\cap hasflag(? x, side2) \cap swrlb : equal(side1, side2) \rightarrow holdfire(? x, ? y)$

2) Matching Rules. Most of the matching rules in the BOM ontology model appear as conditions in SWRL, e.g. the pre/post conditions based on assumption 3) in section 4, state transition of state machine, etc. These conditions to actions or simulation model would be very useful in some aspects, for they describe the constraints dominating the functionality of a component, and be used in detecting two semantically mismatching actions supposed to be composed horizontally. The following rules defined a state transition process of OWL class “Company” from state “Company_ReadytoFire” to “Company_OnFire”.

$Company(? x) \cap hasState(? x, Company_ReadytoFire) \cap Plane(? y)$
 $\cap PatternAction(AAGunFire) \cap hasSender(AAGunFire, ? x)$
 $\cap hasReceiver(AAGunFire, ? y) \rightarrow hasState(? x, Company_OnFire)$

V. CASE STUDY

In order to evaluate our BOM ontology model, a simple Air Defence Experiment System scenario test application was implemented, which utilizes the Jess rule engine. The scenario can be seen at Fig.2 in sequence diagram format. There are three entities in the scenario: Plane, Company and Radar. Each component is supported by KCs/KIs based on our BOM ontology model.

A. Model Searching based on BOM Ontology

The goal of model searching is to find the KCs/KIs of interests in the repository. The problem of the traditional keyword-based searching is the lack of the data semantics. Both keyword and attribute value are only existence as machine symbols and not concerned with the semantic of the concepts. By introducing OWL, semantic search is enabled, which will improve the accuracy of the searching. As the primary focus of our work is not implementing the searching phase, we briefly describe how KCs/KIs can be extracted from a repository:

- 1) Find out possible entity types in the repository through their name(“company”) and associated attributes(e.g. hasSeriesNum, hasPosition);
- 2) Get a list of candidate KCs/KIs by properties (e.g.hasAssociateEntity) that describing relationships between entity type and BOM type;
- 3) Filter out those KCs/KIs not containing the specified event/interaction in the scenario;
- 4) Check the candidate KCs/KIs left from 3), examine the num/type of the parameters, insure that they are either identical or compatible[11].
- 5) Repeat 1-4 until find candidate component for all the entity in the scenario. For those entities can’t find a candidate component, constructing a new component and stored in the repository.

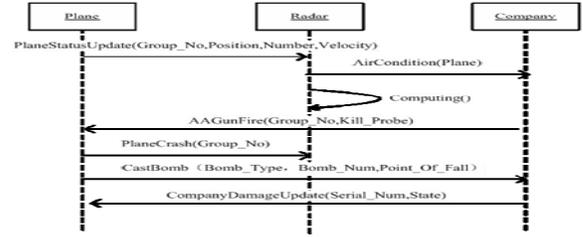


Figure 2. Scenario of ADES

B. Model Matching based on BOM Ontology

Model matching is used to find possible compositions to retrieve the conceptual models corresponding to complex requirements scenarios. From the previous assumptions, we know that simulation models/system can be seen as combinations of components and events. A successful composition depends on whether the component in the models/system have consensus on the events and actions of the simulation, not only syntactically, but also semantically. As BOM describes dynamic behaviors of the component mainly through the POIs and State Machine in BOM, the matching of BOM ontology is equal to the matching of POI and State Machines of the BOM-based KCs/KIs[11]. Therefore, the horizontal composition of the components in ADES can be described in OWL terms language as following:

Step 1: Check the POI matching status for KCs/KIs. The candidate KCs are compared in the specified events/interactions of the scenario, examining whether the participant KCs/KIs are agreed/matched against each other on the properties of those events/interactions, such as sender, receiver, entity type, interaction parameter, etc. Make sure that the KCs/KIs participating in the simulation are matched in syntactic and static semantic level.

Step 2: Check the State Machine matching status for KCs/KIs. Fig. 3, Fig.4 and Fig.5 are partial state machine of the three candidate KCs/KIs acquired after step 1. Fig.6 shows one possible sequence of event passing among the entities from the result of state machine composition. This sequence of event passing can be got by using Jess engine on the relevant OWL facts and SWRL rules[12]. In this example the composition of State Machine is shown to be successful since the sequence diagram we get is consistent with that in the scenario which means that all the events are sent when the sender entities are in the “right” state and are received by the receiving entities when they are in the state that allows them to accept those events.

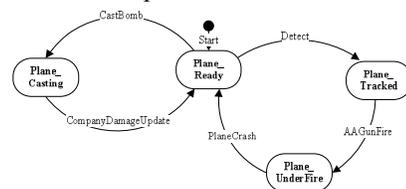


Figure 3. partial state machine of plane

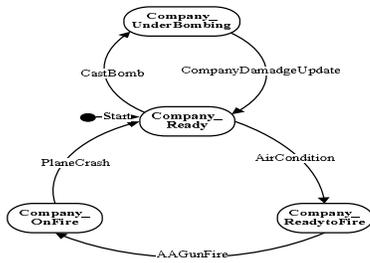


Figure 4. Partial state machine of Company

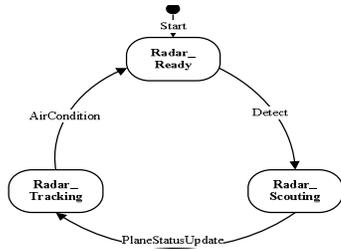


Figure 5. Partial state machine of Radar

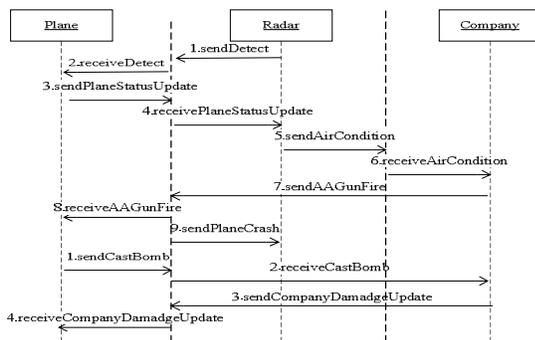


Figure 6. Sequence diagram after composition

VI. SUMMARIES AND FUTURE WORK

We have in this paper proposed an approach re-describing BOM with OWL DL. The purpose of this approach is to enable component-based simulation from the semantic perspective, while keeping original BOM unaltered. By designing and building a BOM ontology, we could a) capture the syntactic and semantic information of the concepts without fostering any extensions in BOM; b) classify the component to enable an automated reasoning and

combination on the OWL level, as well as more complex reasoning enabled by SWRL.

Practices proved that these semantically enhanced, BOM-based concepts can greatly improve the interoperability and reusability of the component, satisfying semantic-level composable requirement of component-based rapid simulation development. As for the further research we planned to a) Refine the BOM ontology model, enrich its SWRL capabilities to support automated composition of the components in a deeper level; b) Test the BOM ontology model in more scenario, design a algorithm for state machine fastly matching, optimizing the process of model composition.

REFERENCES

- [1] Base Object Model(BOM) Template Specification[S]. SISO-STD-003-2006.2006..
- [2] Guide for Base Object Model (BOM) Use and Implementation[S]. SISO-STD-003.1-2006.2006
- [3] Andreas Tolk, James Muguira. The Levels of Conceptual Interoperability Model (LCIM)[C]. 03F-SIW-007, 2003.9.
- [4] Morad F, Ayani R. A Rule-based Approach to Syntactic and Semantic Composition of BOMs[C]. 11th IEEE Symposium on Distributed Simulation and Real-time Applications, 2007.
- [5] Moradi F, Ayani R, Mokarizadeh S, Tan G. A rule-based semantic matching of base object models[J]. Simulation and Process Modelling, Vol. 5, No. 2, pp.132-145, 2009.
- [6] Mojtahed V, Andersson B, Kabilan V. BOM++, A semantically enriched BOM[C]. 08S-SIW-050,2008.
- [7] World Wide Web Consortium OWL Working Group . OWL Ontology Language Guide[S/OL]. 2009-10-10. <http://www.w3c.org/RT/owl-guide/>. Last access.
- [8] Brahim Mojtahed, Athman Bouguettaya,. A Multilevel Composability Model for Semantic Web Services[J], Journal of IEEE Transactions on Knowledge and Data Engineering, VOL. 17, No. 7,PP.954-968, 2006.
- [9] Protégé Ontology Editor: Open source tool hosted by Stanford University. Available online at <http://protege.stanford.edu/> Last accessed 2010-03-20
- [10] World Wide Web Consortium (W3C). SWRL: A Semantic Web Rule Language Combining OWL and RuleML[OL]. 2010-12-15. <http://www.w3.org/Submission/SWRL/>. Last accessed.
- [11] Mahmood I.,Ayani, R.,Vlassov V, Moradi F. Statemachine Matching in BOM based model Composition[C]. 13th IEEE/ACM International Symposium on Distributed Simulation and RealTime Applications, 2009.
- [12] Miller John A. Simulation and the semantic web[c] || Proceedings of the 2005 Winter Simulation Conference.Orlando,FL : United States, 2005 : 2371-2377.